globalscape
by HelpSystems

EFT REST API Reference

Contents

Authentication REST API

Endpoint: /admin/v1/authentication

Creates a session for the user and stores the authentication token. This token will be required for any subsequent REST API requests.

There are several supported Authentication types, EFT and AD. When using an AD account, configure your authentication type to “AD” instead of “EFT”.

- POST

Method & URL

POST https://[server URL]:[port]/admin/v2/server

POST Body Sample

```
@host = 192.168.1.206:4450
@AdminUser = API_admin
@password = APIADMIN_Password!
###

# @name login
POST https://{{host}}/admin/v1/authentication HTTP/1.1
Content-Type: application/json

{"userName": "{{AdminUser}}", "password": "{{password}}",
"authType": "EFT"}
###
@AuthToken = {{login.response.body.$.authToken}}
###
```

POST Response Body Sample

```
HTTP/1.1 200 OK Date: Tue, 10 May 2022 14:02:47 GMT Content-
Type: application/vnd.api+json Content-Length: 215 Set-Cookie:
csrftoken=6FF1EF51-9771-E2EF-9227-79C73F660B20; path=/; secure;
SameSite=Lax Cache-Control: no-cache, no-store, must-revalidate
Expires: -1 X-Content-Type-Options: nosniff X-XSS-Protection: 1;
mode=block Content-Security-Policy: default-src 'self' 'unsafe-
inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer {
"authToken":
"95267D88A45EBA20B858CA8C4EB15B00F27F1FBBC6BC0B2F13DDE25889FF2C9
E", "permissions": [ { "policy": "ServerManagement" }, {
"policy": "COMManagement" }, { "policy": "ReportManagement" }, {
"policy": "ManagePersonalData" } ] }
```

POST PowerShell Sample

```
$baseURL = "https://192.168.100.165:4450/admin"
$AdminUser = "admin"
$password = "alaska"

# Only to ignore certificates errors
add-type @"
    using System.Net;
    using System.Security.Cryptography.X509Certificates;

        public class IDontCarePolicy : ICertificatePolicy {
        public IDontCarePolicy() {}
        public bool CheckValidationResult(
            ServicePoint sPoint, X509Certificate cert,
            WebRequest wRequest, int certProb) {
            return true;
        }
    }
"@
[System.Net.ServicePointManager]::CertificatePolicy = new-object
IDontCarePolicy
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12


# authentication
$authBody = "{""userName"": ""$AdminUser"", ""password"": ""$password"",
""authType"": ""EFT""}"
$auth = Invoke-RestMethod -Uri "$baseURL/v1/authentication" -Method 'POST' -
Body $authBody

$authToken = $auth.authToken
$authHeader = New-Object
"System.Collections.Generic.Dictionary[[String],[String]]"
$authHeader.Add("Authorization", "EFTAdminAuthToken $authToken")
```

Server REST API

Endpoint: /server

Returns and update server settings.

- GET

- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/server

GET Body Sample

```
GET https://{{host}}/admin/v2/server HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": {
"type": "server",
"id": "1",
"attributes": {
 "version": "8.0.6.16",
 "general": {
   "configFilePath": "C:\\ProgramData\\Globalscape\\EFT  Server
Enterprise\\",
   "enableUtcInListings":  true,
    "lastModifiedBy":  "admin",
    "lastModifiedTime":  1633717057
   },
   "listenerSettings":  {
     "adminPort": 1100,
     "enableRemoteAdministration":  true,
      "listenIps":  [ "0.0.0.0" ],
…

}
```

GET PowerShell Sample

```
#get Server
Write-Output "Get Server"
Write-Output "-----"
$serverDetails = Invoke-RestMethod -Uri "$baseURL/v2/server" -Method 'GET' -
Headers $authHeader
Write-Output $serverDetails | ConvertTo-Json
foreach ($server in $serverDetails.data)
{
    Write-Output ("Server $($serverDetails.data)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/server

PATCH Body Sample

```
PATCH  https://{{host}}/admin/v2/server HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}


{
"data": {
    "type": "server",
    "id": "1",
    "attributes":{
        "version":"",
    "smtp": {
        "login": "",
        "password": "",
        "port": 587,
        "senderAddr": "EFT@mailserver.net",
        "senderName": "Globalscape EFT",
        "server": "smtp.mailserver.net",
        "useAuthentication": false,
        "useImplicitTLS": false
      }
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

PATCH PowerShell Sample

```
##Patch Server - patching the SMTP settings
Write-Output "Server Patch"
Write-Output "-----"
$update =
'{
    "data":{
        "type": "server",
        "id": "1",
        "attributes": {
            "version": "",
        "smtp": {
            "login": "",
            "password": "",
            "port": 587,
            "senderAddr": "EFT@mailserver.net",
            "senderName": "Globalscape EFT",
            "server": "smtp.mailserver.net",
            "useAuthentication": false,
            "useImplicitTLS": false
         }
         }
     }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/server" -Method 'PATCH' -
Headers $authHeader -Body $update
```

Endpoint:/server/status

Returns server information

- GET

Method & URL

GET https://[server URL]:[port]/admin/v1/server/status

GET Body Sample

```
GET https://{{host}}/admin/v1/server/status HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

{"nodes":[{"activity":{"activeClientDownloadBytesPerSecond":0,"activeClientDownloadCount":0,"activeClientUploadBytesPerSecond":0,"activeClientUploadCount":0,"activeServerDownloadBytesPerSecond":0,"activeServerDownloadCount":0,"activeServerUploadBytesPerSecond":0,"activeServerUploadCount":0,"connectedAdminCount":1,"connectedUserCount":0,"runningAwTaskCount":0,"runningEventRuleCount":0},"isInCluster":false,"isMaster":false,"isStarted":true,"licenses":[{"module":"AAMModule","registrationState":"Evaluation period"},{"module":"ARMModule","registrationState":"Evaluation period"},{"module":"AS2Module","registrationState":"Evaluation period"},{"module":"AWModule","registrationState":"Evaluation

GET PowerShell Sample

```
#get Server Metrics
Write-Output "Get Server Metrics"
Write-Output "-----"
$serverMetrics = Invoke-RestMethod -Uri "$baseURL/v1/server/status" -Method
'GET' -Headers $authHeader
Write-Output $serverMetrics | ConvertTo-Json
foreach ($server in $serverMetrics.data)
{
    Write-Output ("Server $($serverMetrics.data)")
}
```

Endpoint: /server/metrics

Returns server metrics.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/server/metrics

GET Body Sample

```
GET https://{{host}}/admin/v2/server/metrics HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": {
  "type": "serverMetrics",
   "id": "1",
   "attributes": {
      "activeClientDownloadBytesPerSecond": 0,
       "activeClientDownloadCount": 0,
       "activeClientUploadBytesPerSecond": 0,
       "activeClientUploadCount": 0,
       "activeServerDownloadBytesPerSecond": 0,
       "activeServerDownloadCount": 0,
       "activeServerUploadBytesPerSecond": 0,
       "activeServerUploadCount": 0,
       "connectedAdminCount": 1,
       "connectedUserCount": 0,
       "runningAwTaskCount": 0,
       "runningEventRuleCount": 0
      }
    },
    "links": {
      "self": "admin/v2/server/metrics"
  }
}
```

GET PowerShell Sample

```
#get Server Metrics
Write-Output "Get Server Metrics"
Write-Output "-----"
$serverMetrics = Invoke-RestMethod -Uri "$baseURL/v2/server/metrics" -Method
'GET' -Headers $authHeader
Write-Output $serverMetrics | ConvertTo-Json
foreach ($server in $serverMetrics.data)
{
    Write-Output ("Server $($serverMetrics.data)")
}
```

Endpoint: /server/security/tls

Returns and updates server TLS security settings.

- Get
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/server/security/tls

GET Body Sample

```
GET https://{{host}}/admin/v2/server/security/tls HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": {
  "type": "tlsServerSecuritySettings",
  "id": "1",
  "attributes": {
   "tlsSettings": {
      "minimumProtocolVersion": "TLSv1.3",
      "selectedAlgorithms":
"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA2
56:EECDH+CHACHA20:EECDH+AES128:RSA+AES128:EECDH+AES256:RSA+AES256:EECDH+3
DES:RSA+3DES:EECDH+RC4:RSA+RC4:!MD5:!SRP:!PSK:!EXP:!RC4:!SSLv3:!SHA1",
     "resultingCipherList":
"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA2
56:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-
AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-
CCM8:ECDHE-ECDSA-AES128-CCM:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-
SHA256:AES128-GCM-SHA256:AES128-CCM8:AES128-CCM:AES128-SHA256:ECDHE-
ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-
CCM8:ECDHE-ECDSA-AES256-CCM:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-
SHA384:AES256-GCM-SHA384:AES256-CCM8:AES256-CCM:AES256-SHA256",
     "manuallySelectedAlgorithms": {
       "enabled": false,
       "algorithms":
"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA2
56:EECDH+CHACHA20:EECDH+AES128:RSA+AES128:EECDH+AES256:RSA+AES256:EECDH+3
DES:RSA+3DES:EECDH+RC4:RSA+RC4:!MD5:!SRP:!PSK:!EXP:!RC4:!SSLv3:!SHA1"
     }
   },
   "tlsFipsEnabled": false
   } },
   "links": {
      "self": "admin/v2/server/security/tls"
   }
}
```

GET PowerShell Sample

```
#get Server TLS Security Settings
Write-Output "Get Server TLS Settings"
Write-Output "-----"
$serverTLS = Invoke-RestMethod -Uri "$baseURL/v2/server/security/tls" -Method
'GET' -Headers $authHeader
Write-Output $serverTLS | ConvertTo-Json
foreach ($server in $serverTLS.data)
{
    Write-Output ("Server $($serverTLS.data)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/server/security/tls

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/server/security/tls HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "tlsServerSecuritySettings",
    "id": "1",
    "attributes": {
      "tlsFipsEnabled": true
    }
  },
  "links": {
    "self": "admin/v2/server/security/tls"
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

PATCH PowerShell Sample

```
###(PATCH) TLS Security Settings (enable FIPS)
Write-Output "Server TLS Enable FIPS Patch"
Write-Output "-----"
$update =
'{
 "data": {
    "type": "tlsServerSecuritySettings",
    "id": "1",
    "attributes": {
      "tlsFipsEnabled": true
    }
  },
  "links": {
    "self": "admin/v2/server/security/tls"
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/server/security/tls" -
Method 'PATCH' -Headers $authHeader -Body $update
```

Endpoint: /server/security/ssh

Returns and update server SSH security settings.

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/server/security/ssh

GET Body Sample

```
GET https://{{host}}/admin/v2/server/security/ssh HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": {
  "type": "sshServerSecurity",
  "id": "1",
  "attributes": {
    "sshSettings": {
     "allowedCiphersList": "aes256-gcm@openssh.com,aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-gcm@openssh.com,aes128-ctr,aes128-cbc",
     "allowedKexesCiphersList": "ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group16-sha512,diffie-hellman-group14-sha256,diffie-hellman-group-exchange-sha256",
     "allowedMacsList": "hmac-sha2-512-etm@openssh.com,hmac-sha2-512,hmac-sha2-256-etm@openssh.com,hmac-sha2-256,hmac-sha1-etm@openssh.com,hmac-sha1",
     "softwareVersion": "8.1.0.0_openssh",
     "comments": "Globalscape"
    },
     "sshFipsEnabled": false
   }
 },
 "links": {
   "self": "admin/v2/server/security/ssh"
 }
}
```

GET PowerShell Sample

```
#get Server SSH Security Settings
Write-Output "Get Server SSH Settings"
Write-Output "-----"
$serverSSH = Invoke-RestMethod -Uri "$baseURL/v2/server/security/ssh" -Method
'GET' -Headers $authHeader
Write-Output $serverSSH | ConvertTo-Json
foreach ($server in $serverSSH.data)
{
    Write-Output ("Server $($serverSSH.data)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/server/security/ssh

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/server/security/ssh HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "sshServerSecurity",
    "id": "1",
    "attributes": {
      "sshFipsEnabled": true
    }
  },
  "links": {
    "self": "admin/v2/server/security/ssh"
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

PATCH PowerShell Sample

```
###(PATCH) SSH Security Settings (enable FIPS)
Write-Output "Server SSH Enable FIPS Patch"
Write-Output "-----"
$update =
'{
 "data": {
    "type": "sshServerSecurity",
    "id": "1",
    "attributes": {
      "sshFipsEnabled": true
    }
  },
  "links": {
    "self": "admin/v2/server/security/ssh"
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/server/security/ssh" -
Method 'PATCH' -Headers $authHeader -Body $update
```

Endpoint: /server/reports

Returns the DPIA and PCI-compliance reports in JSON format.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/server/reports/DPIA

GET Body Sample

```
GET https://{{host}}/admin/v2/server/reports/DPIA HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": {
  "attributes": {
    "dpia": [ {
        "articles": [ {
            "desc": "The processing of personal data falls within Article 2's definition of material scope. Risk is increased only if set to unknown scope.",
            "name": "Material scope",
            "num": 2,
            "score": 1,
            "value": "Unknown or undefined" }, {
            "desc": "Specifies the territorial scope your organization has identified in context with this application's usage.",
            "name": "Territorial scope",
            "num": 3,
            "score": 1,
            "value": "Unknown or undefined" }, {
            "desc": "If Not in Union was chosen for territorial scope then all users must have their data subject status set to either Yes or No.",


…
  }
```

GET PowerShell Sample

```
#get Server DPIA Report
Write-Output "Get Server DPIA Report"
Write-Output "-----"
$serverDPIA = Invoke-RestMethod -Uri "$baseURL/v2/server/reports/DPIA" -Method
'GET' -Headers $authHeader
Write-Output $serverDPIA | ConvertTo-Json -Depth 20
foreach ($server in $serverDPIA.data)
{
    Write-Output ("Server $($serverDPIA.data)")
}
```

Method & URL

(Please note this is a V1 endpoint)

GET https://[server URL]:[port]/admin/v1/server/reports/PCICompliance

GET Body Sample

```
GET https://{{host}}/admin/v1/server/reports/PCICompliance HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Tue, 09 Aug 2022 17:30:22 GMT Content-Type:
application/vnd.api+json Content-Length: 234 Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval'
data:; Referrer-Policy: no-referrer { "PCICompliance": [ { "failed": 12,
"passed": 12, "siteId": "e82de253-b8ec-4b70-a72d-d017c3317f01",
"siteName": "MySite", "warnings": 6 }, { "failed": 6, "passed": 11, "siteId":
"ff493a39-1e28-40c5-ab18-2e6a4f33f296", "siteName": "PCI_SITE",
"warnings": 5 } ] }
```

GET PowerShell Sample

```
#get Server PCIC Report
Write-Output "Get Server PCIC Report"
Write-Output "-----"
$serverPCIC = Invoke-RestMethod -Uri
"$baseURL/v1/server/reports/PCICompliance" -Method 'GET' -Headers $authHeader
Write-Output $serverPCIC | ConvertTo-Json -Depth 20
foreach ($server in $serverPCIC.data)
{
    Write-Output ("Server $($serverPCIC.data)")
}
```

Endpoint: /server/admin-users

Returns list of EFT admin users and creates new admin users.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/server/admin-users

GET Body Sample

```
GET https://{{host}}/admin/v2/admin-users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": [
 {
     "id": "59774c12-870d-5560-a44a-c5b35cf68d4c",
  "type": "adminUser",
  "attributes": {
   "name": "admin"
  }
 },
 {
  "id": "b76a1d7b-c3ad-5ac4-8636-06ce71ba3af3",
  "type": "adminUser",
  "attributes": {
   "name": "Local computer\\Administrators"
  }
 }
 ],
 "links": {
  "self": "admin/v2/admin-users"
 }
}
```

GET PowerShell Sample

```
#get Server admin users
Write-Output "Get Server admin users"
Write-Output "-----"
$serverAdmins = Invoke-RestMethod -Uri "$baseURL/v2/admin-users" -Method
'GET' -Headers $authHeader
Write-Output $serverAdmins | ConvertTo-Json
foreach ($server in $serverAdmins.data)
{
    Write-Output ("Server $($serverAdmins.data)")
}
```

Method & URL

POST https://[server URL]:[port]/admin/v2/server/admin-users

POST Body Sample

```
POST https://{{host}}/admin/v2/admin-users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
 "data":{
   "attributes":{
   "name":"Test_admin2",
   "password":"test",
   "adminConsolePermissions":{
    "accountPolicy":"server",
    "enableCom":true,
    "enableEditingAndReporting":true,
    "enablePersonalDataAccess":true
   },
   "restPermissions":{
    "enabled":true,
    "restAdminRole":true
   }
  }
 }
}
```

POST Response Body Sample

```
HTTP/1.1 201 Created Date: Tue, 09 Aug 2022 18:18:00 GMT Content-Type:
application/vnd.api+json ETag: "BF7E8C0E" Content-Length: 394 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer

{ "data":

{ "id": "9693cbcf-d4a9-49ac-aa1c-59efd91a692b", "type": "adminUser",
"attributes": { "adminConsolePermissions": { "accountPolicy": "Server",
"enableCom": true, "enableEditingAndReporting": true,
"enablePersonalDataAccess": true }, "name": "Test_admin2",
"restPermissions": { "enabled": true, "restAdminRole":
"server_full_access" } } }, "links": { "self": "/admin/v2/admin-
users/9693cbcf-d4a9-49ac-aa1c-59efd91a692b" } }
```

POST PowerShell Sample

```
#POST Server Admin user (New User)
Write-Output "POST Server Admin (New Admin user)"
Write-Output "-----"
$update =
'{
  "data": {
    "id": "1",
    "attributes": {
      "name": "Test_admin",
      "password": "test",
      "adminConsolePermissions": {
        "accountPolicy": "server",
        "enableCom": true,
        "enableEditingAndReporting": true,
        "enablePersonalDataAccess": true
      },
      "restPermissions": {
        "enabled": true,
        "restAdminRole": true
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/admin-users" -Method
'POST' -Headers $authHeader -Body $update
```

Endpoint: /server/admin-users/{adminId}

Returns EFT admin user (adminID required) information, which allows PATCH (update) and DELETE of admin user. To obtain the admin user ID a GET request will need to be done against the /admin-users endpoint prior to making this REST API request.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/server/admin-users/{adminID}

GET Body Sample

```
GET https://{{host}}/admin/v2/admin-users/59774c12-870d-5560-a44a-
c5b35cf68d4c HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": {
  "id": "59774c12-870d-5560-a44a-c5b35cf68d4c",
  "type": "adminUser",
  "attributes": {
    "adminConsolePermissions": {
      "accountPolicy": "Server",
      "enableCom": true,
      "enableEditingAndReporting": true,
      "enablePersonalDataAccess": true
     },
     "name": "admin",
     "restPermissions": {
       "enabled": true,
       "restAdminRole": "server_full_access"
     }
   }
 },
 "links": {
   "self": "admin/v2/admin-users/59774c12-870d-5560-a44a-c5b35cf68d4c"
 }
}
```

GET PowerShell Sample

```
#Get Server Admin Details when providing a Server Admin ID
Write-Output "Get Server Admin Details"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri "$baseURL/v2/admin-
users/59774c12-870d-5560-a44a-c5b35cf68d4c" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($server in $serverAdminDetails.data)
{
    Write-Output ("Server $($serverAdminDetails.data)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/server/admin-users/{adminID}.

PATCH Body Sample

```
@adminID = 13564c1b-8e13-4e72-b8e0-9480fad06221
PATCH https://{{host}}/admin/v2/admin-users/{{adminID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "0",
    "attributes": {
        "adminConsolePermissions": {
        "enableCom": false,
        "enableEditingAndReporting": true,
        "enablePersonalDataAccess": true
      },
      "restPermissions": {
        "enabled": true,
        "restAdminRole": true
      }
    }
  }
}
```

PATCH PowerShell Sample

```
###
#PATCH Server Admin Details when providing a Server Admin ID
Write-Output "Server Admin User Patch"
$adminID = "13564c1b-8e13-4e72-b8e0-9480fad06221"
Write-Output "Admin ID: $adminID"
Write-Output "-----"
$update =
'{
  "data": {
    "id": "0",
    "attributes": {
        "adminConsolePermissions": {
        "enableCom": false,
        "enableEditingAndReporting": true,
        "enablePersonalDataAccess": true
      },
      "restPermissions": {
        "enabled": true,
        "restAdminRole": true
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/admin-users/$adminID" -
Method 'PATCH' -Headers $authHeader -Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/server/admin-users/{adminID}.

DELETE Body Sample

```
@adminID = 13564c1b-8e13-4e72-b8e0-9480fad06221
DELETE https://{{host}}/admin/v2/admin-users/{{adminID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE PowerShell Sample

```
###
#DELETE EFT admin by Admin ID
Write-Output "Server Admin DELETE"
$adminID = "f98d2a35-4d37-4891-89ed-66cb6ddf9436"
Write-Output "Admin ID: $adminID"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri "$baseURL/v2/admin-
users/$adminID" -Method 'DELETE' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($server in $serverAdminDetails.data)
{
    Write-Output ("Server $($serverAdminDetails.data)")
}
```

Endpoint: /server/admin-users-policy

Returns EFT admin user policies such as account and password policy. Additionally, the PATCH request allows updates to the EFT admin policies.

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/server/admin-users-policy

GET Body Sample

```
GET https://{{host}}/admin/v2/admin-users-policy HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": {

      "id": "1",

      "type": "adminUserPolicy",

      "attributes": {

        "accountPolicy": {

          "invalidLogin": {

            "account": {

              "enabled": false,

              "attemptPeriodMinutes": 5,  "attemptRetries": 5,
"lockoutMinutes": 30 }}, "inactivity": { "disconnectDueToTimeout": {
"enabled": false,   "timeoutMinutes": 5},"removeInactiveAccounts":
{"enabled": false,  maxInactiveDays": 90}}},   "passwordPolicy": {
"complexity": {          "enabled": false,          "minLength": 8,
"userNameSimilarity": {            "enabled": true,
"maxAllowedChars": 3 }, "repeatingCharLimit": { "enabled": true,
"maxAllowedChars": 3 }, "characterCategories": { "enabled": true,
"characterCount": 4, "numericRequired": true, "upperCaseRequired": true,
"lowerCaseRequired": true, "nonAlphaNumericRequired": false,
"non7bitAsciiRequired": false }, "forbiddenDictionary": { "enabled":
true, "backwardsWordsAllowed": false, "dictionaryFilePath": "" } },
"forceResetAfterFirstLogin": false, "history": { "enabled": false,
"historyDepth": 4 }, "reset": { "expiration": { "enabled": false,
"daysToExpire": 90 } } } } }, "links": { "self": "admin/v2/admin-users-
policy" } }
```

GET PowerShell Sample

```
###
#GET Server Admin user policy
Write-Output "Get Server Admin User Policy"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri "$baseURL/v2/admin-users-policy"
-Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($server in $serverAdminDetails.data)
{
    Write-Output ("Server $($serverAdminDetails.data)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/server/admin-users-policy

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/admin-users-policy HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "1",
    "type": "adminUserPolicy",
    "attributes": {
      "accountPolicy": {
          "inactivity": {
          "disconnectDueToTimeout": {
            "enabled": false,
            "timeoutMinutes": 5
          }

        }
      }
    }
  }
}
```

PATCH PowerShell Sample

```
###
#PATCH Server Admin user policy (disable Admin UI disconnect)
Write-Output "Server Admin User Policy Patch"
Write-Output "-----"
$update =
'{
  "data": {
    "id": "1",
    "type": "adminUserPolicy",
    "attributes": {
      "accountPolicy": {
          "inactivity": {
          "disconnectDueToTimeout": {
            "enabled": false,
            "timeoutMinutes": 5
          }

        }
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/admin-users-policy" -Method
'PATCH' -Headers $authHeader -Body $update
```

Endpoint: /experimental/server/templates/{templateName}

Note: Available as of EFT 8.1.0

Returns and updates the contents of the specified server template.

- GET
- PATCH

Method & URL

GET HTTPS://{{host}}/admin/experimental/server/templates/{templateName}

GET Body Sample

```
#GET Server Templates -- ##
GET
https://{{host}}/admin/experimental/server/templates/WorkspacesAdminAddedParticipantMs
g.html HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Mon, 08 Aug 2022 13:17:16 GMT Content-Type:
application/vnd.api+json Content-Length: 370 Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer { "data":
"<html>\r\n<body>\r\n <p>Hello %USER.LOGIN%,</p>\r\n <p>The following
<b>Enhanced File Transfer</b> Workspace has been shared with you:
%FOLDER_NAME% (owner: %WS_OWNER_NAME%).</p>\r\n <p>Please use the
following URL to access this Workspace:</p>\r\n <a
href=\"%LINK%\">%LINK%</a><br/><br/>\r\n \r\n %WS_MESSAGE%<br/><br/>\r\n
<hr/>\r\n</body>\r\n</html>" }
```

GET PowerShell Sample

```
#####################################
#GET Server Templates -- ##
Write-Output "GET Server Templates Detail"
Write-Output "-------------------------------"
#$siteID = "Enter site ID"
#$TemplateName= "WorkspacesAdminAddedParticipantMsg.html"
$TemplateDetails = Invoke-RestMethod -Uri
"$baseURL/experimental/server/templates/WorkspacesAdminAddedParticipantMsg.htm
l" -Method 'GET' -Headers $authHeader
Write-Output $TemplateDetails | ConvertTo-Json -Depth 25
```

Method & URL

PATCH HTTPS://{{host}}/admin/experimental/server/templates/{templateName}

PATCH Body Sample

```
###
#PATCH - Update Server template ###
PATCH
https://{{host}}/admin/experimental/server/templates/PasswordResetReminderMsg.html
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": "<html>\r\n<body>\r\n HELLO USERS %USER.FULL_NAME%,<br>\r\nThe -___API
TEST___ password for account: %USER.LOGIN% will expire in %DAYS_LEFT% days. Please
change your password at your earliest convenience.\r\nInstructions for changing your
password via FTP, SFTP, and HTTP/S are provided below for your convenience:<br>\r\n1.
Please enter the following URL into your browser:
%SITE.ACCOUNT_MANAGEMENT_URL%<br>\r\n2. Supply your current password when
prompted<br>\r\n3. Enter your new password and confirm<br>\r\n4. If approved, exit the
browser and login as normal<br>\r\n</body>\r\n</html>"
}
```

PATCH PowerShell Sample

```
### Update Content of Template###
### Modify Server Template - ###
Write-Output "PATCH Server Template"
Write-Output "---------------------"
$update =
'
{
  "data":  "Notice: ___PATCH TEST__The SSL certificate: %certid% is set to expire %d% day(s)
from now on %expirationdate%, which may impact operations or communications with the
entity associated with this certificate."
}
'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$TemplateDetails = Invoke-RestMethod -Uri
"$baseURL/experimental/server/templates/SSLCertExpirationWarning.txt" -Method 'PATCH' -
Headers $authHeader -Body $update
```

Endpoint: /server/ha/upgrade

Note: Available as of EFT 8.1.0

The GET request returns node HA upgrade state. Additionally, the POST request changes the entire HA cluster state to the upgrade or normal state.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/server/ ha/upgrade

GET Body Sample

```
### GET server upgrade state ###
GET https://{{host}}/admin/v2/server/ha/upgrade HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": "notInUpgrade" }
```

GET PowerShell Sample

```
#########
#Get Server Upgrade State
Write-Output "Get Server Upgrade State: "
Write-Output "-----"
$upgradeState = Invoke-RestMethod -Uri "$baseURL/v2/server/ha/upgrade" -Method
'GET' -Headers $authHeader
Write-Output $upgradeState | ConvertTo-Json -Depth 10
```

Method & URL

POST https://[server URL]:[port]/admin/v2/server/ ha/upgrade

POST Body Sample

```
### "Normal" state -> "Upgrade" state ###
POST https://{{host}}/admin/v2/server/ha/upgrade HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data":
    {
        "action":"begin"
    }
}
###
```

POST PowerShell Sample

```
###
#POST Server Upgrade State to “begin”
Write-Output "Server Upgrade State POST"
Write-Output "-----"
$update =
'{
    "data":
    {
        "action":"begin"
    }
}'
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/server/ha/upgrade" -Method
'POST' -Headers $authHeader -Body $update
```

Endpoint:/server/admin/v2/authentication

Note: Available as of EFT 8.1.0

Return info about admin session

- GET

Method & URL

GET https://{{host}}/admin/v2/authentication

GET Body Sample

```
###
GET https://{{host}}/admin/v2/authentication HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":{"authToken":"1B1EFBE3D705EEAABC6404F3E1705C46C93E9962C6DA618A8EE534133A81C097","idleTimeoutMinutes":1,"username":"admin"}}
```

GET PowerShell Sample

```
# authentication
$authBody = "{""userName"": ""$AdminUser"", ""password"": ""$password"",
""authType"": ""EFT""}"
$auth = Invoke-RestMethod -Uri "$baseURL/v1/authentication" -Method 'POST' -
Body $authBody

$authToken = $auth.authToken
$authHeader = New-Object
"System.Collections.Generic.Dictionary[[String],[String]]"
$authHeader.Add("Authorization", "EFTAdminAuthToken $authToken")

# keep-alives
Write-Output "administration metrics"
Write-Output "----"
for( $count = 0; $count -lt 10; $count++) {
   $adminInfo = Invoke-RestMethod -Uri "$baseURL/v2/authentication" -Method
'GET' -Headers $authHeader
   Write-Output $adminInfo
   }
```

Nodes REST API

Endpoint: /nodes

Returns a list of EFT Nodes.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/nodes

GET Body Sample

```
GET https://{{host}}/admin/v2/nodes HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": [
    {
      "type": "node",
      "id": "WIN-4NFJASSP163",
      "links": {
        "metrics": "admin/v2/nodes/WIN-4NFJASSP163/metrics"
      },
      "attributes": {
        "isStarted": true,
        "isInCluster": false,
        "fullDomainName": "WIN-4NFJASSP163",
        "version": "8.0.7.4",
        "hostIps": [
          "127.0.0.1",
          "fe80::351a:6044:dfe2:f53%2"
        ] } } ], "links": { "self": "admin/v2/nodes/" } }
```

GET PowerShell Sample

```
###
#GET SERVER NODES
Write-Output "Get EFT Server Nodes"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri "$baseURL/v2/nodes" -Method 'GET'
-Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($server in $serverAdminDetails.data)
{
    Write-Output ("Server $($serverAdminDetails.data)")
}
```

Endpoint: /nodes/{nodeId}/metrics

Returns an EFT server's metrics, including transfers and server status. To obtain the node ID a GET request will need to be done against the /nodes endpoint prior to making this REST API request .

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/nodes/{nodeID}/metrics

GET Body Sample

```
GET https://{{host}}/admin/v2/nodes HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "type": "nodeMetrics",
    "id": "WIN-4NFJASSP163",
    "attributes": {
      "activity": { "activeClientDownloadBytesPerSecond": 0,
      "activeClientDownloadCount": 0,
      "activeClientUploadBytesPerSecond": 0,
      "activeClientUploadCount": 0,
      "activeServerDownloadBytesPerSecond": 0,
      "activeServerDownloadCount": 0,
      "activeServerUploadBytesPerSecond": 0,
      "activeServerUploadCount": 0,
      "connectedAdminCount": 1,
      "connectedUserCount": 0,
      "runningAwTaskCount": 0,
      "runningEventRuleCount": 0
     },
      "isInCluster": false, "isMaster": false, "isStarted": true,
```

GET PowerShell Sample

```
###
#GET SERVER NODE METRICS
Write-Output "Get EFT Server Node Metrics"
$serverNodeID = "WIN-4NFJASSP163"
Write-Output "Server Node ID: $serverNodeID"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/nodes/$serverNodeID/metrics" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($server in $serverAdminDetails.data)
{
    Write-Output ("Server $($serverAdminDetails.data)")
}
```

Endpoint: /nodes/{nodeId}/licenses

Returns list of licenses on an EFT node and provides the ability to register EFT or a module.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/nodes/{nodeId}/licenses

GET Body Sample

```
@NodeID = WIN-4NFJASSP163
GET https://{{host}}/admin/v2/nodes/{{NodeID}}/licenses HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
"data": [
  {
    "attributes": {
      "registrationStatus": "Evaluation period"
    },
    "id": "AAMModule",
    "links": {
      "self": "admin/v2/nodes/WIN-4NFJASSP163/licenses/AAMModule"
    },
     "type": "license" }, { "attributes": { "registrationStatus":
"Evaluation period" }, "id": "ARMModule", "links": { "self":
"admin/v2/nodes/WIN-4NFJASSP163/licenses/ARMModule" }, "type": "license"
},
```

GET PowerShell Sample

```
###
#GET SERVER NODE LICENSE
Write-Output "Get EFT Server Node Licenses"
$serverNodeID = "WIN-4NFJASSP163"
Write-Output "Server Node ID: $serverNodeID"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/nodes/$serverNodeID/licenses" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Method & URL

POST https://[server URL]:[port]/admin/v2/nodes/{nodeId}/licenses

POST Body Sample

```
@NodeID = WIN-4NFJASSP163
POST https://{{host}}/admin/v2/nodes/{{NodeID}}/licenses HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": [
    {
      "id": "ARMModule",
      "attributes": {
        "serialNumber": "MySerialNumberHere"
      }
    }
  ]
}
```

POST PowerShell Sample

```
###
#POST SERVER NODE LICENSE
Write-Output "Register EFT Server Node"
$serverNodeID = "WIN-4NFJASSP163"
Write-Output "Server Node ID: $serverNodeID"
Write-Output "-----"
$update =
'{
  "data": [
    {
      "id": "ARMModule",
      "attributes": {
        "serialNumber": "MySerialNumberHere"
      }
    }
  ]
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/nodes/$serverNodeID/licenses" -Method 'POST' -Headers $authHeader
-Body $update
```

Available options for ids:

- AAMModule (Advanced Authentication Modes)
- ARMModule (Auditing and Reporting Module)
- AS2Module (Applicability Statement 2 Module)
- AWModule (Advanced Workflow Engine Module)
- AccelerateModule (FAST Protocol – ***Deprecated***)
- CloudConnectorModule (CCM – Cloud Connector Module)
- DMZGWModule (DMZ Gateway Module)
- EFTServer (EFT Server)
- EnterpriseActionsWModule (Enterprise Actions Module)
- FTPModule (FTP/S Module)

- FileTransferClientModule (File Transfer Client Module)
- FolderMonitorEventModule (Folder Monitor Event Module)
- HSModule (*will be deprecated*)
- HTTPModule (HTTP/s Module)
- PGPModule (OpenPGP Module)
- RAModule (Remote Agent Module)
- SSHModule (SFTP Module)
- SecureFormsModule (Secure Forms Module)
- TimerEventModule (Timer Event Module)
- WorkspacesModule (Workspaces Module)

Endpoint: /nodes/{nodeId}/licenses/{moduleName}

Returns registration status of a given module

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/nodes/{nodeId}/licenses/{moduleName}

GET Body Sample

```
###
#GET Module Registration Status
@moduleName = EFTServer
GET https://{{host}}/admin/v2/nodes/{{NodeId}}/licenses/{{moduleName}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { "registrationState": "Evaluation period" },
"id": "EFTServer", "type": "license" }, "links": { "self":
"admin/v2/nodes/DTY2019WinServe/licenses/EFTServer" } }
```

GET PowerShell Sample

```
###
#GET A SERVER NODE LICENSE MODULE INFORMATION
Write-Output "Get EFT Server Node Licenses"
$serverNodeID = "DTY2019WinServe"
$licenseModule = "EFTServer"

Write-Output "License Module: $licenseModule"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/nodes/$serverNodeID/licenses/$licenseModule" -Method 'GET' -
Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Endpoint: /nodes/{nodeId}/licenses/trial-extension-challenge

Returns trial extension challenge string

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/nodes/{nodeId}/licenses/trial-extension-challenge

GET Body Sample

```
###
#GET TRIAL EXTENSION CHALLENGE
@NodeID = “DTY2019WinServe”
GET https://{{host}}/admin/v2/nodes/{{NodeId}}/licenses/trial-extension-
challenge HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { "challengeString":
"nnPwL0AAq2304GWn0KnniO0z85owwH6xL3vw8pQm0LvdUVE10=" }, "id": "1",
"type": "trialExtension" }, "links": { "self": "admin/v2/licenses/trial-
extension-challenge" } }
```

GET PowerShell Sample

```
###
#GET TRIAL EXTENSION CHALLENGE
Write-Output "Get EFT Server Node Licenses"
$serverNodeID = "DTY2019WinServe"

Write-Output "$serverNodeID"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/nodes/$serverNodeID/licenses/trial-extension-challenge" -Method
'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Endpoint: /nodes/{nodeId}/licenses/trial-extension-response

Allows the user to provide response string to extend the trial.

- POST

Method & URL

POST https://[server URL]:[port]/admin/v2/nodes/{nodeId}/licenses/trial-extension-response

POST Body Sample

```
@NodeID = "DY-Win2019-1a"

POST https://{{host}}/admin/v2/nodes/{{NodeID}}/licenses/trial-extension-response HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "responseString":
"XN00P00PSJ00P00PUD00P00PK02JZLM3RJM7968J3LKKKQ01VWZVBU3VLXMXKF03VNLMXAJOLOKKWONYUE3COMNQV8KE2JFEO6YNQ3KZT3AU3RMOTBJXU0MLUE3UQELK3JOM1JN8L3XOS3TNYKE2TL7NJ3TKWMTLZJ0XP00PT8WE5TWETUFK39TB396UDMTM5VW1VM3PSWTQV9TKE1TQ3P6JEI2KE7VUE0TKW2TO6WTWTKEZTWE3W7TUF3RKFW3M3PS83PS03PS03RCPSB3RCPS83RCT"
    }
  }
}
###
```

POST Response Body Sample

```
HTTP/1.1 200 OK Date: Tue, 09 Aug 2022 22:17:23 GMT Content-Type:
application/vnd.api+json Content-Length: 212 Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer {
"data": { "attributes": { "response": "Trial Extended! Your new trial END
date is 8/24/2022." }, "id": "1", "type": "trialExtension" }, "links": {
"self": "/admin/v2/nodes/\"DY-Win2019-1a\"/licenses/trial-extension-
response" } }
```

Site REST API

Endpoint: /sites

Returns and update site(s) settings.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites

GET Body Sample

```
### GET SITE ###
# @name siteList

GET https://{{host}}/admin/v2/sites HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": [ { "type": "site", "id": "892b16dc-24a8-473f-a74e-
c597b824c879", "attributes": { "name": "MySite" }, "links": { "self":
"admin/v2/sites/892b16dc-24a8-473f-a74e-c597b824c879", "metrics":
"admin/v2/sites/892b16dc-24a8-473f-a74e-c597b824c879/metrics" } },

{ "type": "site", "id": "5f99540a-da95-41c4-8b3c-d423848bdd46",
"attributes": { "name": "MySite2" }, "links": { "self":
"admin/v2/sites/5f99540a-da95-41c4-8b3c-d423848bdd46", "metrics":
"admin/v2/sites/5f99540a-da95-41c4-8b3c-d423848bdd46/metrics" } } ],
"links": { "self": "admin/v2/sites" } }
```

GET PowerShell Sample

```
# Get all sites
Write-Output "Site"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites" -Method 'GET' -
Headers $authHeader
Write-Output $siteList | ConvertTo-Json
foreach ($site in $siteList.data)
{
    Write-Output ("Site  $($site.id): $($site.attributes.name)")
}

$siteID = $siteList.data[0].id
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites

POST Body Sample

```
#POST Create a new Site
POST https://{{host}}/admin/v2/sites HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "site",
    "attributes": {
      "pciComplianceEnabled": true,
      "siteLabel": "MySite5",
      "siteRootFolder": "C:\\InetPub\\EFTRoot\\MySite\\",
      "createUnixStyleSubFolders": true,
      "createFoldersForNewUsers": true
    }
  }
}
```

POST Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated POST request(s).

POST PowerShell Sample

```
####Added 2/01/2020
#POST Site/ (Create a new Site)
Write-Output "POST Site (Create a new Site)"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "site",
    "attributes": {
      "pciComplianceEnabled": true,
      "siteLabel": "MySite6",
      "siteRootFolder": "C:\\InetPub\\EFTRoot\\MySite\\",
      "createUnixStyleSubFolders": true,
      "createFoldersForNewUsers": true
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites" -Method 'POST' -
Headers $authHeader -Body $update
```

Endpoint:/sites/status

Returns the current status of all sites

- GET

Method & URL

GET https://[server URL]:[port]/admin/v1/sites/status

GET Body Sample

```
### GET SITE ###
# @name siteList

GET https://{{host}}/admin/v1/sites/status HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

{"info":[{"nodeName":"DY-Win2019-1a","sites":[{"DMZConnected":false,"activity":{"activeClientDownloadBytesPerSecond":0,"activeClientDownloadCount":0,"activeClientUploadBytesPerSecond":0,"activeClientUploadCount":0,"activeServerDownloadBytesPerSecond":0,"activeServerDownloadCount":0,"activeServerUploadBytesPerSecond":0,"activeServerUploadCount":0,"connectedUserCount":0,"runningAWTaskCount":0,"runningEventRuleCount":0},"id":"5ceae6e3-11b1-40c6-b4e4-3078a8e88a35","lastUserLoginTime":0,"name":"MySite","startTime":1659543775,"state":"Running"}]}]}

GET PowerShell Sample

```
# Get all sites
Write-Output "Site"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v1/sites/status" -Method 'GET' -
Headers $authHeader
Write-Output $siteList | ConvertTo-Json
foreach ($site in $siteList.data)
{
    Write-Output ("Site  $($site.id): $($site.attributes.name)")
}
```

Endpoint: /sites/{siteId}

Returns all site settings for the specified Site ID; To obtain the Site ID a GET request will need to be done against the /sites endpoint prior to making this request.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteid}

GET Body Sample

```
### Returns all site settings for the specified Site ID ###
GET https://{{host}}/admin/v2/sites/{{Siteid}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
      "type": "site",
      "id": "892b16dc-24a8-473f-a74e-c597b824c879",
      "attributes": { "name": "MySite",
      "general": { "authType": "EFT",
       "rootFolder": "C:\\InetPub\\EFTRoot\\MySite\\",
       "lastModifiedBy": "WIN-IKTLG8MULV9\\Administrator",
       "lastModifiedTime": 1643636803
      },
      "listenerSettings": {
       "accountManagementUrl": "https://172.22.150.62/manageaccount",
       "httpDomain": "172.22.150.62",
       "listenIps": [ "0.0.0.0" ],
       "ftps": { "explicitPort": 21,
       "implicitPort": 990 },
       "https": { "port": 443 },
       "sftp": { "port": 22 } },
       "workspacesSettings": { "enabled": true },
       "dmz": { "enabled": false,
       "server": [ { "nodeName": "WIN-IKTLG8MULV9",
       "ipAddresses": [ "" ],
       "port": 44500 } ] } } },
       "links": { "metrics": "admin/v2/sites/892b16dc-24a8-473f-a74e-
      c597b824c879/metrics",
      "self": "admin/v2/sites/892b16dc-24a8-473f-a74e-c597b824c879" }
      }
```

GET PowerShell Sample

```
####Added 2/01/2020
### Returns all site settings for the specified Site ID ###
### User must obtain the SiteID from Get Sites
Write-Output "Site"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID" -Method 'GET' -
Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Endpoint: /sites/{siteId}/metrics

Returns all site metrics for the specified Site ID; To obtain the Site ID a GET request will need to be done against the /sites endpoint prior to making this request.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteid}/metrics

GET Body Sample

```
### GET Site Metrics ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/metrics/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "id": "4d1b02ee-5c9c-4129-bf1d-663534248b3f",
    "type": "siteMetrics",
    "attributes": {
      "name": "MySite6",
      "startTime": " 12:00:00 AM",
      "isRunning": false,
      "isDMZGatewayConnected": false,
      "definedUsers": 0,
      "disabledUserCount": 0,
      "nodes": [
        {
          "activeClientDownloadBytesPerSecond": 0,
          "activeClientDownloadCount": 0,
          "activeClientUploadBytesPerSecond": 0,
          "activeClientUploadCount": 0,
          "activeServerDownloadBytesPerSecond": 0,
          "activeServerDownloadCount": 0,
          "activeServerUploadBytesPerSecond": 0,
          "activeServerUploadCount": 0,
          "connectedUserCount": 0,
          "name": "WIN-IKTLG8MULV9",
          "runningAWTaskCount": 0,
          "runningEventRuleCount": 0
        }
      ]
    }
```

GET PowerShell Sample

```
### Returns all site Metrics for the specified Site ID ###
### User must obtain the SiteID from Get Sites
Write-Output "Site Metrics"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/metrics" -Method
'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Endpoint: /sites/{siteId}/authentication

Returns current authentication settings for the Site ID specified on the request; To obtain the Site ID a GET request will need to be done against the /sites endpoint prior to making this request.

- GET
- PATCH

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteid}/authentication

GET Body Sample

```
### GET Site Authentication Provider (EFT, AD, LDAP, ODBC) ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/authentication/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "id": "3e2cae3a-db0f-44bd-812d-4ca6b0f04c91",
    "type": "siteAuthentication",
    "attributes": {
      "advancedAuth": {},
      "userAuth": {
        "type": "LDAP",
        "ldap": {
          "advanced": {
            "ldapVersion": {
              "enabled": true,
              "version": 3
            },
            "overrideSearchPage": {
              "enabled": true,
              "size": 1000
            },
            "referralChasingEnabled": false,
            "searchScope": {
              "enabled": true,
              "level": 2
            },
            "serverSidePageControlEnabled": true,
            "timeOut": {
              "enabled": true,
              "seconds": 60
            },
```

GET PowerShell Sample

```
####Added 2/02/2022
### GET Site Authentication Provider (EFT, AD, LDAP, ODBC) ###
### User must obtain the SiteID from Get Sites
Write-Output "Site Authentication Provider "
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/authentication"
-Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteid}/authentication

PATCH Body Sample

```
###This example will Set ODBC as the Authentication type and enable
radius
#PATCH Site Authentication type
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/authentication/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
  "data": {
    "id": "",
    "type": "site",
    "attributes": {
      "userAuth": {
        "type": "ODBC",
        "odbc": {
          "connectionString":
"Provider=MSDASQL;SERVER=;DATABASE=ODBC;UID=sa;PWD= ",
          "encryptUserPassword": true,
          "refreshInterval": {
            "inherited": true,
            "enabled": false,
            "minutes": 0
          }
        }
      },
      "advancedAuth": {
        "type": "radius",
        "radius": {
          "server": "",
          "port": 0,
          "nasIdentifier": "",
          "sharedSecret": "",
          "connectionRetries": 0,
```

GET Response Body Sample

```
{
  "data": {
    "id": "c4d4bf76-2172-41e7-9ed1-3c2d0de1254c",
    "type": "siteAuthentication",
    "attributes": {
      "advancedAuth": {
        "type": "radius",
        "radius": {
          "connectionRetries": 0,
          "nasIdentifer": "",
          "port": 0,
          "server": "",
          "sharedSecret": "",
          "timeOutSeconds": 0
        }
      },
      "userAuth": {
        "type": "ODBC",
        "odbc": {
        "connectionString":
"Provider=MSDASQL;SERVER=;DATABASE=ODBC;UID=sa;PWD=",
          "encryptUserPassword": true,
          "refreshInterval": {
            "inherited": true,
            "enabled": 0,
            "minutes": 0
          }
        }
```

PATCH PowerShell Sample

```
### PATCH Site Authentication Provider (EFT, AD, LDAP, ODBC) ###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH Site Authentication Provider"
Write-Output "-----"
$update =
'{
  "data": {
    "id": "",
    "type": "site",
    "attributes": {
      "userAuth": {
        "type": "ODBC",
        "odbc": {
          "connectionString":
"Provider=MSDASQL;SERVER=192.168.100.70;DATABASE=ODBC;UID=sa;PWD=alaska",
          "encryptUserPassword": true,
          "refreshInterval": {
            "inherited": true,
            "enabled": false,
            "minutes": 0
          }
        }
      },
      "advancedAuth": {
        "type": "radius",
        "radius": {
          "server": "",
          "port": 0,
          "nasIdentifier": "",
          "sharedSecret": "",
          "connectionRetries": 0,
          "timeOutSeconds": 0
        }
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/authentication" -Method 'PATCH' -Headers
$authHeader -Body $update
```

Endpoint: /sites/{siteId}/general

Returns and update the site general settings.

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/general

GET Body Sample

```
### GET Site General Settings ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/general/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "type": "siteGeneral",
    "attributes": {
      "overrideVfsCredentials": {
        "enabled": false,
        "value": {
          "login": "",
          "password": ""
        }
      },
     "siteRootFolder":  "C:\\InetPub\\EFTRoot\\MySite\\"
    }
  },
  "links": {
    "self": "/admin/v2/sites/892b16dc-24a8-473f-a74e-
c597b824c879/general"
  }
}
```

GET PowerShell Sample

```
### GET Site General Settings###
### User must obtain the SiteID from Get Sites
Write-Output "Site General Settings"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/general" -Method
'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Note: As of EFT 8.1.0 the general REST API endpoint will also return any VFS Site Root folder configuration details when the site root has been configured in the cloud.

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/general

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/general/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "",
    "type": "",
    "attributes": {
      "siteRootFolder": "C:\\InetPub\\EFTRoot\\MySite\\",
      "overrideVfsCredentials": {
        "enabled": true,
        "value": {
          "login": "test",
          "password": "passwordtest"
        }
      }
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
{
  "data": {
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "type": "siteGeneral",
    "attributes": {
      "overrideVfsCredentials": {
        "enabled": true,
        "value": {
          "login": "test",
          "password": "cGFzc3dvcmR0ZXN0"
        }
      },
```

PATCH PowerShell Sample

```
### PATCH Site General Settings###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH Site General Settings"
Write-Output "-----"
$update =
'{
  "data": {
    "id": "",
    "type": "",
    "attributes": {
      "siteRootFolder": "C:\\InetPub\\EFTRoot\\MySite\\",
      "overrideVfsCredentials": {
        "enabled": true,
        "value": {
          "login": "test",
          "password": "passwordtest"
        }
      }
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/general" -
Method 'PATCH' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteID}/ftps

Returns and update the FTP(s) settings

- GET

- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ftps

GET Body Sample

```
### GET Site FTP(s) Settings ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/ftps/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "type": "site",
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "attributes": {
      "ftp": {
        "enabled": false,
        "port": 21
      },
      "explicitFtps": {
        "enabled": true,
        "port": 21
      },
      "implicitFtps": {
        "enabled": true,
        "port": 990
      },
      "ftpConfig": {
        "enableClearCommandChannel": false,
        "enableFtpClientAntiTimeout": true,
```

GET PowerShell Sample

```
### GET Site FTP(S) Settings###
### User must obtain the SiteID from Get Sites
Write-Output "Site FTP(S) Settings"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/ftps" -Method
'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/general

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/ftps/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes":{
      "ftp": {
        "enabled": true,
        "port": 10
      }
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
{
  "data": {
    "type": "site",
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "attributes": {
      "ftp": {
        "enabled": true,
        "port": 10
      },
      "explicitFtps": {
        "enabled": true,
        "port": 10
      },
      "implicitFtps": {
```

PATCH PowerShell Sample

```
### PATCH FTP(S) Settings###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH FTP(S) Settings"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes":{
      "ftp": {
        "enabled": true,
        "port": 25
      }
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/ftps" -Method
'PATCH' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteId}/https

Returns and update the HTTP(s) settings

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/https

GET Body Sample

```
### GET Site HTTP(s) Settings ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/https/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "type": "site",
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "attributes": {
      "http": {
        "enabled": false,
        "port": 80
      },
      "https": {
        "enabled": true,
        "port": 411
      },
      "domain": "172.22.150.62",
      "enableRestPage": false,
      "redirectHttpToHttps": false,
      "enableHsts": true,
      "allowWebServices": false,
      "as2": {
        "enabled": false,
```

GET PowerShell Sample

```
### GET Site HTTP(s) Settings###
### User must obtain the SiteID from Get Sites
Write-Output "Site HTTPS(S) Settings"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/https" -Method
'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/https

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/https/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes":{
      "https": {
        "enabled": true,
        "port": 490
      }
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
{
  "data": {
    "type": "site",
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "attributes": {
      "http": {
        "enabled": false,
        "port": 80
      },
      "https": {
        "enabled": true,
        "port": 490
      },
      "domain": "172.22.150.62",
      "enableRestPage": false,
      "redirectHttpToHttps": false,
      "enableHsts": true,
      "allowWebServices": false,
      "as2": {
        "enabled": false,
```

PATCH PowerShell Sample

```
### PATCH  Site HTTP(s) Settings###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH  Site HTTP(s) Settings"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes":{
      "https": {
        "enabled": true,
        "port": 25
      }
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/https" -
Method 'PATCH' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteId}/networkAndSecurity

Returns and update the network and security settings.

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/networkandsecurity/

GET Body Sample

```
### GET Site Network and Security Settings ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/networkandsecurity/
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "type": "site",
    "id": "892b16dc-24a8-473f-a74e-c597b824c879",
    "attributes": {
      "connectionLimits": {
        "maxTransferSpeed": false,
        "maxConcurrentSocketConnections": false,
        "maxConcurrentLogins": false,
        "maxConnectionsPerUser": false,
        "maxConnectionFromSameIP": false
      },
      "autoBan": {
        "enabled": true,
        "value": {
          "banIfExcessiveInvalidCmdsReceived": true,
          "banIpsPermanently": false,
          "numberOfInvalidCmds": 5,
          "sensitivityLevel": "Medium"
        }
```

GET PowerShell Sample

```
####Added 2/03/2022
### GET Site Network and Security Settings###
### User must obtain the SiteID from Get Sites
Write-Output "GET Site Network and Security Settings"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/networkandsecurity" -Method 'GET' -Headers
$authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/1networkandsecurity/

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/networkandsecurity/
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "connectionLimits": {
        "maxConcurrentLogins": true,
        "maxConcurrentSocketConnections": true,
        "maxConnectionFromSameIP": true,
        "maxConnectionsPerUser": true,
        "maxTransferSpeed": true
      },
      "denialOfService": {
        "autoBan": {
          "enabled": true,
          "value": {
            "banIfExcessiveInvalidCmdsReceived": true,
            "numberOfInvalidCmds": true,
            "sensitivityLevel": "Low",
            "banIpsPermenantly": true
          }
        }
      }
    }
  }
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
HTTP/1.1 200 OK Date: Thu, 25 Aug 2022 18:58:06 GMT Content-Type:
application/vnd.api+json ETag: "88131696" Content-Length: 663 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "type": "site", "id": "c9551b5d-f904-4612-b690-
d51bfde2eb1f", "attributes": { "connectionLimits": { "maxTransferSpeed":
true, "maxTransferSpeedValue": 820, "maxConcurrentSocketConnections":
true, "maxConcurrentSocketConnectionsValue": 10, "maxConcurrentLogins":
true, "maxConcurrentLoginsValue": 10, "maxConnectionsPerUser": true,
"maxConnectionsPerUserValue": 1, "maxConnectionFromSameIP": true,
"maxConnectionFromSameIPValue": 1 }, "autoBan": { "enabled": true,
"value": { "banIfExcessiveInvalidCmdsReceived": true,
"banIpsPermanently": false, "numberOfInvalidCmds": 5, "sensitivityLevel":
"Medium" } } } }, "links": { "self": "/admin/v2/sites/c9551b5d-f904-4612-
b690-d51bfde2eb1f/networkAndSecurity" } }
```

PATCH PowerShell Sample

```
##Patch SITE Network Security settings
Write-Output "Site Network Security PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes": {
      "connectionLimits": {
        "maxConcurrentLogins": true,
        "maxConcurrentSocketConnections": true,
        "maxConnectionFromSameIP": true,
        "maxConnectionsPerUser": true,
        "maxTransferSpeed": true
      },
      "denialOfService": {
        "autoBan": {
          "enabled": true,
          "value": {
            "banIfExcessiveInvalidCmdsReceived": true,
            "numberOfInvalidCmds": true,
            "sensitivityLevel": "Low",
            "banIpsPermenantly": true
          }
        }
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchNSReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/networkandsecurity" -Method 'PATCH' -Headers
$authHeader -Body $update
Write-Output $patchNSReturn | ConvertTo-Json
```

Endpoint: /sites/{siteId}/ipAccess

Returns and update the network and security Site IP access/ban list Settings.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAccessList

GET Body Sample

```
### GET Site IP access/ban list Settings ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/ipAccessList HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": [
    {
      "id": "192.168.100.*",
      "type": "access",
      "attributes": {
        "accessType": "Allow",
        "date": "2022-02-04",
        "isAutoBan": false,
        "reason": "Allow for test",
        "sequenceNumber": 1
      }
    },
    {
      "id": "0.0.0.0",
      "type": "access",
      "attributes": {
        "accessType": "Deny",
        "date": "1969-12-31",
        "isAutoBan": true,
```

GET PowerShell Sample

```
### GET Site IP access/ban list Settings###
### User must obtain the SiteID from Get Sites
Write-Output "GET Site IP access/ban list Settings"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/ipAccessList" -
Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAccessList

POST Body Sample

```
POST https://{{host}}/admin/v2/sites/{{Siteid}}/ipAccessList HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "139:10:112:56::246",
    "type": "access",
    "attributes": {
      "accessType": "Allow",
      "reason": "API TEST",
      "sequenceNumber": 4
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
{
  "data": {
    "id": "139:10:112:56::246",
    "type": "access",
    "attributes": {
      "accessType": "Allow",
      "date": "2022-02-04",
      "isAutoBan": false,
      "reason": "API TEST",
      "sequenceNumber": 4
    }
  },
  "links": {
    "self": "/admin/v2/sites/a70ccebd-1317-4992-84e7-
bb4cbddc02e4/ipAccessList/139:10:112:56::246"
  }
}
```

PATCH PowerShell Sample

```
### POST Site IP access/ban list Settings###
### User must obtain the SiteID from Get Sites
Write-Output "POST Site IP access/ban list Settings"
Write-Output "-----"
$update =
'
{
  "data": {
    "id": "139:10:112:57::246",
    "type": "access",
    "attributes": {
      "accessType": "Allow",
      "reason": "API TEST",
      "sequenceNumber": 5
    }
  }
}
'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/ipAccessList"
-Method 'POST' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteId}/ipAccessList/{ipAccessId}

Returns, update and deletes network and security Site IP access/ban list Settings for the specified IP.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAccessList/ipAccessId

GET Body Sample

```
### GET Site IP access/ban Settings For specific IP###
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/ipAccessList/{{ipAccessId}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "id": "139:10:112:56::246",
    "type": "access",
    "attributes": {
      "accessType": "Allow",
      "date": "2022-02-04",
      "isAutoBan": false,
      "reason": "API TEST",
      "sequenceNumber": 2
    }
  },
  "links": {
    "self": "/admin/v2/sites/a70ccebd-1317-4992-84e7-
bb4cbddc02e4/ipAccessList/139:10:112:56::246"
  }
}
```

GET PowerShell Sample

```
####Added 2/07/2022
### GET Site IP access/ban Settings For specific IP ###
### User must obtain the SiteID from Get Sites
Write-Output "GET Site IP access/ban list Settings For specific IP "
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ipAccessList/$ipAccessId" -Method 'GET' -Headers
$authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAccessList/ipAccessList/{ipAccessId}

PATCH Body Sample

```
## PATCH Site Network and Security Settings For specific IP###
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/ipAccessList/{{ipAccessId}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "139:10:112:56::247",
    "attributes": {
      "accessType": "Allow",
      "reason": "This is a test for API PATCH on IP",
      "sequenceNumber": 2
    }
  }
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
{ "data": { "id": "139:10:112:56::247", "type": "access", "attributes": {
"accessType": "Allow", "date": "2022-02-07", "isAutoBan": false,
"reason": "This is a test for API PATCH on IP", "sequenceNumber": 2 } },
"links": { "self": "/admin/v2/sites/a70ccebd-1317-4992-84e7-
bb4cbddc02e4/ipAccessList/139:10:112:56::247" } }
```

PATCH PowerShell Sample

```
### PATCH Site Network and Security Settings For specific IP###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH Site Network and Security Settings For specific IP"
Write-Output "-----"
$update =
'{
  "data": {
    "id": "139:10:112:56::247",
    "attributes": {
      "accessType": "Allow",
      "reason": "This is a test for PS PATCH on IP",
      "sequenceNumber": 2
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 9
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ipAccessList/$ipAccessId" -Method 'PATCH' -Headers
$authHeader -Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAccessList/{ipAccessId}

DELETE Body Sample

```
### DELETE Site IP access/ban Settings For specific IP###
DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/ipAccessList/{{ipAccessId}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Tue, 09 Aug 2022 20:34:49 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1 X-
Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
####Added 2/07/2022
### DELETE Site IP access/ban Settings For specific IP ###
### User must obtain the SiteID from Get Sites
### User must obtain the IP id to be deleted ###
Write-Output "DELETE Site IP access/ban Settings For specific IP ###"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ipAccessList/$ipAccessId" -Method 'DELETE' -Headers
$authHeader
Write-Output $siteList | ConvertTo-Json
```

Endpoint: /sites/{siteId}/ipAutoBanList

Returns information about the Auto-banned Ips list

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAutoBanList

GET Body Sample

```
### GET Autoban List###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/ipAutoBanList HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{"data":[{"id":"10.91.160.139","type":"accessBan"}],"links":{"self":"/admin/v2/sites/5ceae6e3-11b1-40c6-b4e4-3078a8e88a35/ipAutoBanList"}}
```

GET PowerShell Sample

```
$siteID = $siteList.data[0].id


####Added 2/01/2020
### Returns all site auto ban list###
Write-Output "Site Auto Ban List"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/IpAutoBanList" -
Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Endpoint: /sites/{siteId}/ipAutoBanList/{ipAutoBanId}

Remove a specific IP from the Auto-banned IP list

- DELETE

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/ipAccessList{ipAutoBanId)

DELETE Body Sample

```
@banId = {{banList.response.body.$.data[0].id}}


### GET Autoban List###
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/ipAutoBanList/{{banId}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content
Date: Thu, 04 Aug 2022 19:43:37 GMT
Content-Length: 0
Cache-Control: no-cache, no-store, must-revalidate
Expires: -1
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
$banID = $banList.data[0].id

$banList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/IpAutoBanList/$banID" -Method 'DELETE' -Headers
$authHeader
Write-Output $banList | ConvertTo-Json
```

Endpoint: /sites/{siteId}/sms

Returns and updates the SMS configuration for the site.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/sms

GET Body Sample

```
### GET SMS Configuration####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/sms HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": [
    {
      "id": "Twilio Generic",
      "type": "smsSettings",
      "attributes": {
        "serviceType": "twilioGeneric",
        "services": {
          "accountSid": "xxxxxxx ...",
          "authToken": "xxxxx ....",
          "twilioNumber": "+1210...",
          "message": "Your passcode is: %Account_Session_OTP%",
          "postUrl": "https://api.twilio.com/2010-04-
01/Accounts/%SID%/Messages"
        },
```

GET PowerShell Sample

```
####Added 2/08/2022 ###
###GET SMS Settings for Site###
Write-Output "GET SMS Settings for Site"
Write-Output "----"
#$siteID = "Enter site ID"
$smsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/sms" -Method
'GET' -Headers $authHeader
Write-Output $smsList | ConvertTo-Json -Depth 25
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/sms

POST Body Sample

```
### POST add an SMS profile###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/sms HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "name": "Test Post SMS",
      "serviceType": "genericSmsProviderProfile",
      "services": {
        "accountSid": "AAAAAAAAAAAAAAAAAAAAAe",
        "authToken": "0dddd0dddd0dddd0dddd",
        "twilioNumber": "+1210210210",
        "message": "Your passcode is: %Account_Session_OTP%",
        "postUrl": "https://api.twilio.com/2010-04-
01/Accounts/%SID%/Messages"
      },
      "proxy": {
        "enabled": true,
        "type": "192.168.100.100",
        "hostName": "testproxy",
        "port": 1443,
        "userName": "",
        "password": ""
      }
    }
```

POST Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated POST request(s).

```
{
  "data": {
    "id": "Test Post SMS",
    "type": "smsSettings",
    "attributes": {
      "name": "Test Post SMS",
      "serviceType": "twilioGeneric",
      "services": {
        "accountSid": "AAAAAAAAAAAAAAAAAAAAAAe",
        "authToken":  "0dddd0dddd0dddd0dddd",
```

POST PowerShell Sample

```
### POST add a new SMS profile ###
### User must obtain the SiteID from Get Sites
Write-Output "add a new SMS profile "
Write-Output "----------------------"
$update =
'{
  "data": {
    "attributes": {
      "name": "PowerShell Example",
      "serviceType": "genericSmsProviderProfile",
      "services": {
        "accountSid": "AAAAAAAAAAAAAAAAAAAAAAe",
        "authToken": "0dddd0dddd0dddd0dddd",
        "twilioNumber": "+1210210210",
        "message": "Your passcode is: %Account_Session_OTP%",
        "postUrl": "https://api.twilio.com/2010-04-01/Accounts/%SID%/Messages"
      },
      "proxy": {
        "enabled": true,
        "type": "192.168.100.100",
        "hostName": "testproxy",
        "port": 1443,
        "userName": "TestUser",
        "password": "example"
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 25
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/sms" -Method
'POST' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteId}/sms/{profileName}

Returns, updates and deletes the SMS configuration for specific profile.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/sms/{profileName}

GET Body Sample

```
### GET SMS Configuration####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/sms/{profileName} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{
  "data": {
    "id": "Test Post SMS",
    "type": "smsSettings",
    "attributes": {
      "name": "Test Post SMS",
      "serviceType": "twilioGeneric",
      "services": {
        "accountSid": "AAAAAAAAAAAAAAAAAAAAAAe",
        "authToken": "0dddd0dddd0dddd0dddd",
        "twilioNumber": "+1210210210",
        "message": "Your passcode is: %Account_Session_OTP%",
        "postUrl": "https://api.twilio.com/2010-04-
```

GET PowerShell Sample

```
####Added 2/08/2022 ###
###GET specific profile SMS Configuration Details###
Write-Output "GET specific profile SMS Configuration Details"
Write-Output "----"
$profileName = "PowerShell Example"
$smsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/sms/$profileName"
-Method 'GET' -Headers $authHeader
Write-Output $smsList | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/sms/{profileName}

PATCH Body Sample

```
### PATCH SMS specific profile ###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/sms/{{profileName}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "name": "",
      "serviceType": "genericSmsProviderProfile",
      "services": {
        "accountSid": "43243243214324234321FDGFDSG",
        "authToken": "1234567899",
        "twilioNumber": "210101010101",
        "message": "Your passcode is: %Account_Session_OTP%",
        "postUrl": "https://api.twilio.com/2010-04-
01/Accounts/%SID%/Messages"
      }
    }
  }
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).

```
{

  "data": {

    "id": "Test Post SMS",

    "type": "smsSettings",

    "attributes": {

      "name": "Test Post SMS",

      "serviceType": "twilioGeneric",

      "services": {

        "accountSid": "43243243214324234321FDGFDSG",

        "authToken": "1234567899",

        "twilioNumber": "210101010101",

        "message": "Your passcode is: %Account_Session_OTP%",

        "postUrl": "https://api.twilio.com/2010-04-
01/Accounts/%SID%/Messages"
```

PATCH PowerShell Sample

```
####Added 2/08/2022
### PATCH - Updates existing SMS profile ###
### User must obtain the SiteID from Get Sites
Write-Output "add a new SMS profile "
Write-Output "----------------------"
$profileName = "PowerShell Example"
$update =
'{
  "data": {
    "attributes": {
      "name": "",
      "serviceType": "genericSmsProviderProfile",
      "services": {
        "accountSid": "43243243214324234321FDGFDSG",
        "authToken": "1234567899",
        "twilioNumber": "210101010101",
        "message": "Your passcode is: %Account_Session_OTP%",
        "postUrl": "https://api.twilio.com/2010-04-01/Accounts/%SID%/Messages"
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 25
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/sms/$profileName" -Method 'PATCH' -Headers
$authHeader -Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/sms/{profileName}

DELETE Body Sample

```
### DELETE SMS specific profile###
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/sms/{{profileName}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
{ "data": [ { "id": "Twilio Generic", "type": "smsSettings",
"attributes": { "serviceType": "twilioGeneric", "services": {
"accountSid": "Aabdd54dgsb2e9357b03a", "authToken":
```

DELETE PowerShell Sample

```
###### DELETE SMS specific profile######
Write-Output "### DELETE SMS specific profile###"
Write-Output "----"
$profileName = "PowerShell Example"
$smsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/sms/$profileName"
-Method 'DELETE' -Headers $authHeader
Write-Output $smsList | ConvertTo-Json -Depth 25
```

Endpoint: /sites/{siteId}/accountSecurity

Returns and updates account security configuration for specific profile

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/sms/accountSecurity

GET Body Sample

```
### GET Account Security Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/accountSecurity/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{

  "data": {

    "type": "siteSecurity",

    "id": "0de7387f-1506-427d-b997-e18bcdb8cbae",

    "attributes": {

      "accountPolicy": {

        "invalidLogin": {

          "ip": {

            "enabled": true,

            "attemptPeriodMinutes": 1,

            "attemptRetries": 5,
```

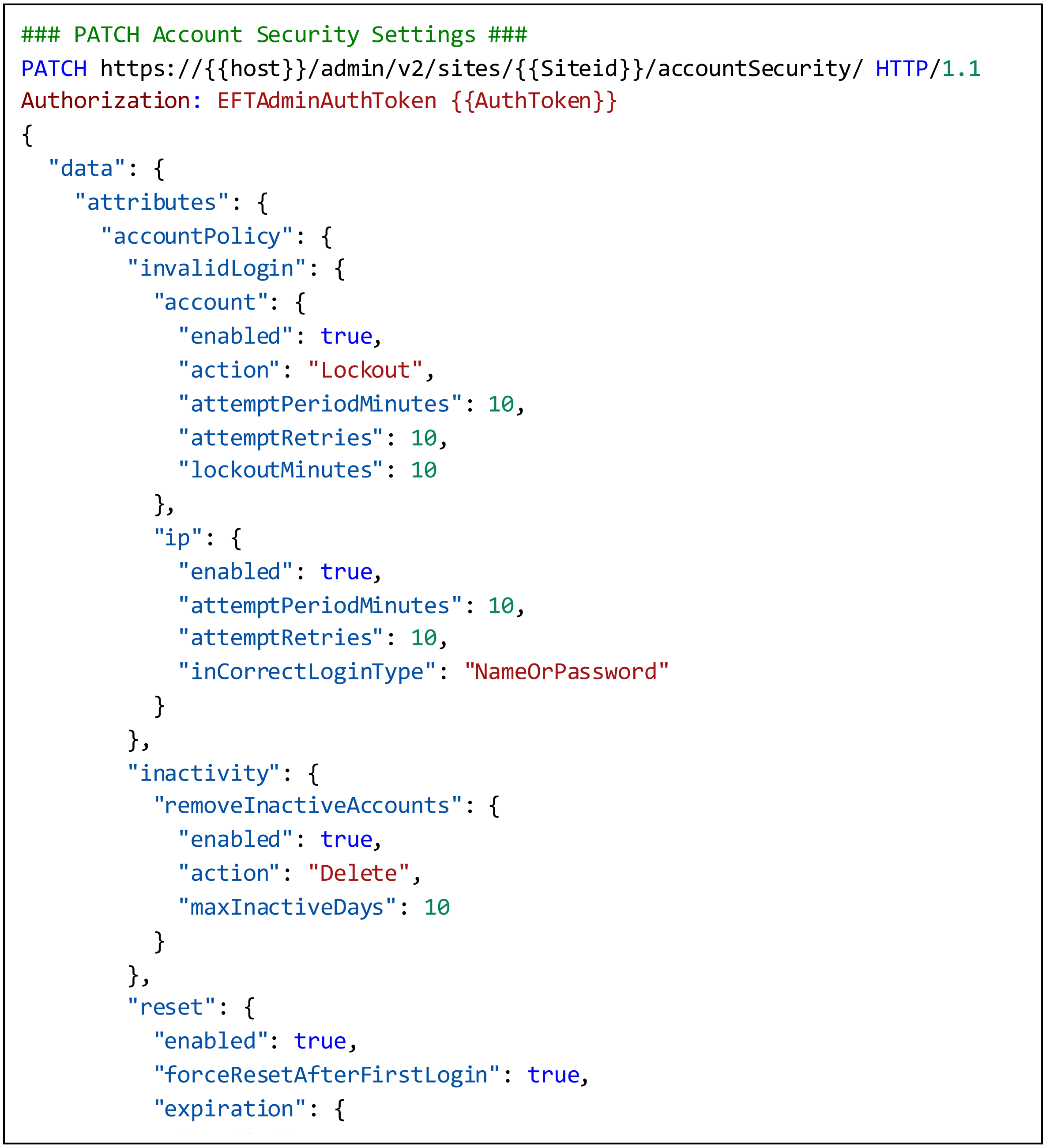
GET PowerShell Sample

```
####Added 2/09/2022 ###
###  GET Account Security Configuration Details ###
Write-Output "GET Account Security Configuration Details"
Write-Output "----"
$profileName = "PowerShell Example"
$smsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/accountSecurity"
-Method 'GET' -Headers $authHeader
Write-Output $smsList | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/sms/accountSecurity

PATCH Body Sample



```
### PATCH Account Security Settings ###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/accountSecurity/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
  "data": {
    "attributes": {
      "accountPolicy": {
        "invalidLogin": {
          "account": {
            "enabled": true,
            "action": "Lockout",
            "attemptPeriodMinutes": 10,
            "attemptRetries": 10,
            "lockoutMinutes": 10
          },
          "ip": {
            "enabled": true,
            "attemptPeriodMinutes": 10,
            "attemptRetries": 10,
            "inCorrectLoginType": "NameOrPassword"
          }
        },
        "inactivity": {
          "removeInactiveAccounts": {
            "enabled": true,
            "action": "Delete",
            "maxInactiveDays": 10
          }
        },
        "reset": {
          "enabled": true,
          "forceResetAfterFirstLogin": true,
          "expiration": {
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated Patch request(s).

```
{ "data": { "type": "site", "id": "0de7387f-1506-427d-b997-e18bcdb8cbae",
```

PATCH PowerShell Sample

```
### PATCH Account Security Configuration Details ###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH Account Security Configuration Details"
Write-Output "----------------------"
$update =
'{
  "data": {
    "attributes": {
      "accountPolicy": {
        "invalidLogin": {
          "account": {
            "enabled": true,
            "action": "Disable",
            "attemptPeriodMinutes": 3,
            "attemptRetries": 3,
            "lockoutMinutes": 3
```

Endpoint: /sites/{ siteId}/workspaces

Returns and updates workspaces settings

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/workspaces/

GET Body Sample

```
### GET Workspaces - Folder Sharing Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/workspaces/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
"data": { "type": "workspaces", "id": "0de7387f-1506-427d-b997-
e18bcdb8cbae", "attributes": { "enabled": true, "newUser": { "enabled":
true, "allowedDomains": { "enabled": false, "list": [ "*.*" ] },
"deniedDomains": { "enabled": false, "list": [ "*.*" ] } },
"maxExpiration": { "value": 1, "units": "Year" }, "secondVerification":
"email", "guestAccounts": { "expiration": { "enabled": true, "action":
"DeleteAccountOnly", "daysAfterLinksExpiration": 7 }, "grantHomeFolder":
true } } }, "links": { "self": "admin/v2/sites/0de7387f-1506-427d-b997-
e18bcdb8cbae/workspaces" } }
```

GET PowerShell Sample

```
###GET workspaces Configuration Details###
Write-Output "GET workspaces Configuration Details"
Write-Output "----"
#$siteID = "Enter site ID"
$wksList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/workspaces" -
Method 'GET' -Headers $authHeader
Write-Output $wksList | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/workspaces

PATCH Body Sample

```
### PATCH Workspaces - Folder Sharing Setting ###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/workspaces/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "enabled": true,
      "newUsers": {
        "enabled": true,
        "allowedDomains": {
          "enabled": false
        }
      }
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated Patch request(s).

```
HTTP/1.1 200 OK Date: Thu, 25 Aug 2022 19:20:16 GMT Content-Type:
application/vnd.api+json ETag: "FC63C399" Content-Length: 507 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "type": "workspaces", "id": "c9551b5d-f904-4612-
b690-d51bfde2eb1f", "attributes": { "enabled": true, "newUsers": {
"enabled": true, "allowedDomains": { "enabled": false, "list": [ "*.*" ]
}, "deniedDomains": { "enabled": false, "list": [ "*.*" ] } },
"maxExpiration": { "value": 1, "units": "Year" }, "secondVerification":
"email", "guestAccounts": { "expiration": { "enabled": false, "action":
"Disable", "daysAfterLinksExpiration": 7 }, "grantHomeFolder": false } }
}, "links": { "self": "/admin/v2/sites/c9551b5d-f904-4612-b690-
d51bfde2eb1f/workspaces" } }
```

PATCH PowerShell Sample

```
##Patch SITE Workspaces settings
Write-Output "Site Workspaces PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes": {
      "enabled": true,
      "newUsers": {
        "enabled": true,
        "allowedDomains": {
          "enabled": false
        }
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchWSReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/workspaces"
-Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $patchWSReturn | ConvertTo-Json
```

Endpoint: /sites/{siteId}/sendPortal

Returns and updates send portal configuration for specific site

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/sendPortal

GET Body Sample

```
### GET Send Portal Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/sendPortal/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"sendPortal", "attributes": { "enabled": true, "recipients":
"UsersGuestsAnonymous", "allowedDomains": { "enabled": false, "list": [
"*.*" ] }, "deniedDomains": { "enabled": false, "list": [ "*.*" ] },
"reservedPath": "/send", "protocol": "https", "hostName":
"172.31.223.227", "port": "443", "autoAttachInOutlook": { "enabled":
```

GET PowerShell Sample

```
###GET Send Portal Configuration Details###
Write-Output "GET Send Portal Configuration Details"
Write-Output "-----------------------------------"
#$siteID = "Enter site ID"
$SendList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/sendPortal" -
Method 'GET' -Headers $authHeader
Write-Output $Sendlist | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/sendPortal

PATCH Body Sample

```
### PATCH Send Portal Settings ###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/sendPortal/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "sendPortal",
    "attributes": {
      "enabled": true,
      "recipients": "UsersGuestsAnonymous",
      "allowedDomains": {
        "enabled": true,
        "list": [
          "hotmail.*", "gmail.*"
        ]
      },
      "deniedDomains": {
        "enabled": true,
        "list": [
          "GSB.*"
        ]
      },
      "reservedPath": "/send",
      "protocol": "https",
      "hostName": "172.31.223.227",
      "port": "443",
      "autoAttachInOutlook": {
        "enabled": false,
        "whenExceedMbytes": 10
      },
      "linkExpiration": {
        "value": 1,
        "units": "Month"
      },
      "retainFilesAfterLinkExpiration": {
        "value": 0,
        "unit": "Day"
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated Patch request(s).

```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"sendPortal", "attributes": { "enabled": true, "recipients":
"UsersGuestsAnonymous", "allowedDomains": { "enabled": false, "list": [
"hotmail.*", "gmail.*" ] }, "deniedDomains": { "enabled": false, "list":
[ "GSB.*" ] }, "reservedPath": "/send", "protocol": "http", "hostName":
"", "port": "113", "autoAttachInOutlook": { "enabled": false,
"whenExceedMbytes": 10 }, "linkExpiration": { "value": 1, "units":
"Month" }, "retainFilesAfterLinkExpiration": { "value": 0, "unit": "Day"
}, "allowRepliesWithFiles": true, "sendEntireMessageSecurely":
"LetUserChoose", "outOfBandPasscode": "Sender chooses",
"secondVerification": "email", "fileRequestPortal": { "enabled": false,
"reservedPath": "/request" } } }, "links": { "self":
"/admin/v2/sites/0de7387f-1506-427d-b997-e18bcdb8cbae/sendPortal" } }
```

PATCH PowerShell Sample

```
Write-Output "PATCH Send Portal Configuration Details"
Write-Output "----------------------------------------"
$update =
'{
  "data": {
    "type": "sendPortal",
    "attributes": {
      "enabled": true,
      "recipients": "UsersGuestsAnonymous",
      "allowedDomains": {
        "enabled": true,
        "list": [
          "outlook.*", "yahoo.*"
        ]
      },
      "deniedDomains": {
        "enabled": true,
        "list": [
          "GSB.*"
        ]
```

Endpoint: /sites/{ siteId}/dropOffPortal

Returns and updates drop off portal configuration for specific site

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/dropOffPortal

GET Body Sample

```
## GET Drop Off Portal Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/dropOffPortal/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "id": "df3be28b-f499-4b0a-9c53-7cee33a58161",
"type": "dropOffPortal", "attributes": { "enabled": true,
"reservedPath": "/dropoff", "captcha": { "type": "reCAPTCHA",
"value": { "siteKey": "6LeXY8ofdsfdsfdsfasq8RdygpeBeclHqrVBUd-
afar", "secretKey": "6LeXY8oUAAAAABIVq8RdygpeBeclHqrVBUd-afar" }
}, "linkExpiration": { "value": 1, "units": "Week" },
"maxMessageSize": { "enabled": true, "mBytes": 2048 },
"sendEntireMessageSecurely": "LetUserChoose", "userEmailOption":
{ "allowToEnterToEmailAddress": true, "allowedDomains": {
"enabled": true, "list": [ "*.*" ] } }, "addressBookList": [
"user2<ivan@mail.com>", "User1<elver@mail.com>" ] } }, "links":
{ "self": "admin/v2/sites/df3be28b-f499-4b0a-9c53-
```

GET PowerShell Sample

```
####Added 2/10/2022 ###
###GET Drop Off Portal Settings###
Write-Output "GET Drop Off Portal Settings"
Write-Output "----------------------------"
#$siteID = "Enter site ID"
$DropOfflist = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/dropOffPortal" -Method 'GET' -Headers $authHeader
Write-Output $DropOfflist | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/dropOffPortal

PATCH Body Sample

```
### PATCH Drop Off Portal Settings###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/dropOffPortal/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "dropOffPortal",
    "attributes": {
      "enabled": true,
      "reservedPath": "/dropoff",
      "captcha": {
        "type": "reCAPTCHA",
        "value": {
          "siteKey": "6LeXY8odfasdfdsafdsafdsfsdfdsfdsfsdfdsfds-afar",
          "secretKey": "6Ldfasdfdsafdsafsdafdsafsdasfdsafdsfsd-afar"
        }
      },
      "linkExpiration": {
        "value": 1,
        "units": "Week"
      },
      "maxMessageSize": {
        "enabled": true,
        "mBytes": 2048
      },
      "sendEntireMessageSecurely": "LetUserChoose",
      "userEmailOption": {
        "allowToEnterToEmailAddress": true,
        "allowedDomains": {
          "enabled": true,
          "list": [
            "*.*"
          ]
        }
      },
      "addressBookList": ["user2<user2@mail.com>",
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated Patch request(s).

```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"dropOffPortal", "attributes": { "enabled": true, "reservedPath":
"/dropoff", "captcha": { "type": "reCAPTCHA", "value": { "siteKey":
"6LeXY8oUAAAAABIVq8RdygpeBeclHqrVBUd-afar", "secretKey":
"6LeXY8oUAAAAABIVq8RdygpeBeclHqrVBUd-afar" } }, "linkExpiration": {
"value": 1, "units": "Week" }, "maxMessageSize": { "enabled": true,
"mBytes": 2048 }, "sendEntireMessageSecurely": "LetUserChoose",
"userEmailOption": { "allowToEnterToEmailAddress": true,
"allowedDomains": { "enabled": true, "list": [ "*.*" ] } },
"addressBookList": [] } }, "links": { "self": "admin/v2/sites/0de7387f-
```

PATCH PowerShell Sample

```
Write-Output "PATCH Drop Off Portal Settings"
Write-Output "------------------------------"
$update =
'{
  "data": {
    "type": "dropOffPortal",
    "attributes": {
      "enabled": true,
      "reservedPath": "/dropoff",
      "captcha": {
        "type": "reCAPTCHA",
        "value": {
          "siteKey": "6LeXY8odfasdfdsafdsafdsfsdfdsfdsfsdfdsfds-afar",
          "secretKey": "6Ldfasdfdsafdsafsdafdsafsdasfdsafdsfsd-afar"
```

Endpoint: /sites/{siteId}/privacyPolicy

Returns and updates the privacy policy configuration for specific site

- GET
- PATCH

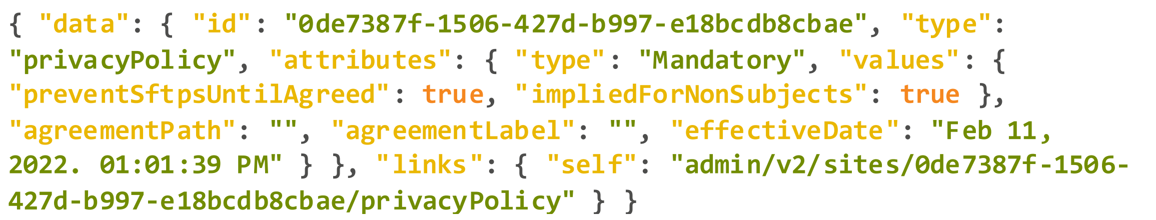
Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/privacyPolicy

GET Body Sample

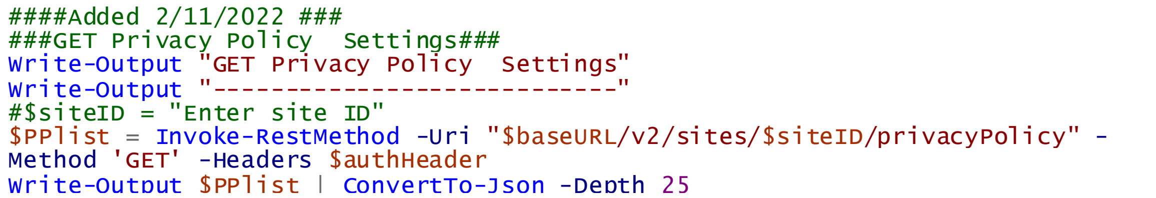
```
### GET Privacy Policy  Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/privacyPolicy/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample



```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"privacyPolicy", "attributes": { "type": "Mandatory", "values": {
"preventSftpsUntilAgreed": true, "impliedForNonSubjects": true },
"agreementPath": "", "agreementLabel": "", "effectiveDate": "Feb 11,
2022. 01:01:39 PM" } }, "links": { "self": "admin/v2/sites/0de7387f-1506-
427d-b997-e18bcdb8cbae/privacyPolicy" } }
```

GET PowerShell Sample



```
####Added 2/11/2022 ###
###GET Privacy Policy  Settings###
Write-Output "GET Privacy Policy  Settings"
Write-Output "----------------------------"
#$siteID = "Enter site ID"
$PPlist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/privacyPolicy" -
Method 'GET' -Headers $authHeader
Write-Output $PPlist | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/privacyPolicy

PATCH Body Sample

```
### PATCH Privacy Policy  Settings###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/privacyPolicy/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "privacyPolicy",
    "attributes": {
      "type": "Mandatory",
      "values": {
        "preventSftpsUntilAgreed": false,
        "impliedForNonSubjects": false
      },
      "agreementPath": "",
      "agreementLabel": "",
      "effectiveDate": "Feb 11, 2020. 01:01:39 PM"
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated Patch request(s).

```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"privacyPolicy", "attributes": { "type": "Mandatory", "values": {
"preventSftpsUntilAgreed": false, "impliedForNonSubjects": false },
"agreementPath": "", "agreementLabel": "", "effectiveDate": "Feb 11,
2022. 01:01:39 PM" } }, "links": { "self": "/admin/v2/sites/0de7387f-
1506-427d-b997-e18bcdb8cbae/privacyPolicy" } }
```

PATCH PowerShell Sample

```
### PATCH Privacy Policy  Settings###
### User must obtain the SiteID from Get Sites
Write-Output "Privacy Policy  Settings"
Write-Output "------------------------------"
$update =
'{
  "data": {
    "type": "privacyPolicy",
    "attributes": {
      "type": "Mandatory",
      "values": {
        "preventSftpsUntilAgreed": true,
        "impliedForNonSubjects": true
      },
      "agreementPath": "",
      "agreementLabel": "",
      "effectiveDate": "Feb 11, 2020. 01:01:39 PM"
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 30
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/privacyPolicy" -Method 'PATCH' -Headers $authHeader
-Body $update
```

Endpoint: /sites/{siteId}/termsOfService

Returns and updates the terms of service configuration for specific site

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/termsOfService

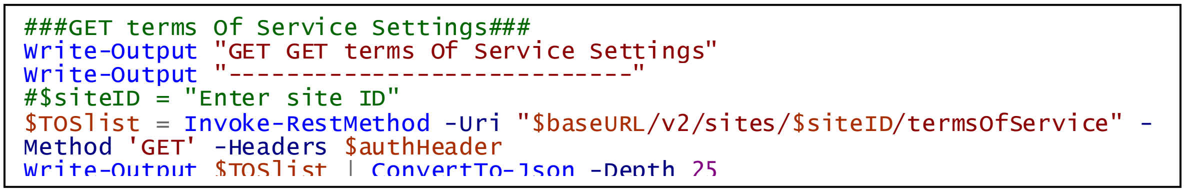
GET Body Sample

```
### GET Terms of Services Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/termsOfService/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"termsOfService", "attributes": { "type": "Disabled", "agreementPath":
"", "agreementLabel": "", "effectiveDate": "Dec 16, 1950. 10:05:17 AM" }
}, "links": { "self": "admin/v2/sites/0de7387f-1506-427d-b997-
e18bcdb8cbae/termsOfService" } }
```

GET PowerShell Sample



```
###GET terms Of Service Settings###
Write-Output "GET GET terms Of Service Settings"
Write-Output "----------------------------"
#$siteID = "Enter site ID"
$TOSlist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/termsOfService" -
Method 'GET' -Headers $authHeader
Write-Output $TOSlist | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/termsOfService

PATCH Body Sample

```
### PATCH Terms of Services Settings###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/termsOfService/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "termsOfService",
    "attributes": {
      "type": "Mandatory",
      "values": {
        "preventSftpsUntilAgreed": true
      },
      "agreementPath": "",
      "agreementLabel": "",
      "effectiveDate": "Dec 16, 2000. 10:05:17 AM"
    }
  }
}
```

PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated Patch request(s).

```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type":
"termsOfService", "attributes": { "type": "Mandatory", "values": {
"preventSftpsUntilAgreed": true }, "agreementPath": "", "agreementLabel":
"", "effectiveDate": "Feb 11, 2022. 01:01:53 PM" } }, "links": { "self":
"admin/v2/sites/0de7387f-1506-427d-b997-e18bcdb8cbae/termsOfService" } }
```

PATCH PowerShell Sample

```
####Added 2/11/2022
### PATCH terms Of Service Settings###
### User must obtain the SiteID from Get Sites
Write-Output "terms Of Service Settings"
Write-Output "-------------------------"
$update =
'{
  "data": {
    "type": "termsOfService",
    "attributes": {
      "type": "Mandatory",
      "values": {
        "preventSftpsUntilAgreed": true
      },
      "agreementPath": "",
      "agreementLabel": "",
      "effectiveDate": "Dec 16, 2000. 10:05:17 AM"
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 30
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/termsOfService" -Method 'PATCH' -Headers
$authHeader -Body $update
```

Endpoint: /sites/{siteId}/gdpr

Returns and updates the GDPR configuration for specific profile

- GET
- POST

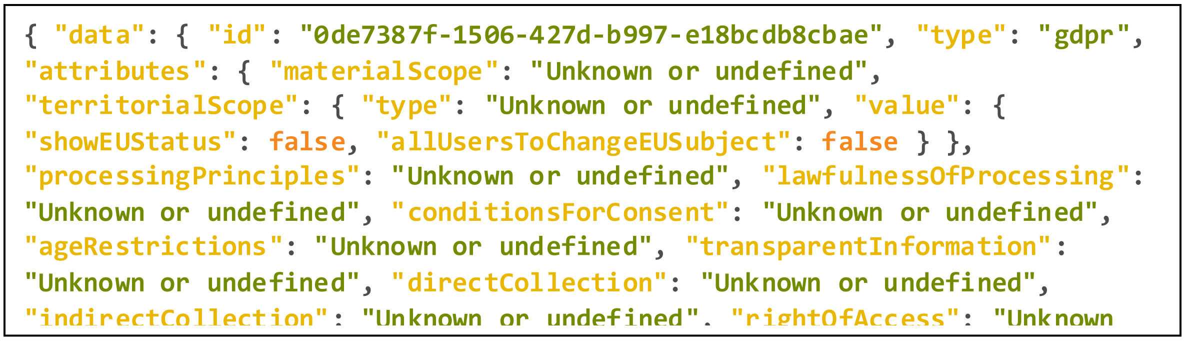
Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/gdpr

GET Body Sample

```
### GET the GDRP Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/gdpr/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample



```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type": "gdpr",
"attributes": { "materialScope": "Unknown or undefined",
"territorialScope": { "type": "Unknown or undefined", "value": {
"showEUStatus": false, "allUsersToChangeEUSubject": false } },
"processingPrinciples": "Unknown or undefined", "lawfulnessOfProcessing":
"Unknown or undefined", "conditionsForConsent": "Unknown or undefined",
"ageRestrictions": "Unknown or undefined", "transparentInformation":
"Unknown or undefined", "directCollection": "Unknown or undefined",
"indirectCollection": "Unknown or undefined", "rightOfAccess": "Unknown
```

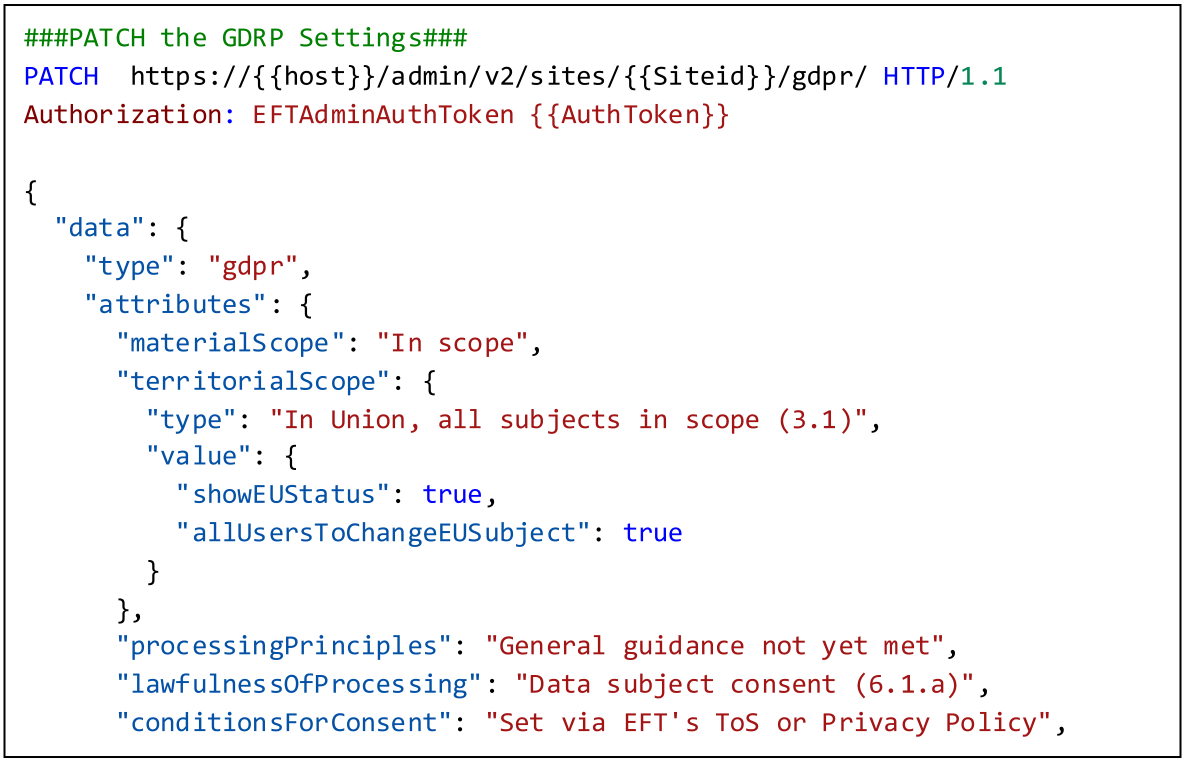
GET PowerShell Sample

```
###GET GDRP Settings###
Write-Output "GET GDRP Settings"
Write-Output "-----------------"
#$siteID = "Enter site ID"
$GDPRlist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/gdpr" -Method
'GET' -Headers $authHeader
Write-Output $GDPRlist | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/gdpr

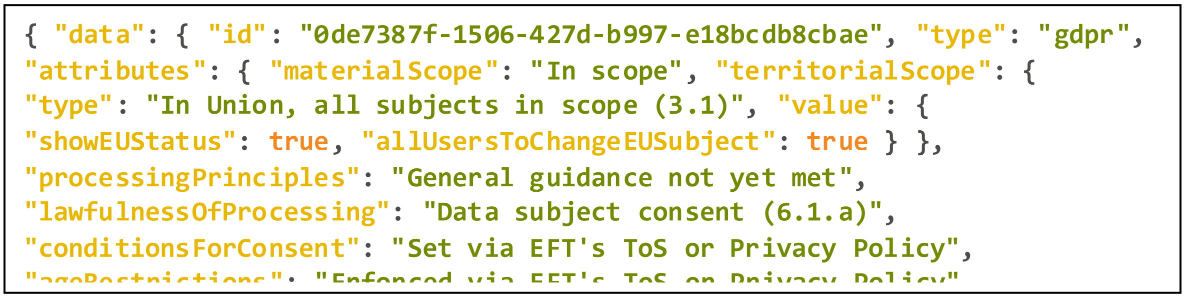
PATCH Body Sample



```
###PATCH the GDRP Settings###
PATCH  https://{{host}}/admin/v2/sites/{{Siteid}}/gdpr/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "gdpr",
    "attributes": {
      "materialScope": "In scope",
      "territorialScope": {
        "type": "In Union, all subjects in scope (3.1)",
        "value": {
          "showEUStatus": true,
          "allUsersToChangeEUSubject": true
        }
      },
      "processingPrinciples": "General guidance not yet met",
      "lawfulnessOfProcessing": "Data subject consent (6.1.a)",
      "conditionsForConsent": "Set via EFT's ToS or Privacy Policy",
```

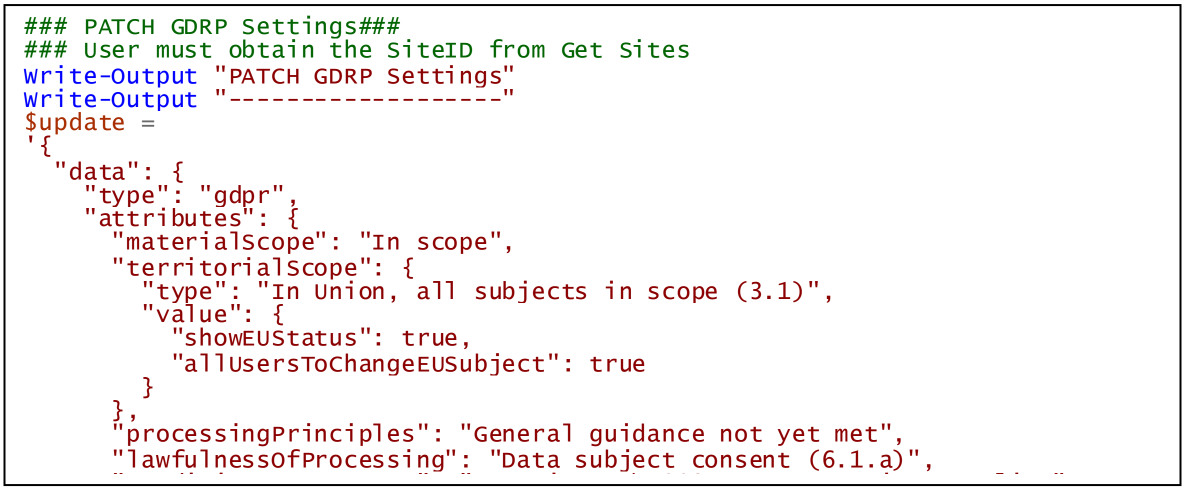
PATCH Response Body Sample

The response will return the same response as the GET request, however, it will now include the updated PATCH request(s).



```
{ "data": { "id": "0de7387f-1506-427d-b997-e18bcdb8cbae", "type": "gdpr",
"attributes": { "materialScope": "In scope", "territorialScope": {
"type": "In Union, all subjects in scope (3.1)", "value": {
"showEUStatus": true, "allUsersToChangeEUSubject": true } },
"processingPrinciples": "General guidance not yet met",
"lawfulnessOfProcessing": "Data subject consent (6.1.a)",
"conditionsForConsent": "Set via EFT's ToS or Privacy Policy",
"ageRestrictions": "Enforced via EFT's ToS or Privacy Policy",
```

PATCH PowerShell Sample



```
### PATCH GDRP Settings###
### User must obtain the SiteID from Get Sites
Write-Output "PATCH GDRP Settings"
Write-Output "--------------------"
$update =
'{
  "data": {
    "type": "gdpr",
    "attributes": {
      "materialScope": "In scope",
      "territorialScope": {
        "type": "In Union, all subjects in scope (3.1)",
        "value": {
          "showEUStatus": true,
          "allUsersToChangeEUSubject": true
        }
      },
      "processingPrinciples": "General guidance not yet met",
      "lawfulnessOfProcessing": "Data subject consent (6.1.a)",
```

Endpoint: /sites/{siteId}/uploadForms

Returns and updates the upload forms configuration for specific profile

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/uploadForms

GET Body Sample

```
### GET the Upload Forms Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/uploadForms/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": [ { "type": "site", "id": "0de7387f-1506-427d-b997-
e18bcdb8cbae", "attributes": { "name": "MySite1" }, "links": { "self":
"admin/v2/sites/0de7387f-1506-427d-b997-e18bcdb8cbae", "metrics":
"admin/v2/sites/0de7387f-1506-427d-b997-e18bcdb8cbae/metrics" } } ],
```

GET PowerShell Sample

```
###GET Upload Forms Settings###
Write-Output "GET Upload Forms Settings"
Write-Output "-----------------"
#$siteID = "Enter site ID"
$UpFormslist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/uploadForms"
-Method 'GET' -Headers $authHeader
Write-Output $UpFormslist | ConvertTo-Json -Depth 25
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/

POST Body Sample

```
###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/uploadForms/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "uploadFormName",
    "attributes": {
      "description": "Testing Forms from POST REST API",
      "displayName": "FORM_1",
      "displayInstruction": "Enter Text",
      "enabled": true,
      "showForEachFileUploaded": true,
      "userGroupId": "A0DB9377-92A3-4A38-8C0F-09F268D78FF1",
      "uploadFormElements": [
        {
          "name": "",
          "displayLabel": "FORM_1",
          "type": "string",
          "values": "domain.com",
          "required": true
        }
      ]
    }
  }
}
```

POST Response Body Sample

```
{"data":{"type":"uploadForm","id":"uploadFormName","attributes":{"id":"up
loadFormName","type":"uploadForm","attributes":{"description":"Testing
Forms from POST REST
API","displayName":"FORM_1","displayInstruction":"Enter
Text","enabled":true,"showForEachFileUploaded":true,"userGroupId":"a0db93
77-92a3-4a38-8c0f-
09f268d78ff1","uploadFormElements":[{"name":"","displayLabel":"FORM_1","t
ype":"String","values":"domain.com","required":true}]}}},"links":{"self":
"/admin/v2/sites/6203305a-d7c6-499e-a59b-
81500765f8f2/uploadForm/uploadFormName","userGroups":"/admin/v2/sites/620
```

POST PowerShell Sample

```
$update =
'{
  "data": {
    "id": "uploadFormName",
    "attributes": {
      "description": "Testing Forms from POST REST API",
      "displayName": "FORM_once",
      "displayInstruction": "Enter Text",
      "enabled": true,
      "showForEachFileUploaded": true,
      "userGroupId": "A0DB9377-92A3-4A38-8C0F-09F268D78FF1",
      "uploadFormElements": [
        {
          "name": "",
          "displayLabel": "FORM_once",
          "type": "string",
          "values": "domain.com",
          "required": true
        }
      ]
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$postUploadFormsReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/uploadForms" -Method 'POST' -Headers $authHeader -
Body $update
Write-Output $postUploadFormsReturn | ConvertTo-Json
```

Endpoint: /sites/{siteId}/uploadForms/{uploadFormName}

Returns site upload settings. Update site upload forms settings. Deletes upload form.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/uploadForms/{uploadFormName}

GET Body Sample

```
### GET Specific Upload Forms Settings####
### Enter the form ID

@upLoadFormsID = {{formsList.response.body.$.data[0].id}}

GET https://{{host}}/admin/v2/sites/{{Siteid}}/uploadForms/{{upLoadFormsID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":{"id":"uploadFormName","type":"uploadForm","attributes":{"description":"Testing Forms from POST REST API","displayName":"FORM_1","displayInstruction":"Enter Text","enabled":true,"showForEachFileUploaded":true,"userGroupId":"a0db9377-92a3-4a38-8c0f-09f268d78ff1","uploadFormElements":[{"name":"","displayLabel":"FORM_1","type":"String","values":"domain.com","required":true}]}},"links":{"self":"/admin/v2/sites/6203305a-d7c6-499e-a59b-81500765f8f2/uploadForm/uploadFormName","userGroups":"/admin/v2/sites/6203305a-d7c6-499e-a59b-81500765f8f2/userGroups"}}
```

GET PowerShell Sample

```
###GET Upload Forms Settings###
Write-Output "GET Upload Forms Settings"
Write-Output "-----------------"
#$siteID = "Enter site ID"
$UpFormslist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/uploadForms" -Method 'GET' -Headers $authHeader
Write-Output $UpFormslist | ConvertTo-Json -Depth 25
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/uploadForms/{uploadFormName}

PATCH Body Sample

```
###
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/uploadForms/{{upLoadFormsID
}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "uploadFormName",
    "type": "uploadForm",
    "attributes": {
      "description": "Testing Forms from POST REST API",
      "displayName": "FORM_1",
      "displayInstruction": "Enter Text",
      "enabled": true,
      "showForEachFileUploaded": false,
      "userGroupId": "a0db9377-92a3-4a38-8c0f-09f268d78ff1",
      "uploadFormElements": [
        {
          "name": "",
          "displayLabel": "FORM_1",
          "type": "String",
          "values": "domain.com",
          "required": true
        }
      ]
    }
  }
}
```

POST Response Body Sample

```
{"data":{"id":"uploadFormName","type":"uploadForm","attributes":{"desc
ription":"Testing Forms from POST REST
API","displayName":"FORM_1","displayInstruction":"Enter
Text","enabled":true,"showForEachFileUploaded":false,"userGroupId":"a0
db9377-92a3-4a38-8c0f-
09f268d78ff1","uploadFormElements":[{"name":"","displayLabel":"FORM_1"
,"type":"String","values":"domain.com","required":true}]}},"links":{"s
elf":"/admin/v2/sites/6203305a-d7c6-499e-a59b-
81500765f8f2/uploadForms/uploadFormName","userGroups":"/admin/v2/sites
/6203305a-d7c6-499e-a59b-81500765f8f2/userGroups"}}
```

PATCH PowerShell Sample

```
$update =
'{
  "data": {
    "id": "uploadFormName",
    "type": "uploadForm",
    "attributes": {
      "description": "Testing Forms from POST REST API",
      "displayName": "FORM_1",
      "displayInstruction": "Enter New Text",
      "enabled": true,
      "showForEachFileUploaded": false,
      "userGroupId": "a0db9377-92a3-4a38-8c0f-09f268d78ff1",
      "uploadFormElements": [
        {
          "name": "",
          "displayLabel": "FORM_1",
          "type": "String",
          "values": "domain.com",
          "required": true
        }
      ]
    }
  }
}
'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchUploadFormsReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/uploadForms/$formsID" -Method 'PATCH' -Headers
$authHeader -Body $update
Write-Output $patchUploadFormsReturn | ConvertTo-Json
```

DELETE Body Sample

```
### GET Specific Upload Forms Settings####
### Enter the form ID

@upLoadFormsID = {{formsList.response.body.$.data[0].id}}

DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/uploadForms/{{upLoadFormsID
}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content
Date: Mon, 08 Aug 2022 20:43:09 GMT
Content-Length: 0
Cache-Control: no-cache, no-store, must-revalidate
Expires: -1
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-
eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
$formsID = $uploadFList.data[0].id

$deleteUploadFList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/uploadForms/$formsID" -Method 'DELETE' -Headers
$authHeader
Write-Output $deleteUploadFList | ConvertTo-Json
```

Endpoint: /sites/{siteId}/userGroups

Returns the user groups for specific site

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/userGroups

GET Body Sample

```
### GET the User groups Settings####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/userGroups HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": [ { "attributes": { "name": "All Users" }, "id": "A0DB9377-
92A3-4A38-8C0F-09F268D78FF1", "type": "userGroup" }, { "attributes": {
"name": "Administrators" }, "id": "07CACEB2-6698-4EF7-BAEB-F437FEC972A2",
"type": "userGroup" }, { "attributes": { "name": "Guests" }, "id":
```

GET PowerShell Sample

```
###GET User Groups###
Write-Output "GET User Groups"
Write-Output "-----------------"
#$siteID = "Enter site ID"
$Uglist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/userGroups" -
Method 'GET' -Headers $authHeader
Write-Output $Uglist | ConvertTo-Json -Depth 25
```

Endpoint: /sites/{siteId}/outlookDeploymentScript

Returns the outlook deployment script.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/outlookDeploymentScript

GET Body Sample

```
## GET outlook Deployment Script####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/outlookDeploymentScript
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
msiexec /lv installation.log /qn /i EFTOutlookAddinInstaller.msi
EFTHOST=http://172.31.223.227 EFTPORT=443 EFTAUTHTYPE=MANUAL
EFTUSERNAME=username EFTPASSWORD=test123!
INSTALLLOCATION="%userprofile%\AppData\Roaming\GlobalSCAPE\EFT Outlook
Add-in" FILESTORELOCATION="%userprofile%\AppData\Local\GlobalSCAPE\EFT
Outlook Add-in"
```

GET PowerShell Sample

```
###GET outlook Deployment Script###
Write-Output "GET outlook Deployment Script"
Write-Output "-----------------"
#$siteID = "Enter site ID"
$ODSlist = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/outlookDeploymentScript" -Method 'GET' -Headers
$authHeader
Write-Output $ODSlist | ConvertTo-Json -Depth 25
```

Endpoint: /sites/{siteId}/contentIntegrity

Returns and updates the content integrity settings for specific site.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/contentIntegrity

GET Body Sample

```
### GET a list of current content integrity profiles####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/contentIntegrity HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": [ { "id": "6f61c6c0-221e-4f01-94d0-46ba78ec0548", "type":
"contentIntegrity", "attributes": { "profileName": "TEST API" } }, {
"id": "03bebeda-1a89-40d2-a8c8-78ec078257db", "type": "contentIntegrity",
"attributes": { "profileName": "TEST API 2" } } ], "links": { "self":
"admin/v2/sites/0de7387f-1506-427d-b997-e18bcdb8cbae/contentIntegrity" }
```

GET PowerShell Sample

```
###GET list of Content Integrity Profiles###
Write-Output "GET list of Content Integrity Profiles"
Write-Output "--------------------------------------"
#$siteID = "Enter site ID"
$CIlist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/contentIntegrity"
-Method 'GET' -Headers $authHeader
Write-Output $CIlist | ConvertTo-Json -Depth 25
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/contentIntegrity

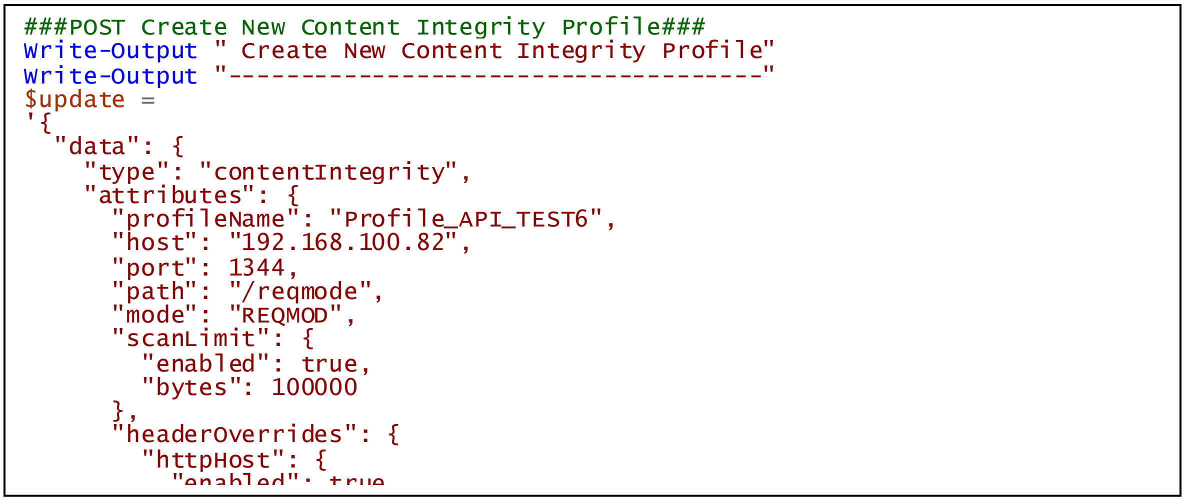
POST Body Sample

```
### #POST content integrity profile ####
POST https://{{host}}/admin/v2/sites/{{Siteid}}/contentIntegrity/
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
  "data": {
    "type": "contentIntegrity",
    "attributes": {
      "profileName": "Profile_API_TEST5",
      "host": "192.168.100.82",
      "port": 1344,
      "path": "/reqmode",
      "mode": "REQMOD",
      "scanLimit": {
        "enabled": true,
        "bytes": 100000
      },
      "headerOverrides": {
        "httpHost": {
          "enabled": true,
          "value": "192.168.100.83"
        },
        "xClientIp": {
          "enabled": true,
          "value": "192.168.100.52"
        },
        "xServerIp": {
          "enabled": true,
          "value": "192.168.100.60"
        },
        "xSubscriberIp": {
          "enabled": true,
          "value": "192.168.100.70"
        },
        "xAuthenticatedUser": {
          "enabled": true,
```

POST Response Body Sample

```
{ "data": { "id": "53f00be5-9aa0-41ed-9fdf-ea99f6784238", "type":
"contentIntegrity", "attributes": { "profileName": "Profile_API_TEST",
"host": true, "port": 1344, "path": true, "mode": "REQMOD", "scanLimit":
{ "enabled": true, "bytes": 100000 }, "headerOverrides": { "httpHost": {
"enabled": true, "value": "192.168.100.83" }, "xClientIp": { "enabled":
true, "value": "192.168.100.52" }, "xServerIp": { "enabled": false,
```

POST PowerShell Sample



```
###POST Create New Content Integrity Profile###
Write-Output " Create New Content Integrity Profile"
Write-Output "-------------------------------------"
$update =
'{
  "data": {
    "type": "contentIntegrity",
    "attributes": {
      "profileName": "Profile_API_TEST6",
      "host": "192.168.100.82",
      "port": 1344,
      "path": "/reqmode",
      "mode": "REQMOD",
      "scanLimit": {
        "enabled": true,
        "bytes": 100000
      },
      "headerOverrides": {
        "httpHost": {
          "enabled": true
```

Endpoint: /sites/{siteId}/contentIntegrity/{profileId}

Returns, updates, and deletes content integrity profile settings.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/contentIntegrity/{profileId}

GET Body Sample

```
### GET spacific content integrity profile settings####
###Pass the ID of the CIC profile###
@profileId = 53f00be5-9aa0-41ed-9fdf-ea99f6784238
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/contentIntegrity/{{profileId}}
/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

Or

### GET spacific content integrity profile settings####
###Pass the ID of the CIC profile from another variable###
###@profileId = 53f00be5-9aa0-41ed-9fdf-ea99f6784238
```

GET Response Body Sample

```
{ "data": { "id": "53f00be5-9aa0-41ed-9fdf-ea99f6784238", "type":
"contentIntegrity", "attributes": { "profileName": "Profile_API_TEST",
"host": true, "port": 1344, "path": true, "mode": "REQMOD", "scanLimit":
{ "enabled": true, "bytes": 100000 }, "headerOverrides": { "httpHost": {
"enabled": true, "value": "192.168.100.83" }, "xClientIp": { "enabled":
true, "value": "192.168.100.52" }, "xServerIp": { "enabled": false,
"value": "" }, "xSubscriberIp": { "enabled": false, "value": "" },
"xAuthenticatedUser": { "enabled": false, "value": "" },
"xAuthenticatedGroups": { "enabled": false, "value": "" } },
"responseHandling": { "continueOnConnectivityError": "Continue",
"continueOnHttpError": "Continue", "continueOnIcapViolation": "Fail",
"continueOnIcapRedaction": "Continue", "customHeadersToAudit": {
"enabled": true, "headers": [] } } } }, "links": { "self":
"admin/v2/sites/0de7387f-1506-427d-b997-
```

GET PowerShell Sample

```
####Added 2/15/2022 ###
###GET specific Content Integrity Profile settings###
### User must obtain the Profile ID and manually enter it or by holding the
value on a variable ###
Write-Output "GET specific Content Integrity Profile settings"
Write-Output "---------------------------------------------"
#$profileID = "53f00be5-9aa0-41ed-9fdf-ea99f6784238"
$SCIlist = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/contentIntegrity/$profileId" -Method 'GET' -Headers
$authHeader
Write-Output $SCIlist | ConvertTo-Json -Depth 35
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/contentIntegrity/{profileId}

PATCH Body Sample

```
### #PATCH content integrity profile ####
###Pass the ID of the CIC profile###
###@profileId = 53f00be5-9aa0-41ed-9fdf-ea99f6784238
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/contentIntegrity/{{profileId}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
  "data": {
    "type": "contentIntegrity",
    "attributes": {
      "profileName": "Profile_API_TEST_PATCH",
      "host": true,
      "port": 1344,
      "path": true,
      "mode": "REQMOD",
      "scanLimit": {
        "enabled": true,
        "bytes": 100000
      },
      "headerOverrides": {
        "httpHost": {
          "enabled": true,
          "value": "192.168.100.00"
        },
        "xClientIp": {
          "enabled": true,
          "value": "192.168.100.00"
        },
        "xServerIp": {
          "enabled": false
```

PATCH Response Body Sample

```
{ "data": { "id": "d7c7c566-e2e1-4dd9-8332-9c2aa63f7bff", "type":
"contentIntegrity", "attributes": { "profileName":
"Profile_API_TEST_PATCH", "host": true, "port": 1344, "path": true,
"mode": "REQMOD", "scanLimit": { "enabled": true, "bytes": 100000 },
"headerOverrides": { "httpHost": { "enabled": true, "value":
"192.168.100.00" }, "xClientIp": { "enabled": true, "value":
"192.168.100.00" }, "xServerIp": { "enabled": false, "value": "" },
"xSubscriberIp": { "enabled": false, "value": "" }, "xAuthenticatedUser":
```

PATCH PowerShell Sample

```
### PATCH specific Content Integrity Profile settings###
### User must obtain the Profile ID and manually enter it or by holding the
value on a variable ###
Write-Output "PATCH specific Content Integrity Profile settings"
Write-Output "------------------------------------------------"
$update =
'{
  "data": {
    "type": "contentIntegrity",
    "attributes": {
      "profileName": "Profile_API_TEST_PATCH",
      "host": true,
      "port": 1344,
      "path": true,
      "mode": "REQMOD",
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/contentIntegrity/{profileId}

DELETE Body Sample

```
###DELETE spacific content integrity profile settings####
###Pass the ID of the CIC profile###
###@profileId = 53f00be5-9aa0-41ed-9fdf-ea99f6784238
DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/contentIntegrity/{{profileId}}
/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
{ "data": [], "links": { "self": "/admin/v2/sites/0de7387f-1506-427d-
b997-e18bcdb8cbae/contentIntegrity" } }
```

DELETE PowerShell Sample

```
###DELETE specific content integrity profile settings####
###Pass the ID of the CIC profile###
###@profileId = 53f00be5-9aa0-41ed-9fdf-ea99f6784238
Write-Output "DELETE specific content integrity profile settings"
Write-Output "-------------------------------------------------"
#$profileID = "53f00be5-9aa0-41ed-9fdf-ea99f6784238"
$DCIClist = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/contentIntegrity/$profileId" -Method 'DELETE' -
Headers $authHeader
Write-Output $DCIClist | ConvertTo-Json -Depth 35
```

Endpoint: /sites/{siteId}/custom-commands

Note: Available as of EFT 8.1.0

This endpoint provides the ability to return a list of custom commands defined on an EFT site or create a new custom command.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/custom-commands

GET Body Sample

```
###
##Get Site Custom Commands
GET https://{{host}}/admin/v2//sites/{{siteId}}/custom-commands HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": [ { "id": "9a0c4aec-862e-4e72-aec4-9b1f04f00bb3", "type":
"customCommand", "attributes": { "name": "Command 1" }, "links": {
"self": "/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-01a10524c048/custom-
commands/9a0c4aec-862e-4e72-aec4-9b1f04f00bb3" } }, { "id": "090dde3a-
c039-44ab-b03d-5cc994548ce7", "type": "customCommand", "attributes": {
"name": "Run to the Moon" }, "links": { "self":
"/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-01a10524c048/custom-
```

GET PowerShell Sample

```
###
#Get Site Custom Commands
Write-Output "Get Site Custom Commands"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/custom-commands" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/custom-commands

POST Body Sample

```
## Create Custom Command on a specific EFT Site
POST https://{{host}}/admin/v2/sites/{{Siteid}}/custom-commands HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data":{
        "id":"",
        "type": "customCommand",
            "attributes": {
                "enabled": true,
                "name": "Command 5",
                "description": "this is a test to run command 5",
                "executable": "c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"",
                "parameters": "",
                "redirectOutputToLog": true,
                "useProcessTimeOut": true,
                "processTimeOut": 30,
                "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log",
                "redirectOutputToClient": true,
                "useMinNumOfParams": false,
                "minNumOfParams": 1,
                "minNumOfParamsMsg": ""
            }
    }
}
###
```

POST Response Body Sample

```
{ "data": { "id": "f1d88549-e4aa-49da-b685-536fe0d93b62", "type":
"customCommand", "attributes": { "enabled": true, "name": "Command 5",
"description": "this is a test to run command 5", "executable":
"c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"", "parameters": "",
"redirectOutputToLog": true, "useProcessTimeOut": true, "processTimeOut":
30, "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log", "redirectOutputToClient": true,
"useMinNumOfParams": false, "minNumOfParams": 1, "minNumOfParamsMsg": ""
}, "relationships": { "executionPermissions": { "data": { "users": [],
"userGroups": [ { "type": "userGroup", "id": "a0db9377-92a3-4a38-8c0f-
09f268d78ff1", "meta": { "name": "All Users" } } ] } } }, "links": {
"self": "/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-01a10524c048/custom-
```

POST PowerShell Sample

```
###
#POST new site Custom Command
Write-Output "POST Custom Command (New Custom Command)"
Write-Output "-----"
$update =
'{
    "data":{
        "id":"",
        "type": "customCommand",
            "attributes": {
                "enabled": true,
                "name": "Command 5",
                "description": "this is a test to run command 5",
                "executable": "c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"",
                "parameters": "",
                "redirectOutputToLog": true,
                "useProcessTimeOut": true,
                "processTimeOut": 30,
                "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log",
                "redirectOutputToClient": true,
                "useMinNumOfParams": false,
                "minNumOfParams": 1,
                "minNumOfParamsMsg": ""
            }
    }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-
```

Endpoint: /sites/{siteId}/custom-commands/{customCommandID}

Note: Available as of EFT 8.1.0

Returns a custom command defined on an EFT site and allows the ability to get, patch (update), and delete the custom command.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/custom-commands/{customCommandID}

GET Body Sample

```
### Get Custom Commands for a Site ###
# @name customCommandList
GET https://{{host}}/admin/v2/sites/{{Siteid}}/custom-commands HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@customCommandID = {{customCommandList.response.body.$.data[0].id}}

### Get specific Custom Command from a Site ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/custom-
commands/{{customCommandID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": { "id": "9a0c4aec-862e-4e72-aec4-9b1f04f00bb3", "type":
"customCommand", "attributes": { "enabled": true, "name": "Command 1",
"description": "this is a test to run command 1", "executable":
"c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"", "parameters": "",
"redirectOutputToLog": true, "useProcessTimeOut": true, "processTimeOut":
30, "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log", "redirectOutputToClient": true,
"useMinNumOfParams": false, "minNumOfParams": 1, "minNumOfParamsMsg": ""
}, "relationships": { "executionPermissions": { "data": { "users": [],
"userGroups": [ { "type": "userGroup", "id": "a0db9377-92a3-4a38-8c0f-
09f268d78ff1", "meta": { "name": "All Users" } } ] } } }, "links": {
```

GET PowerShell Sample

```
###
#Get Site Custom Commands
Write-Output "Get Site Custom Commands"
Write-Output "-----"
$commands = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-commands"
-Method 'GET' -Headers $authHeader
#Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($commands in $commands.id)
{
    Write-Output ("Custom Command found:  $($commands.id):
$($commands.attributes.name)")
}
$customCommandID = $commands.data[0].id

###
#Get Custom Command
Write-Output "Get Custom Commands for: "  $customCommandID
Write-Output "-----"
$commandList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-
commands/$customCommandID" -Method 'GET' -Headers $authHeader
Write-Output $commandList | ConvertTo-Json -Depth 10
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/custom-commands/{customCommandID}

PATCH Body Sample

```
### Get Custom Commands for a Site which is needed to patch a Custom
Commmand ###
# @name customCommandList
GET https://{{host}}/admin/v2/sites/{{Siteid}}/custom-commands HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@customCommandID = {{customCommandList.response.body.$.data[0].id}}

###
#Path Custom Command on a specific EFT site
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/custom-
commands/{{customCommandID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data":{
        "id":"",
        "type": "customCommand",
            "attributes": {
                "enabled": true,
                "name": "Updated Custom Command Name",
                "description": "Updated Custom Command Description",
                "executable": "c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"",
                "parameters": "",
                "redirectOutputToLog": true,
                "useProcessTimeOut": false,
                "processTimeOut": 30,
                "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log",
                "redirectOutputToClient": true,
                "useMinNumOfParams": false,
                "minNumOfParams": 1,
                "minNumOfParamsMsg": ""
            }
    }
}
```

PATCH Response Body Sample

```
{ "data": { "id": "9a0c4aec-862e-4e72-aec4-9b1f04f00bb3", "type":
"customCommand", "attributes": { "enabled": true, "name": "Updated Custom
Command Name", "description": "Updated Custom Command Description",
"executable": "c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"", "parameters": "",
"redirectOutputToLog": true, "useProcessTimeOut": false,
"processTimeOut": 30, "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log", "redirectOutputToClient": true,
"useMinNumOfParams": false, "minNumOfParams": 1, "minNumOfParamsMsg": ""
}, "relationships": { "executionPermissions": { "data": { "users": [],
"userGroups": [ { "type": "userGroup", "id": "a0db9377-92a3-4a38-8c0f-
09f268d78ff1", "meta": { "name": "All Users" } } ] } } }, "links": {
```

PATCH PowerShell Sample

```
############################
#Patch a Custom Command
###
#Get Site Custom Commands
Write-Output "Get Site Custom Commands in order to Patch"
Write-Output "-----"
$commands = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-commands"
-Method 'GET' -Headers $authHeader
#Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($commands in $commands.id)
{
    Write-Output ("Custom Command found:  $($commands.id):
$($commands.attributes.name)")
}
$customCommandID = $commands.data[0].id

#PATCH Custom Command
Write-Output "Custom Command Patch"
Write-Output "Custom Command: $customCommandID"
Write-Output "-----"
$update =
'{
    "data":{
        "id":"",
        "type": "customCommand",
            "attributes": {
                "enabled": true,
                "name": "Updated Custom Command Name 2",
                "description": "Updated Custom Command Description 2",
                "executable": "c:\\windows\\system32\\cscript.exe
\"c:\\users\\administrator\\desktop\\server.vbs\"",
                "parameters": "",
                "redirectOutputToLog": true,
                "useProcessTimeOut": false,
                "processTimeOut": 30,
                "logFileName": "C:\\ProgramData\\Globalscape\\EFT
Server\\Logs\\cmdout.log",
                "redirectOutputToClient": true,
                "useMinNumOfParams": false,
                "minNumOfParams": 1,
                "minNumOfParamsMsg": ""
            }
    }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/custom-commands/{customCommandID}

DELETE Body Sample

```
##DELETE Custom Commands
### Get Custom Commands for a Site which is needed to delete a Custom
Commmand ###
# @name customCommandList
GET https://{{host}}/admin/v2/sites/{{Siteid}}/custom-commands HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@customCommandID = {{customCommandList.response.body.$.data[0].id}}
#

##Delete Request passing custom command id
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/custom-
commands/{{customCommandID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Fri, 15 Jul 2022 21:48:08 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
### DELETE CUSTOM COMMAND
#Get Site Custom Commands in order to get custom command id
Write-Output "Get Site Custom Commands in order to Delete"
Write-Output "-----"
$commands = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-commands"
-Method 'GET' -Headers $authHeader
#Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($commands in $commands.id)
{
    Write-Output ("Custom Command found:  $($commands.id):
$($commands.attributes.name)")
}
$customCommandID = $commands.data[0].id

Write-Output "Deleting Custom Command: "  $customCommandID
Write-Output "-----"
$commandList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/custom-
commands/$customCommandID" -Method 'DELETE' -Headers $authHeader
Write-Output $commandList | ConvertTo-Json -Depth 10
```

Endpoint: /sites/{siteId}/user-templates

Note: Available as of EFT 8.1.0

Returns a list of user templates defined on an EFT site and allows the ability to get and create templates.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/user-templates

GET Body Sample

```
### Get User Settings Template ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": [ { "attributes": { "as2": { "inbound": { "enabled": true },
"outbound": { "enabled": true } }, "changePassword": { "enabled":
"inherit", "value": { "changeAdminProvidedPasswordUponFirstUse":
"inherit", "passwordExpiration": { "enabled": true, "value": {
"maxAgeDays": 90, "emailUpon": true, "remindPrior": { "enabled": false,
"value": { "daysBefore": 5, "emailUser": true } } } } } },
"connectionTimeout": { "enabled": true, "value": { "maxSec": 600 } },
"connectionsFromSameIpLimit": { "enabled": false, "value": { "maxNumber":
0 } }, "downloadSizeLimit": { "enabled": false, "value": { "maxKBytes": 0
} }, "downloadsPerSessionLimit": { "enabled": false, "value": {
"maxNumber": 0 } }, "enabled": true, "ftp": { "enabled": true, "value": {
"enableFxp": "inherit", "enableNoop": "inherit", "enableComb": "yes",
"enableXcrc": "yes", "enableZlib": true, "banner": { "message": "Login
OK. Proceed.", "usage": "replaceDefault" } } }, "ftps": { "enabled": true
}, "hasHomeFolderAsRoot": true, "homeFolder": { "enabled": true, "value":
{ "path": "/" } }, "http": { "enableHttp": true, "enableHttps": true,
"enableWtc": true, "enableShareFolder": true, "enableSendMessage": true
}, "invalidLoginLimit": { "enabled": "inherit", "value": { "maxCount":
90, "periodMin": 5, "action": "lock", "actionDurationMin": 30 } },
"ipAccessLimit": { "enabled": false, "value": { "rules": [],
"defaultRule": "allowAccess" } }, "name": "Default Settings",
"passwordComplexity": { "enabled": "inherit", "value": { "minLength": 8,
```

GET PowerShell Sample

```
# get sites
Write-Output "Site"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites" -Method 'GET' -Headers
$authHeader
Write-Output $siteList | ConvertTo-Json
foreach ($site in $siteList.data)
{
    Write-Output ("Site  $($site.id): $($site.attributes.name)")
}

$siteID = $siteList.data[0].id

###
#Get Site User Settings Template
Write-Output "Get Site User Settings Template"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
##
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/user-templates

POST Body Sample

```
### POST User Settings Template (Create)###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/user-templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data":{
        "type": "userTemplate",
        "attributes":{
            "name": "GSB Template 4",
            "accountEnabled": true,
            "homeFolder": {
                "enabled": false,
                "value": {
                    "path": ""
                }
            }
        },
        "passwordComplexity": {
          "enabled": "inherit",
          "value": {
            "minLength": 8,
            "characterCategories": {
              "enabled": true,
              "value": {
                "characterCount": 4,
                "upperCaseRequired": true,
                "lowerCaseRequired": true,
                "numericRequired": true,
                "nonAlphaNumericRequired": false,
                "non7bitAsciiRequired": false
              }
            },
            "usernameCharLimit": {
              "enabled": true,
              "value": {
                "minDisallowed": 3
              }
            }
          }
        },
        "sftp": {
          "enabled": true,
          "value": {
            "authenticationType": "password"
          }
        },
        "ssl": {
          "authenticationType": "password"
        }      }}
```

POST Response Body Sample

```
{ "data": { "attributes": { "as2": { "inbound": { "enabled": true },
"outbound": { "enabled": true } }, "changePassword": { "enabled":
"inherit", "value": { "changeAdminProvidedPasswordUponFirstUse":
"inherit", "passwordExpiration": { "enabled": true, "value": {
"maxAgeDays": 90, "emailUpon": true, "remindPrior": { "enabled": false,
"value": { "daysBefore": 5, "emailUser": true } } } } } } },
"connectionTimeout": { "enabled": true, "value": { "maxSec": 600 } },
"connectionsFromSameIpLimit": { "enabled": false, "value": { "maxNumber":
0 } }, "downloadSizeLimit": { "enabled": false, "value": { "maxKBytes": 0
} }, "downloadsPerSessionLimit": { "enabled": false, "value": {
"maxNumber": 0 } }, "enabled": true, "ftp": { "enabled": true, "value": {
"enableFxp": "inherit", "enableNoop": "inherit", "enableComb": "yes",
"enableXcrc": "yes", "enableZlib": true, "banner": { "message": "Login
OK. Proceed.", "usage": "replaceDefault" } } }, "ftps": { "enabled": true
}, "hasHomeFolderAsRoot": true, "homeFolder": { "enabled": false,
"value": { "path": "/" } }, "http": { "enableHttp": true, "enableHttps":
true, "enableWtc": true, "enableShareFolder": true, "enableSendMessage":
true }, "invalidLoginLimit": { "enabled": "inherit", "value": {
"maxCount": 90, "periodMin": 5, "action": "lock", "actionDurationMin": 30
} }, "ipAccessLimit": { "enabled": false, "value": { "rules": [],
"defaultRule": "allowAccess" } }, "name": "GSB Template 4",
"passwordComplexity": { "enabled": "inherit", "value": { "minLength": 8,
"characterCategories": { "enabled": true, "value": { "characterCount": 4,
"upperCaseRequired": true, "lowerCaseRequired": true, "numericRequired":
true, "nonAlphaNumericRequired": false, "non7bitAsciiRequired": false }
}, "usernameCharLimit": { "enabled": true, "value": { "minDisallowed": 3
} }, "repeatingCharLimit": { "enabled": true, "value": { "minDisallowed":
3 } }, "forbiddenDictionary": { "enabled": true, "value": {
"dictionaryFilePath": "", "noBackwardsWord": false } } } },
"passwordHistory": { "enabled": "inherit", "value": { "historyDepth": 4 }
}, "sftp": { "enabled": true, "value": { "authenticationType": "password"
} }, "ssl": { "authenticationType": "password" },
"totalConnectionsLimit": { "enabled": false, "value": { "maxNumber": 0 }
```

POST PowerShell Sample

```
###
#POST Site User Settings Template (Create)
#POST new site Custom Command
Write-Output "POST User Settings Template (New User Settings Template)"
Write-Output "-----"
$update =
'{
    "data":{
        "type": "userTemplate",
        "attributes":{
            "name": "GSB Template 4",
            "accountEnabled": true,
            "homeFolder": {
                "enabled": false,
                "value": {
                    "path": ""
                }
            }
        },
        "passwordComplexity": {
          "enabled": "inherit",
          "value": {
            "minLength": 8,
            "characterCategories": {
              "enabled": true,
              "value": {
                "characterCount": 4,
                "upperCaseRequired": true,
                "lowerCaseRequired": true,
                "numericRequired": true,
                "nonAlphaNumericRequired": false,
                "non7bitAsciiRequired": false
              }
            },
            "usernameCharLimit": {
              "enabled": true,
              "value": {
                "minDisallowed": 3
              }
            }
          }
        },
        "sftp": {
          "enabled": true,
          "value": {
            "authenticationType": "password"
          }
        },
        "ssl": {
          "authenticationType": "password"
        }
      }

}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates" -Method 'POST' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteId}/user-templates/{userTemplateid}

Note: Available as of EFT 8.1.0

Returns a user settings template defined on an EFT site and allows the ability to get, update, and delete the custom command.

- GET

- PATCH
- DELETE

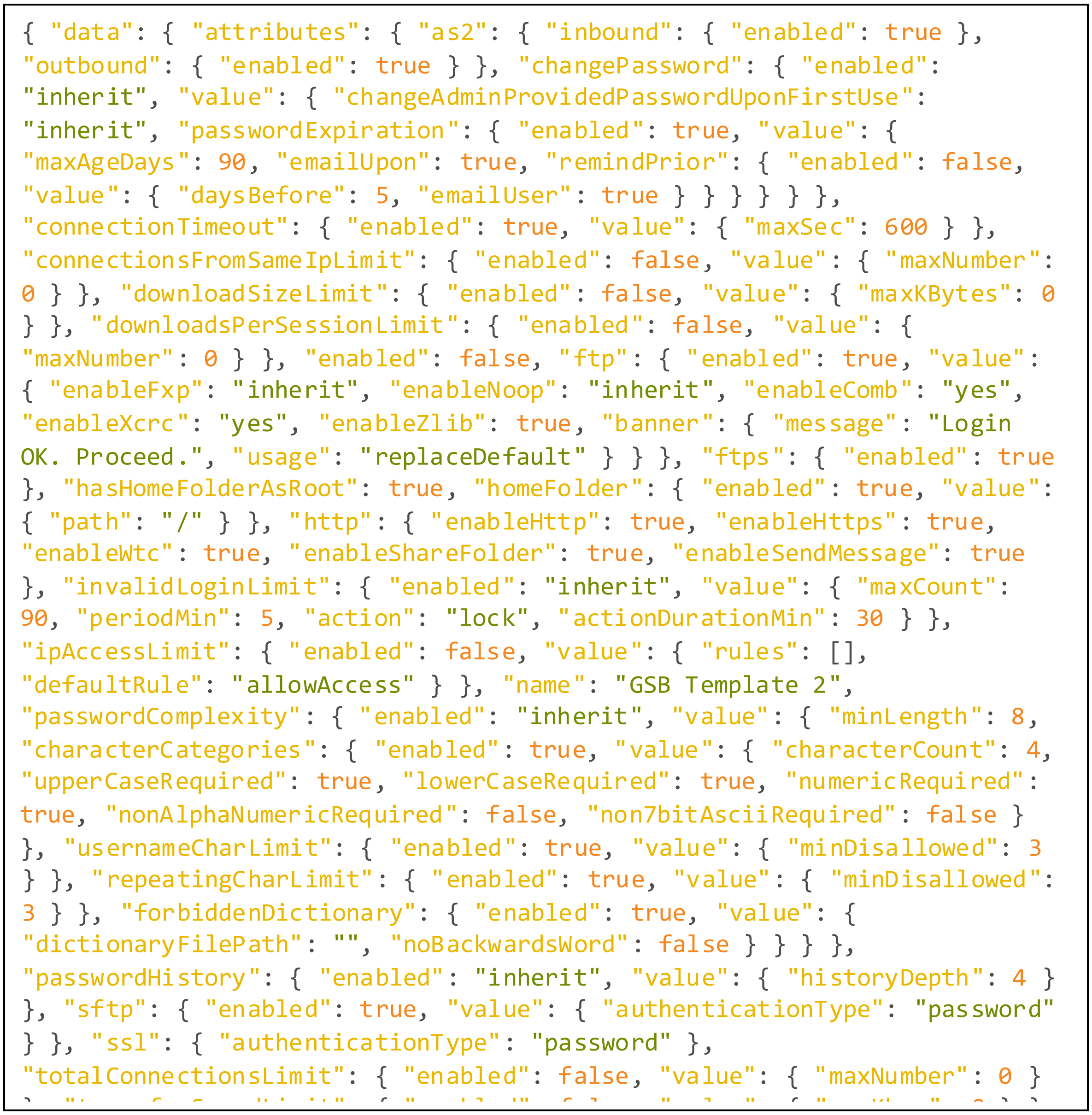
Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/user-templates/{userTemplateid}

GET Body Sample

```
### Get User Settings Template ### Need to obtain templates first
# @name userTemplates
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@userTemplateid = {{userTemplates.response.body.$.data[3].id}}
#Update the array to obtain the desired user template


### GET User Template by Template ID ###
#
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-templates/{{userTemplateid}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample



```
{ "data": { "attributes": { "as2": { "inbound": { "enabled": true },
"outbound": { "enabled": true } }, "changePassword": { "enabled":
"inherit", "value": { "changeAdminProvidedPasswordUponFirstUse":
"inherit", "passwordExpiration": { "enabled": true, "value": {
"maxAgeDays": 90, "emailUpon": true, "remindPrior": { "enabled": false,
"value": { "daysBefore": 5, "emailUser": true } } } } } },
"connectionTimeout": { "enabled": true, "value": { "maxSec": 600 } },
"connectionsFromSameIpLimit": { "enabled": false, "value": { "maxNumber":
0 } }, "downloadSizeLimit": { "enabled": false, "value": { "maxKBytes": 0
} }, "downloadsPerSessionLimit": { "enabled": false, "value": {
"maxNumber": 0 } }, "enabled": false, "ftp": { "enabled": true, "value":
{ "enableFxp": "inherit", "enableNoop": "inherit", "enableComb": "yes",
"enableXcrc": "yes", "enableZlib": true, "banner": { "message": "Login
OK. Proceed.", "usage": "replaceDefault" } } }, "ftps": { "enabled": true
}, "hasHomeFolderAsRoot": true, "homeFolder": { "enabled": true, "value":
{ "path": "/" } }, "http": { "enableHttp": true, "enableHttps": true,
"enableWtc": true, "enableShareFolder": true, "enableSendMessage": true
}, "invalidLoginLimit": { "enabled": "inherit", "value": { "maxCount":
90, "periodMin": 5, "action": "lock", "actionDurationMin": 30 } },
"ipAccessLimit": { "enabled": false, "value": { "rules": [],
"defaultRule": "allowAccess" } }, "name": "GSB Template 2",
"passwordComplexity": { "enabled": "inherit", "value": { "minLength": 8,
"characterCategories": { "enabled": true, "value": { "characterCount": 4,
"upperCaseRequired": true, "lowerCaseRequired": true, "numericRequired":
true, "nonAlphaNumericRequired": false, "non7bitAsciiRequired": false }
}, "usernameCharLimit": { "enabled": true, "value": { "minDisallowed": 3
} }, "repeatingCharLimit": { "enabled": true, "value": { "minDisallowed":
3 } }, "forbiddenDictionary": { "enabled": true, "value": {
"dictionaryFilePath": "", "noBackwardsWord": false } } } },
"passwordHistory": { "enabled": "inherit", "value": { "historyDepth": 4 }
}, "sftp": { "enabled": true, "value": { "authenticationType": "password"
} }, "ssl": { "authenticationType": "password" },
"totalConnectionsLimit": { "enabled": false, "value": { "maxNumber": 0 }
```

GET PowerShell Sample

```
########################################
#GET Specific User Settings Template

###
#Get Specific User Settings Templates
Write-Output "Get Specific User Settings Template"
Write-Output "-----"
$userTemplates = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($userTemplates in $userTemplates.id)
{
    Write-Output ("User Template found:  $($userTemplates.id):
$($userTemplates.attributes.name)")
}
$userTemplateID = $userTemplates.data[3].id

###
#Get Specific User Template
Write-Output "Get User Template for: "  $userTemplateID " with name: "
$userTemplate.data.attributes.name
Write-Output "-----"
$userTemplate = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates/$userTemplateID" -Method 'GET' -Headers $authHeader
Write-Output $userTemplate | ConvertTo-Json -Depth 10
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/user-templates/{userTemplateid}

PATCH Body Sample

```
### Get User Settings Template ### Need to obtain templates first
# @name userTemplates
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@userTemplateid = {{userTemplates.response.body.$.data[3].id}}
#Update the array to obtain the desired user template


### PATCH User Template by Template ID
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/user-
templates/{{userTemplateid}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data": {
        "type": "userTemplate",
        "attributes": {
            "enabled": false,
            "homeFolder": {
                "enabled": true,
                "value": {
                    "path": ""
                }
            }
        }
    }
}
```

PATCH Response Body Sample

```
{ "data": { "attributes": { "as2": { "inbound": { "enabled": true },
"outbound": { "enabled": true } }, "changePassword": { "enabled":
"inherit", "value": { "changeAdminProvidedPasswordUponFirstUse":
"inherit", "passwordExpiration": { "enabled": true, "value": {
"maxAgeDays": 90, "emailUpon": true, "remindPrior": { "enabled": false,
"value": { "daysBefore": 5, "emailUser": true } } } } } },
"connectionTimeout": { "enabled": true, "value": { "maxSec": 600 } },
"connectionsFromSameIpLimit": { "enabled": false, "value": { "maxNumber":
0 } }, "downloadSizeLimit": { "enabled": false, "value": { "maxKBytes": 0
} }, "downloadsPerSessionLimit": { "enabled": false, "value": {
"maxNumber": 0 } }, "enabled": false, "ftp": { "enabled": true, "value":
{ "enableFxp": "inherit", "enableNoop": "inherit", "enableComb": "yes",
"enableXcrc": "yes", "enableZlib": true, "banner": { "message": "Login
OK. Proceed.", "usage": "replaceDefault" } } }, "ftps": { "enabled": true
}, "hasHomeFolderAsRoot": true, "homeFolder": { "enabled": true, "value":
{ "path": "/" } }, "http": { "enableHttp": true, "enableHttps": true,
"enableWtc": true, "enableShareFolder": true, "enableSendMessage": true
}, "invalidLoginLimit": { "enabled": "inherit", "value": { "maxCount":
90, "periodMin": 5, "action": "lock", "actionDurationMin": 30 } },
"ipAccessLimit": { "enabled": false, "value": { "rules": [],
"defaultRule": "allowAccess" } }, "name": "GSB Template 2",
"passwordComplexity": { "enabled": "inherit", "value": { "minLength": 8,
"characterCategories": { "enabled": true, "value": { "characterCount": 4,
"upperCaseRequired": true, "lowerCaseRequired": true, "numericRequired":
true, "nonAlphaNumericRequired": false, "non7bitAsciiRequired": false }
}, "usernameCharLimit": { "enabled": true, "value": { "minDisallowed": 3
} }, "repeatingCharLimit": { "enabled": true, "value": { "minDisallowed":
3 } }, "forbiddenDictionary": { "enabled": true, "value": {
"dictionaryFilePath": "", "noBackwardsWord": false } } } },
"passwordHistory": { "enabled": "inherit", "value": { "historyDepth": 4 }
}, "sftp": { "enabled": true, "value": { "authenticationType": "password"
} }, "ssl": { "authenticationType": "password" },
"totalConnectionsLimit": { "enabled": false, "value": { "maxNumber": 0 }
```

PATCH PowerShell Sample

```
####################################################
##PATCH (Update) specific User Settings Template
###
#Get Specific User Settings Templates
Write-Output "Get Specific User Settings Template"
Write-Output "-----"
$userTemplates = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($userTemplates in $userTemplates.id)
{
    Write-Output ("User Template found:  $($userTemplates.id):
$($userTemplates.attributes.name)")
}
$userTemplateID = $userTemplates.data[3].id

###
#Get Specific User Template
Write-Output "Get User Template for: "  $userTemplateID " with name: "
$userTemplate.data.attributes.name
Write-Output "-----"
$userTemplate = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates/$userTemplateID" -Method 'GET' -Headers $authHeader
Write-Output $userTemplate | ConvertTo-Json -Depth 10


###
##Patch Specific User Settings Template
Write-Output "User Settings Template Patch (update)"
Write-Output "-----"
$update =
'{
    "data":{
        "type": "userTemplate",
          "attributes": {
            "enabled": false,
            "homeFolder": {
                "enabled": true,
                "value": {
                    "path": ""
                }
            }
        }
    }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates/$userTemplateID" -Method 'PATCH' -Headers $authHeader -Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/user-templates/{userTemplateid}

DELETE Body Sample

```
###
#Delete Specific User Settings Template
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/user-
templates/{{userTemplateid}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
#
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Thu, 21 Jul 2022 13:38:16 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
##########################################################
#Delete User Settings Template

#Get Specific User Settings Templates
Write-Output "Get Specific User Settings Template"
Write-Output "-----"
$userTemplates = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
foreach ($userTemplates in $userTemplates.id)
{
    Write-Output ("User Template found:  $($userTemplates.id):
$($userTemplates.attributes.name)")
}
$userTemplateID = $userTemplates.data[3].id

###
#Get Specific User Template
Write-Output "Get User Template for: "  $userTemplateID " with name: "
$userTemplate.data.attributes.name
Write-Output "-----"
$userTemplate = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates/$userTemplateID" -Method 'GET' -Headers $authHeader
Write-Output $userTemplate | ConvertTo-Json -Depth 10

#Delete Request
Write-Output "Deleting User Settings Template: "  $userTemplateID
Write-Output "-----"
$commandList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
templates/$userTemplateID" -Method 'DELETE' -Headers $authHeader
Write-Output $commandList | ConvertTo-Json -Depth 10
```

Endpoint: /sites/{siteId}/user-groups

Note: Available as of EFT 8.1.0

Provides the ability to return a .list of user groups for a specific EFT site or create a new permission group.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups

GET Body Sample

```
### Get List of all User's Permissions Groups for the Site####
#
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": [ { "type": "userGroup", "id": "a0db9377-92a3-4a38-8c0f-
09f268d78ff1", "attributes": { "name": "All Users" }, "relationships": {
"users": { "data": [ { "type": "user", "id": "b90be871-86ff-4060-b3cc-
007903dff7f7", "meta": { "name": "Ivan" } }, { "type": "user", "id":
"953bcb67-fa8c-44e2-a056-ecfb5080d13c", "meta": { "name": "TestUser" } }
] } }, "links": { "self": "/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-
01a10524c048/user-groups/a0db9377-92a3-4a38-8c0f-09f268d78ff1" } }, {
"type": "userGroup", "id": "07caceb2-6698-4ef7-baeb-f437fec972a2",
"attributes": { "name": "Administrators" }, "relationships": { "users": {
"data": [] } }, "links": { "self": "/admin/v2/sites/c7d3ecfe-a100-4bcd-
b830-01a10524c048/user-groups/07caceb2-6698-4ef7-baeb-f437fec972a2" } },
{ "type": "userGroup", "id": "b8448b77-01d8-47d5-8ee7-9f407c95e39f",
"attributes": { "name": "Guests" }, "relationships": { "users": { "data":
```

GET PowerShell Sample

```
# Get sites
Write-Output "Site"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites" -Method 'GET' -Headers
$authHeader
Write-Output $siteList | ConvertTo-Json
foreach ($site in $siteList.data)
{
    Write-Output ("Site  $($site.id): $($site.attributes.name)")
}

$siteID = $siteList.data[0].id
##########
#Get User Permission Groups
Write-Output "Get User Permission Groups"
Write-Output "-----"
$userGroups = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-groups" -
Method 'GET' -Headers $authHeader
Write-Output $userGroups | ConvertTo-Json -Depth 10
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups

POST Body Sample

```
### Create User Permission Group ###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data": {
        "type": "userGroup",
        "attributes": {
            "name": "QA_API_User_Group_TEST"
        }
    }
}
```

POST Response Body Sample

```
{ "data": { "type": "userGroup", "id": "1ba8c63f-6e37-4783-b556-
be5bf831dc6c", "attributes": { "name": "QA_API_User_Group_TEST" },
"relationships": { "users": { "data": [] } }, "links": { "self":
"/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-01a10524c048/user-
groups/1ba8c63f-6e37-4783-b556-be5bf831dc6c" } } }
```

POST PowerShell Sample

```
#POST New Permission Group
Write-Output "POST User Permissions Group (New Permission Group)"
Write-Output "-----"
$update =
'{
  "data": {
        "type": "userGroup",
        "attributes": {
            "name": "QA_API_User_Group_TEST"
        }
    }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-groups"
-Method 'POST' -Headers $authHeader -Body $update
```

Endpoint: /sites/{siteId}/user-groups/{groupid}

Note: Available as of EFT 8.1.0

Provides the ability to return a user group's details, update a permission group name or delete a user group.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups/{groupid}

GET Body Sample

```
### Get List of all User's Permissions Groups for the Site####
# @name UGlist
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@groupid = {{UGlist.response.body.$.data[3].id}}


### GET Details for specific User Permission Group####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups/{{groupid}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "type": "userGroup", "id": "7386d09b-ac9d-42c8-a7dd-
8ffb840d58e3", "attributes": { "name": "QA_API_User_Group_TEST" },
"relationships": { "users": { "data": [] } }, "links": { "self":
"/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-01a10524c048/user-
groups/7386d09b-ac9d-42c8-a7dd-8ffb840d58e3" } } }
```

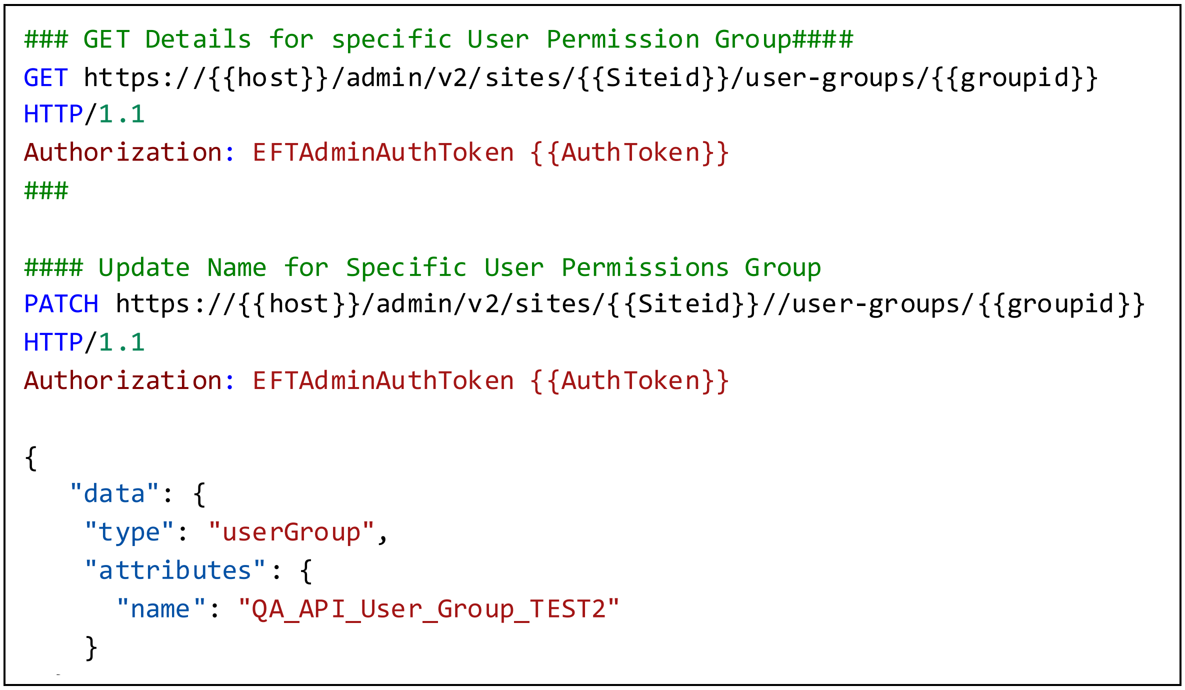
GET PowerShell Sample

```
##########
#Get User Permission Groups
$userGroups = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-groups" -
Method 'GET' -Headers $authHeader
foreach ($userGroups in $userGroups.id)
{
    Write-Output ("User Permission Group found:  $($userGroups.id):
$($userGroups.attributes.name)")
}
$userGroupID = $userGroups.data[3].id


#Get Specific User Permission Group
Write-Output "Get Specific User Permission Group"
Write-Output "-----"
$userGroup = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID" -Method 'GET' -Headers $authHeader
Write-Output $userGroup | ConvertTo-Json -Depth 10
```

Method & URL

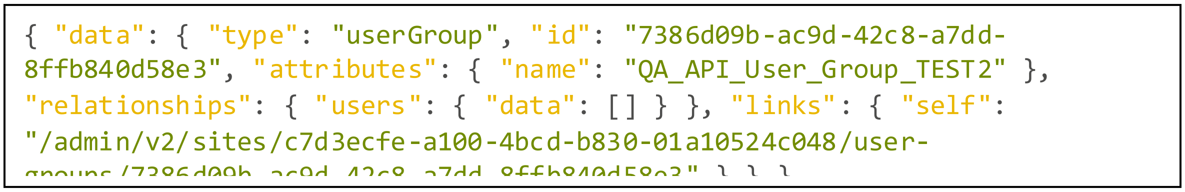
PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups/{groupid}

PATCH Body Sample



```
### GET Details for specific User Permission Group####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups/{{groupid}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###

#### Update Name for Specific User Permissions Group
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}//user-groups/{{groupid}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
   "data": {
    "type": "userGroup",
    "attributes": {
      "name": "QA_API_User_Group_TEST2"
    }
```

PATCH Response Body Sample



```
{ "data": { "type": "userGroup", "id": "7386d09b-ac9d-42c8-a7dd-
8ffb840d58e3", "attributes": { "name": "QA_API_User_Group_TEST2" },
"relationships": { "users": { "data": [] } }, "links": { "self":
"/admin/v2/sites/c7d3ecfe-a100-4bcd-b830-01a10524c048/user-
groups/7386d09b-ac9d-42c8-a7dd-8ffb840d58e3" } } }
```

PATCH PowerShell Sample

```
#Get Specific User Permission Group
Write-Output "Get Specific User Permission Group"
Write-Output "-----"
$userGroup = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID" -Method 'GET' -Headers $authHeader
Write-Output $userGroup | ConvertTo-Json -Depth 10

###
##Patch Specific User Permissions Group
Write-Output "User Permission Group Patch (update)"
Write-Output "-----"
$update =
'{
   "data": {
    "type": "userGroup",
    "attributes": {
      "name": "QA_API_User_Group_TEST2"
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID" -Method 'PATCH' -Headers $authHeader -Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups/{groupid}

DELETE Body Sample

```
### Delete Specific User Permission Group ###
Delete https://{{host}}/admin/v2/sites/{{Siteid}}//user-
groups/{{groupid}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Fri, 22 Jul 2022 22:40:06 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
#Get Specific User Permission Group
Write-Output "Get Specific User Permission Group"
Write-Output "-----"
$userGroup = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID" -Method 'GET' -Headers $authHeader
Write-Output $userGroup | ConvertTo-Json -Depth 10

###
#DELETE Specific User Permission Group
Write-Output "Delete Specific User Permission Group"
Write-Output "-----"
$commandList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID" -Method 'DELETE' -Headers $authHeader
Write-Output $commandList | ConvertTo-Json -Depth 10
```

Endpoint: /sites/{siteId}/user-groups/{groupid}/relationships/users

Note: Available as of EFT 8.1.0

Provides the ability to return a user group's details, update a permission group name or delete a user group.

- POST
- DELETE

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups/{groupid}/relationships/users

POST Body Sample

```
###Get Users
# @name user
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@userid = {{user.response.body.$.data[3].id}}
#Store a user into a variable in order to add to a group later

###
### Get List of all User's Permissions Groups for the Site####
# @name UGlist
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@groupid = {{UGlist.response.body.$.data[3].id}}
#Store a user group in order to add a user to this group later

### Add User(s) to specific Permission Group ###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups/{{groupid}}/relationships/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": [
    {
      "type": "user",
      "id": "{{userid}}"
    }
   ]
}
```

POST Response Body Sample

```
{ "data": [ { "type": "user", "id": "b90be871-86ff-4060-b3cc-007903dff7f7", "meta": { "name": "Ivan" } } ] }
```

POST PowerShell Sample

```
##
#Get User Permission Groups
$userGroups = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-groups" -
Method 'GET' -Headers $authHeader
foreach ($userGroups in $userGroups.id)
{
Write-Output ("User Permission Group found:  $($userGroups.id):
$($userGroups.attributes.name)")
}
$userGroupID = $userGroups.data[3].id

###
#POST Selected User in Selected User Permission Group
Write-Output "POST (Add) User to Selected User Permissions Group"
Write-Output "-----"
$update =
'{
  "data": [
    {
      "type": "user",
      "id": "b90be871-86ff-4060-b3cc-007903dff7f7"
    }
   ]
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID/relationships/users" -Method 'POST' -Headers $authHeader -
Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/user-groups/{groupid}/relationships/users

DELETE Body Sample

```
###Get Users
# @name user
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@userid = {{user.response.body.$.data[3].id}}
#Store a user into a variable in order to add to a group later

###
### Get List of all User's Permissions Groups for the Site####
# @name UGlist
GET https://{{host}}/admin/v2/sites/{{Siteid}}/user-groups HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@groupid = {{UGlist.response.body.$.data[3].id}}
#Store a user group in order to add a user to this group later


### Delete User(s) from specific Permission Group ###
Delete https://{{host}}/admin/v2/sites/{{Siteid}}/user-
groups/{{groupid}}/relationships/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": [
    {
      "type": "user",
      "id": "{{userid}}"
    }
  ]
}
```

DELETE Response Body Sample

```
HTTP/1.1 200 OK Date: Sat, 23 Jul 2022 01:47:11 GMT Content-Type:
application/vnd.api+json Content-Length: 11 Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer {
"data": [] }
```

DELETE PowerShell Sample

```
##
#Get User Permission Groups
$userGroups = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-groups" -
Method 'GET' -Headers $authHeader
foreach ($userGroups in $userGroups.id)
{
      Write-Output ("User Permission Group found:  $($userGroups.id):
$($userGroups.attributes.name)")
}
$userGroupID = $userGroups.data[3].id

#Delete User from User Permission Group
Write-Output "Deleting User from User Permission Group: "
Write-Output "-----"
$update =
'{
  "data": [
    {
      "type": "user",
      "id": "b90be871-86ff-4060-b3cc-007903dff7f7"
    }
   ]
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$commandList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/user-
groups/$userGroupID/relationships/users" -Method 'DELETE' -Headers $authHeader
-Body $update
```

Endpoint: /sites/{siteId}/ram/agents

Note: Available as of EFT 8.1.0

Returns a list of remote agents from a specified site.

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/agents

GET Body Sample

```
### GET Remote Agents
GET  https://{{host}}/admin/v2/sites/{{Siteid}}/ram/agents HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
##
```

GET Response Body Sample

```
{ "data": [ { "id": "4dfd3f28-6ba9-4e19-ae7b-4d3b0b4314f6", "type":
"remote-agent", "attributes": { "name": "ag_IV-2019A", "clid": "00000000-
0000-0000-0000-000000000000", "state": "PendingJoin", "pausing": false,
"resuming": false, "netBios": "IV-2019A", "domain": "", "winOEM": "x64
(AMD or Intel)", "ips": [ "fe80::904:aa5a:7256:ca57 (Ethernet)",
"10.91.160.123 (Ethernet)", "ff01::1 (Ethernet)", "ff02::1 (Ethernet)",
"ff02::c (Ethernet)", "ff02::fb (Ethernet)", "ff02::1:3 (Ethernet)",
"ff02::1:ff56:ca57 (Ethernet)", "224.0.0.1 (Ethernet)", "224.0.0.251
(Ethernet)", "224.0.0.252 (Ethernet)", "239.255.255.250 (Ethernet)", "::1
(Loopback Pseudo-Interface 1)", "127.0.0.1 (Loopback Pseudo-Interface
1)", "ff02::c (Loopback Pseudo-Interface 1)", "239.255.255.250 (Loopback
Pseudo-Interface 1)" ], "macAddresses": [ "0-15-5d-64-e-2c" ],
"lastUpdate": 0, "version": "8.1.0.6", "lastKnownIP": "10.91.160.123",
"nextUpdate": 0, "approvedTime": 0, "secret":
"7102da9c0219696b82679072b9c2f6e6d46fc0da9aa58d3fc1aa2e94f7858cde",
"peakVersion": "8.1.0.6" }, "links": { "self": "/admin/v2/sites/4ebb829f-
f84c-4355-8d55-d43a2852ca12/ram/agents/4dfd3f28-6ba9-4e19-ae7b-
4d3b0b4314f6" } }, { "id": "6ad61dcb-fabe-4627-aebe-065cfe4eadd0",
"type": "remote-agent", "attributes": { "name": "ag_IV-2019C", "clid":
"00000000-0000-0000-0000-000000000000", "state": "PendingJoin",
"pausing": false, "resuming": false, "netBios": "IV-2019C", "domain": "",
"winOEM": "x64 (AMD or Intel)", "ips": [ "fe80::2d7d:3dd6:4ff0:9bba
(Ethernet)", "10.91.160.130 (Ethernet)", "ff01::1 (Ethernet)", "ff02::1
(Ethernet)", "ff02::c (Ethernet)", "ff02::fb (Ethernet)", "ff02::1:3
(Ethernet)", "ff02::1:fff0:9bba (Ethernet)", "224.0.0.1 (Ethernet)",
"224.0.0.251 (Ethernet)", "224.0.0.252 (Ethernet)", "239.255.255.250
(Ethernet)", "::1 (Loopback Pseudo-Interface 1)", "127.0.0.1 (Loopback
Pseudo-Interface 1)", "ff02::c (Loopback Pseudo-Interface 1)",
"239.255.255.250 (Loopback Pseudo-Interface 1)" ], "macAddresses": [ "0-
15-5d-64-e-2e" ], "lastUpdate": 0, "version": "8.1.0.6", "lastKnownIP":
```

GET PowerShell Sample

```
# Get sites
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites" -Method 'GET' -Headers
$authHeader
$siteID = $siteList.data[0].id

###GET Remote Agents
Write-Output "Remote Agents"
Write-Output "----"
$ramAgentList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/ram/agents"
-Method 'GET' -Headers $authHeader
Write-Output $ramAgentList | ConvertTo-Json
foreach ($ramAgentList in $ramAgentList.data)
{
    Write-Output ("Remote Agent  $($ramAgentList.id):
$($ramAgentList.attributes.name)")
}
```

Endpoint: /sites/{siteId}/ram/agents/{remoteAgentID}

Note: Available as of EFT 8.1.0

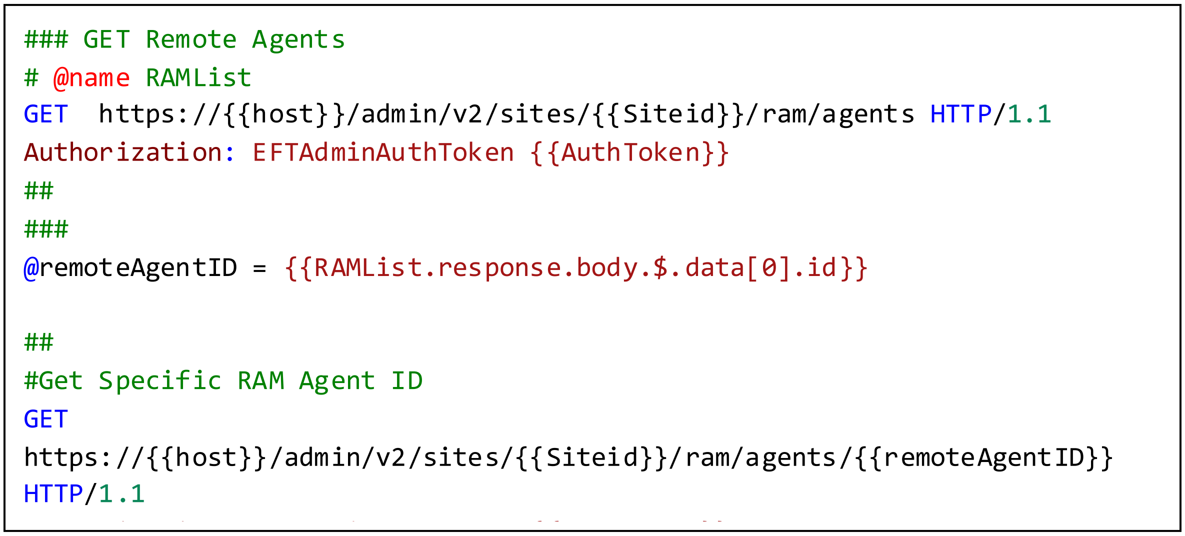
Returns a remote agent’s settings from a specified site.

- GET

Method & URL

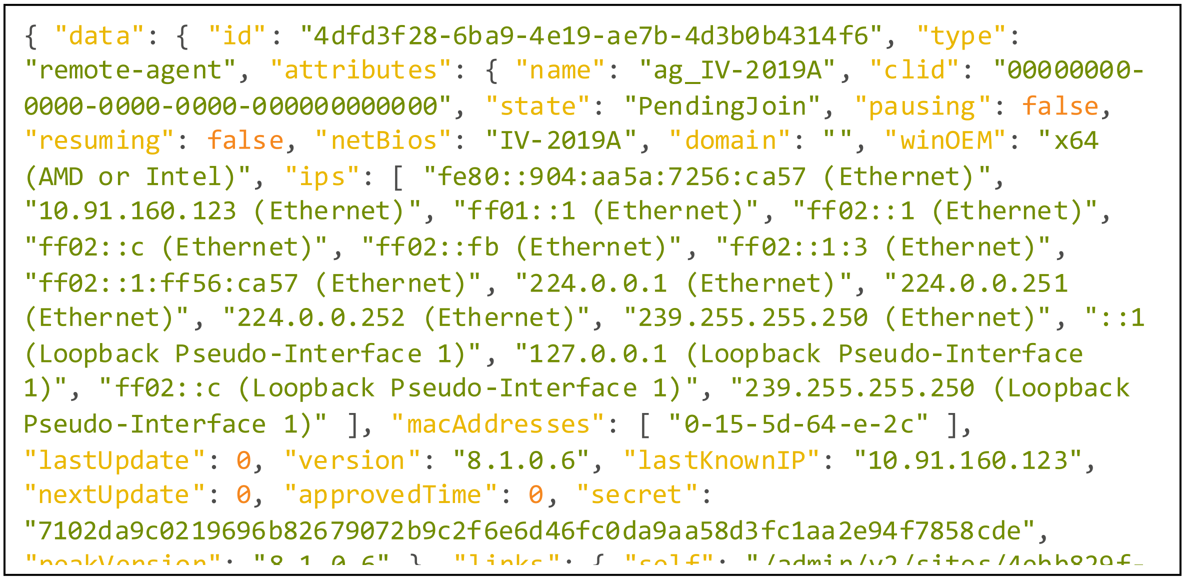
GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/agents/{remoteAgentID}

GET Body Sample



```
### GET Remote Agents
# @name RAMList
GET  https://{{host}}/admin/v2/sites/{{Siteid}}/ram/agents HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
##
###
@remoteAgentID = {{RAMList.response.body.$.data[0].id}}

##
#Get Specific RAM Agent ID
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/ram/agents/{{remoteAgentID}}
HTTP/1.1
```

GET Response Body Sample



```
{ "data": { "id": "4dfd3f28-6ba9-4e19-ae7b-4d3b0b4314f6", "type":
"remote-agent", "attributes": { "name": "ag_IV-2019A", "clid": "00000000-
0000-0000-0000-000000000000", "state": "PendingJoin", "pausing": false,
"resuming": false, "netBios": "IV-2019A", "domain": "", "winOEM": "x64
(AMD or Intel)", "ips": [ "fe80::904:aa5a:7256:ca57 (Ethernet)",
"10.91.160.123 (Ethernet)", "ff01::1 (Ethernet)", "ff02::1 (Ethernet)",
"ff02::c (Ethernet)", "ff02::fb (Ethernet)", "ff02::1:3 (Ethernet)",
"ff02::1:ff56:ca57 (Ethernet)", "224.0.0.1 (Ethernet)", "224.0.0.251
(Ethernet)", "224.0.0.252 (Ethernet)", "239.255.255.250 (Ethernet)", "::1
(Loopback Pseudo-Interface 1)", "127.0.0.1 (Loopback Pseudo-Interface
1)", "ff02::c (Loopback Pseudo-Interface 1)", "239.255.255.250 (Loopback
Pseudo-Interface 1)" ], "macAddresses": [ "0-15-5d-64-e-2c" ],
"lastUpdate": 0, "version": "8.1.0.6", "lastKnownIP": "10.91.160.123",
"nextUpdate": 0, "approvedTime": 0, "secret":
"7102da9c0219696b82679072b9c2f6e6d46fc0da9aa58d3fc1aa2e94f7858cde",
"peakVersion": "8.1.0.6" }, "links": { "self": "/admin/v2/sites/4ebb829f-
```

GET PowerShell Sample

```
###GET Remote Agents
Write-Output "Remote Agents"
Write-Output "----"
$ramAgentList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/ram/agents"
-Method 'GET' -Headers $authHeader
$ramAgentID = $ramAgentList.data[0].id

###GET Remote Agent Settings
Write-Output "GET Specific Remote Agent Settings: " $ramAgentID
Write-Output "----"
$ramAgent = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/agents/$ramAgentID" -Method 'GET' -Headers
$authHeader
Write-Output $ramAgent | ConvertTo-Json
```

Endpoint: /sites/{siteId}/ram/agents/templates

Note: Available as of EFT 8.1.0

Returns a remote agent template list from a specified site or allows the creation of a template on a specific EFT site.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/templates

GET Body Sample

```
#########################
#RAM TEMPLATES
GET https://{{host}}/admin/v2/sites/{{Siteid}}/ram/templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": [ { "id": "04cd2f6f-dea6-4674-bcde-3c7fe5a800a1", "type":
"remote-agents-template", "attributes": { "name": "RAM Template1",
"updateInterval": "EveryFiveMinutes", "eventRuleIds": [ null, "d352632d-
5f3f-49bf-99af-849f7453dd0a" ], "autoEnroll": false, "autoEnrollDate":
1659757989, "homeFolder": "/Usr/Agents/%AgentNetBIOSName%/",
"updateFailure": "Stop_Disable_Unenroll", "enableIPAccess": false,
"ipAccess": { "masks": [ { "ipMask": "0.0.0.0", "reason": "",
"matchAction": "Deny", "timeStamp": 0, "autobanEntry": true } ],
"autoban": [], "defaultAction": "Allow" } }, "links": { "self":
"/admin/v2/sites/4ebb829f-f84c-4355-8d55-
d43a2852ca12/ram/templates/04cd2f6f-dea6-4674-bcde-3c7fe5a800a1" } }, {
"id": "b672dec5-567e-4baa-bc3f-5be948802b42", "type": "remote-agents-
template", "attributes": { "name": "RAM Template 2", "updateInterval":
```

GET PowerShell Sample

```
###############
##GET Remote Agent Templates
Write-Output "Remote Agent Templates"
Write-Output "----"
$ramTemplateList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates" -Method 'GET' -Headers $authHeader
Write-Output $ramTemplateList | ConvertTo-Json -Depth 10
foreach ($ramTemplateList in $ramTemplateList.data)
{
    Write-Output ("Remote Agent Template  $($ramTemplateList.id):
$($ramTemplateList.attributes.name)")
}
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/templates

POST Body Sample

```
### Add New RAM Template ###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/ram/templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "name": "NewRAMTemplateName",
      "updateInterval": "EveryFiveMinutes",
      "eventRuleIds": [
        "d352632d-5f3f-49bf-99af-849f7453dd0a"
      ],
      "autoEnroll": true,
      "autoEnrollDate": 0,
      "homeFolder": "/Usr/Agents/%AgentNetBIOSName%/",
      "updateFailure": "Stop_Disable_Unenroll",
      "enableIPAccess": true,
      "ipAccess": {
        "masks": [
          {
            "ipMask": "0.0.0.0",
            "reason": "",
            "matchAction": "Deny",
            "timeStamp": 0,
            "autobanEntry": true
          }
        ],
        "defaultAction": "Allow"
      }
    }
  }
}
```

POST Response Body Sample

```
{ "data": { "id": "b3ed1f50-ce61-479f-ba48-094cee3c13fd", "type":
"remote-agents-template", "attributes": { "name": "NewRAMTemplateName",
"updateInterval": "EveryFiveMinutes", "eventRuleIds": [ null, "d352632d-
5f3f-49bf-99af-849f7453dd0a" ], "autoEnroll": true, "autoEnrollDate": 0,
"homeFolder": "/Usr/Agents/%AgentNetBIOSName%/", "updateFailure":
"Stop_Disable_Unenroll", "enableIPAccess": true, "ipAccess": { "masks": [
{ "ipMask": "0.0.0.0", "reason": "", "matchAction": "Deny", "timeStamp":
0, "autobanEntry": false } ], "autoban": [], "defaultAction": "Allow" }
}, "links": { "self": "/admin/v2/sites/4ebb829f-f84c-4355-8d55-
d43a2852ca12/ram/templates/b3ed1f50-ce61-479f-ba48-094cee3c13fd" } } }
```

POST PowerShell Sample

```
#######
###
#POST New RAM Template
Write-Output "POST RAM Template"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes": {
      "name": "NewRAMTemplateName",
      "updateInterval": "EveryFiveMinutes",
      "eventRuleIds": [
        "d352632d-5f3f-49bf-99af-849f7453dd0a"
      ],
      "autoEnroll": true,
      "autoEnrollDate": 0,
      "homeFolder": "/Usr/Agents/%AgentNetBIOSName%/",
      "updateFailure": "Stop_Disable_Unenroll",
      "enableIPAccess": true,
      "ipAccess": {
        "masks": [
          {
            "ipMask": "0.0.0.0",
            "reason": "",
            "matchAction": "Deny",
            "timeStamp": 0,
            "autobanEntry": true
          }
        ],
        "defaultAction": "Allow"
      }
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates" -Method 'POST' -Headers $authHeader
-Body $update
```

Endpoint: /sites/{siteId}/ram/agents/templates/{remoteAgentsTemplateID}

Note: Available as of EFT 8.1.0

Returns a remote agent settings template from a specified site, update/patch a remote agent settings template, or allows the deletion of a remote agent settings template on a specific EFT site.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/templates/{remoteAgentsTemplateID}

GET Body Sample

```
############################
#GET RAM Template by ID
#RAM TEMPLATES
# @name RAMTemplates
GET https://{{host}}/admin/v2/sites/{{Siteid}}/ram/templates HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
@ramTemplateID = {{RAMTemplates.response.body.$.data[0].id}}


###
#GET Template by Template ID
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/ram/templates/{{ramTemplateID}
} HTTP/1.1
```

GET Response Body Sample

```
{ "data": { "id": "04cd2f6f-dea6-4674-bcde-3c7fe5a800a1", "type":
"remote-agents-template", "attributes": { "name": "RAM Template1",
"updateInterval": "EveryFiveMinutes", "eventRuleIds": [], "autoEnroll":
false, "autoEnrollDate": 1659757989, "homeFolder":
"/Usr/Agents/%AgentNetBIOSName%/", "updateFailure":
"Stop_Disable_Unenroll", "enableIPAccess": false, "ipAccess": { "masks":
[ { "ipMask": "0.0.0.0", "reason": "", "matchAction": "Deny",
"timeStamp": 0, "autobanEntry": false } ], "autoban": [],
"defaultAction": "Allow" } }, "links": { "self":
"/admin/v2/sites/4ebb829f-f84c-4355-8d55-
```

GET PowerShell Sample

```
###############
##GET Remote Agent Templates
$ramTemplateList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates" -Method 'GET' -Headers $authHeader
$ramTemplateID = $ramTemplateList.data[0].id

####
#GET Remote Agent Template by ID
Write-Output "Remote Agent Template by ID: " $ramTemplateID
Write-Output "----"
$ramTemplate = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates/$ramTemplateID" -Method 'GET' -
Headers $authHeader
Write-Output $ramTemplate | ConvertTo-Json -Depth 10
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/templates/{remoteAgentsTemplateID}

PATCH Body Sample

```
####
#PATCH Template by Template ID
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/ram/templates/{{ramTemplateID}
} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
        "ipAccess": {
        "masks": [
          {
            "ipMask": "0.0.0.0",
            "reason": "",
            "matchAction": "Deny",
            "timeStamp": 0,
            "autobanEntry": false
          }
        ],
        "defaultAction": "Allow"
      }
    }
  }
`
```

PATCH Response Body Sample

```
{ "data": { "id": "04cd2f6f-dea6-4674-bcde-3c7fe5a800a1", "type":
"remote-agents-template", "attributes": { "name": "RAM Template1",
"updateInterval": "EveryFiveMinutes", "eventRuleIds": [], "autoEnroll":
false, "autoEnrollDate": 1659757989, "homeFolder":
"/Usr/Agents/%AgentNetBIOSName%/", "updateFailure":
"Stop_Disable_Unenroll", "enableIPAccess": false, "ipAccess": { "masks":
[ { "ipMask": "0.0.0.0", "reason": "", "matchAction": "Deny",
"timeStamp": 0, "autobanEntry": false } ], "autoban": [],
"defaultAction": "Allow" } }, "links": { "self":
```

PATCH PowerShell Sample

```
##########
##GET Remote Agent Template ID
$ramTemplateList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates" -Method 'GET' -Headers $authHeader
$ramTemplateID = $ramTemplateList.data[0].id

###
##Patch Remote Agent Template by ID
Write-Output "Remote Agent Template Patch (update)"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes": {
        "ipAccess": {
        "masks": [
          {
            "ipMask": "1.1.1.1",
            "reason": "",
            "matchAction": "Deny",
            "timeStamp": 0,
            "autobanEntry": false
          }
        ],
        "defaultAction": "Allow"
      }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates/$ramTemplateID" -Method 'PATCH' -
Headers $authHeader -Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/ram/templates/{remoteAgentsTemplateID}

DELETE Body Sample

```
####
#DELETE Remote Agent Template
#Template must not have existing associated Remote Agents
DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/ram/templates/{{ramTemplateID}
} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Sun, 24 Jul 2022 02:05:46 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
####################################
##########
##GET Remote Agent Template ID
$ramTemplateList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates" -Method 'GET' -Headers $authHeader
$ramTemplateID = $ramTemplateList.data[1].id

####
#DELETE Remote Agent Template by ID
$ramTemplateList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/ram/templates/$ramTemplateID" -Method 'DELETE' -
Headers $authHeader
```

Endpoint:/sites/{siteId}/awe-tasks

Note: Available as of EFT 8.1.0

(GET) Returns advanced workflow task list from specified site and (POST) creates advanced workflow task for specified site

- GET
- POST

Method & URL

GET https://{{host}}/admin/v2/sites/{{SiteId}}/awe-tasks

GET Body Sample

```
### GET AWE TASKS ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":[{"id":"0f340ef3-3628-5844-bb6b-
c659252b6b44","type":"aweTask","attributes":{"name":"Sample - Date Format
Functions","created":"2022-08-
01","description":""},"relationships":{"aweFolder":{"data":{"id":"0000000
0-0000-0000-0000-
000000000000","name":"<root>"}}},"links":{"self":"/admin/v2/sites/b65892e
9-399d-4447-a3b1-477c08f0abca/awe-tasks/0f340ef3-3628-5844-bb6b-
c659252b6b44"}},{"id":"92ac1ba6-5fcc-5b85-b5da-
e1b3c4d4a300","type":"aweTask","attributes":{"name":"Sample - Environment
Variables","created":"2022-08-
01","description":""},"relationships":{"aweFolder":{"data":{"id":"0000000
0-0000-0000-0000-
000000000000","name":"<root>"}}},"links":{"self":"/admin/v2/sites/b65892e
9-399d-4447-a3b1-477c08f0abca/awe-tasks/92ac1ba6-5fcc-5b85-b5da-
e1b3c4d4a300"}},{"id":"432c99b4-fcf3-5472-981d-
416946a9b95a","type":"aweTask","attributes":{"name":"Sample - Excel
Actions","created":"2022-08-
01","description":""},"relationships":{"aweFolder":{"data":{"id":"0000000
0-0000-0000-0000-
000000000000","name":"<root>"}}},"links":{"self":"/admin/v2/sites/b65892e
9-399d-4447-a3b1-477c08f0abca/awe-tasks/432c99b4-fcf3-5472-981d
```

GET PowerShell Sample

```
# get awe-tasks
Write-Output "AWE-TASKS"
Write-Output "-----"
$aweList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks" -
Method 'GET' -Headers $authHeader
Write-Output $aweList | ConvertTo-Json
foreach ($awetask in $aweList.data)
{
    Write-Output ("Task ID $($aweTask.id): Name ->
$($aweTask.attributes.name)")
}
```

POST https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks

POST Body Sample

```
POST https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "aweTask",
    "attributes": {

"code":"PEFNVEFTSz4NCjxBTVRBU0tIRUFEPg0KICAgIDxUQVNLSU5GTyBUQVNLVkVSU0lPT
j0iMTAuNy4xMDAuMSIgLz4NCjwvQU1UQVNLSEVBRD4NCg0KPEFNRlVOQ1RJT04gTkFNRT0iTW
FpbiIgQUNDRVNTPSJwcml2YXRlIiBSRVRVUk5UWVBFPSJ2YXJpYWJsZSI+DQo8IS0tIC0tLS0
tLS0tLS0gLS0+DQo8IS0tIEdFTkVSQUwgVEFTSyBOT1RFUzogIFRoaXMgdGFzayB3aWxsIG91
dHB1dCBhbGwgc3VwcG9ydGVkIGVudmlyb25tZW50IHZhcmlhYmxlcyB0byBhIHRleHQgZmlsZ
S4gWW91IGNhbiB1c2UgdGhlc2UgdmFyaWFibGVzIGluc2lkZSBvZiBBV0UgdGFza3Mgd2hlcm
UgbmVlZGVkLg0KDQpUaGUgZmlsZSB3aWxsIGJlIGNhbGVsZCAiZW52aXJvbm1lbnQgdmFyaWF
ibGVzIiBhbmQgcGxhY2VkIG9uIIHRoZSByb290IG9mIHlvdXIgZHJpdmUgKHVzdWFsbHkgYzpc
KS4gLS0+DQo8IS0tIC0tLS0tLS0tLS0gLS0+DQo8QU1FTlZZWQVJTIEFDVElWSVRZPSJsaXN0I
iBSRVNVTFREQVRBU0VUPSJFbnZybXRWYXJEYXRhc2V0IiBUWVBFPSJtYWNoaW5lIiAvPg0KIC
AgICA8QU1GSUxFU1lTTEVNIEFDVElWSVRZPSJ3cml0ZV9maWxlIiBGSUxFPSIlU3lzdGVtRHJ
```

POST Response Body Sample

```
{"data":{"id":"8405b064-aa9a-46b8-b0b5-
4c60eb3d3e09","type":"aweTask","attributes":{"name":"MyNewAWEScript","cre
ated":"2022-08-01","description":"Hello,
world!","useTimeout":true,"timeout":10,"logLevel":"Verbose","logDirectory
":"","retainSuccessfulLogs":false,"retainFailedLogs":true,"code":"PEFNVEF
TSz4NCjxBTVRBU0tIRUFEPg0KICAgIDxUQVNLSU5GTyBUQVNLVkVSU0lPTj0iMTAuNy4xMDAu
MSIgLz4NCjwvQU1UQVNLSEVBRD4NCg0KPEFNRlVOQ1RJT04gTkFNRT0iTWFpbiIgQUNDRVNTP
SJwcml2YXRlIiBSRVRVUk5UWVBFPSJ2YXJpYWJsZSI+DQo8IS0tIC0tLS0tLS0tLS0gLS0+DQ
o8IS0tIEdFTkVSQUwgVEFTSyBOT1RFUzogIFRoaXMgdGFzayB3aWxsIG91dHB1dCBhbGwgc3V
wcG9ydGVkIGVudmlyb25tZW50IHZhcmlhYmxlcyB0byBhIHRleHQgZmlsZS4gWW91IGNhbiB1
c2UgdGhlc2UgdmFyaWFibGVzIGluc2lkZSBvZiBBV0UgdGFza3Mgd2hlcmUgbmVlZGVkLg0KD
QpUaGUgZmlsZSB3aWxsIGJlIGNhbGVsZCAiZW52aXJvbm1lbnQgdmFyaWFibGVzIiBhbmQgcG
xhY2VkIG9uIIHRoZSByb290IG9mIHlvdXIgZHJpdmUgKHVzdWFsbHkgYzpcKS4gLS0+DQo8IS0
tIC0tLS0tLS0tLS0gLS0+DQo8QU1FTlZZWQVJTIEFDVElWSVRZPSJsaXN0IiBSRVNVTFREQVRB
U0VUPSJFbnZybXRWYXJEYXRhc2V0IiBUWVBFPSJtYWNoaW5lIiAvPg0KICAgICA8QU1GSUxFU
```

POST PowerShell Sample

```
##POST SITE AWE TASKS
Write-Output "Site AWE TASK POST"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "aweTask",
    "attributes": {

"code":"PEFNVEFTSz4NCjxBTVRBU0tIRUFEPg0KICAgIDxUQVNLSU5GTyBUQVNLVkVSU0lPTj0iMT
AuNy4xMDAuMSIgLz4NCjwvQU1UQVNLSEVBRD4NCg0KPEFNRlVOQ1RJT04gTkFNRT0iTWFpbiIgQUND
RVNTPSJwcml2YXRlIiBSRVRVUk5UWVBFPSJ2YXJpYWJsZSI+DQo8IS0tIC0tLS0tLS0tLS0gLS0+DQ
o8IS0tIEdFTkVSQUwgVEFTSyBOT1RFUzogIFRoaXMgdGFzayB3aWxsIG91dHB1dCBhbGwgc3VwcG9y
dGVkIGVudmlyb25tZW50IHZhcmlhYmxlcyB0byBhIHRleHQgZmlsZS4gWW91IGNhbiB1c2UgdGhlc2
UgdmFyaWFibGVzIGluc2lkZSBvZiBBV0UgdGFza3Mgd2hlcmUgbmVlZGVkLg0KDQpUaGUgZmlsZSB3
aWxsIGJlIGNhbGVsZCAiZW52aXJvbm1lbnQgdmFyaWFibGVzIiBhbmQgcGxhY2VkIG9uIHRoZSByb2
90IG9mIHlvdXIgZHJpdmUgKHVzdWFsbHkgYzpcKS4gLS0+DQo8IS0tIC0tLS0tLS0tLS0gLS0+DQo8
QU1FTlZWQVJTIEFDVElWSVRZPSJsaXN0IiBSRVNVTFREQVRBU0VUPSJFbnZybXRWYXJEYXRhc2V0Ii
BUWVBFPSJtYWNoaW5lIiAvPg0KICAgICA8QU1GSUxFU1lTVEVNIEFDVElWSVRZPSJ3cml0ZV9maWxl
IiBGSUxFPSIlU3lzdGVtRHJpdmUlXGVudmlyb25tZW50X3ZhcmlhYmxlcyI+RW52aXJvbm1lbnQgdm
FyaWFibGVzIGZvciBtYWNoaW5lIChzeXN0ZW0pOg0KDQo8L0FNRklMRVNZU1RFTT4NCiAgICAgPEFN
TE9PUCBBQ1RJVklUWT0iZGF0YXNldCIgREFUQVNFVD0iRW52cm10VmFyRGF0YXNldCI+DQogICAgIC
AgICAgPEFNRklMRVNZU1RFTSBBQ1RJVklUWT0id3JpdGVfZmlsZSIgRklMRT0iYzpcZW52aXJvbm1l
bnRfdmFyaWFibGVzIj4lRW52cm10VmFyRGF0YXNldC5WYXJOYW1lJSAlRW52cm10VmFyRGF0YXNldC
5WYXJWYWx1ZSU8L0FNRklMRVNZU1RFTT4NCiAgICAgPC9BTUxPT1A+DQo8QU1FTlZWQVJTIEFDVElW
SVRZPSJsaXN0IiBSRVNVTFREQVRBU0VUPSJFbnZybXRWYXJEYXRhc2V0IiBUWVBFPSJ1c2VyIiAvPg
0KICAgICA8QU1GSUxFU1lTVEVNIEFDVElWSVRZPSJ3cml0ZV9maWxlIiBGSUxFPSIlU3lzdGVtRHJp
dmUlXGVudmlyb25tZW50X3ZhcmlhYmxlcyI+DQpFbnZpcm9ubWVudCB2YXJpYWJsZXMgZm9yIFVzZX
```

Endpoint:/sites/{siteId}/awe-tasks/{aweTaskID}

Note: Available as of EFT 8.1.0

(GET) Returns advanced workflow task from specified site, (PATCH) Updates advanced workflow task settings, and (DELETE) Delete advanced workflow task

- GET
- PATCH
- DELETE

Method & URL

GET https://{{host}}/admin/v2/sites/{{SiteId}}/awe-tasks/{aweTaskID}

GET Body Sample

```
### GET A SPECIFIC AWE TASK
@aweID = {{aweList.response.body.$.data[0].id}}
GET https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks/{{aweID}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":{"id":"0f340ef3-3628-5844-bb6b-
c659252b6b44","type":"aweTask","attributes":{"name":"Sample - Date Format
Functions","created":"2022-08-
01","description":"","useTimeout":true,"timeout":120,"logLevel":"None","l
ogDirectory":"C:\\ProgramData\\Globalscape\\EFT
Server\\AWE\\Temp\\","retainSuccessfulLogs":true,"retainFailedLogs":true,
"code":"PEFNVEFTSz4NCjxBTVRBU0tIRUFEPg0KICAgIDxUQVNLSU5GTyBUQVNLVkVSU0lPT
j0iMTAuNy4xMDAuMSIgLz4NCjwvQU1UQVNLSEVBRD4NCg0KPEFNRlVOQ1RJT04gTkFNRT0iTW
FpbiIgQUNDRVNTPSJwcml2YXRlIiBSRVRVUk5UWVBFPSJ2YXJpYWJsZSI+DQo8IS0tIC0tLS0
tLS0tLS0gLS0+DQo8IS0tIEdFTkVSQUwgVEFTSyBOT1RFUzogIFRoaXMgdGFzayByZXR1cm5z
IHRoZSBjdXJyZW50IHN5c3RlbSB0aW1lIGluIGEgbnVtYmVyIG9mIGRpZmZlcmVudCBmb3JtY
XRzIHVzaW5nIHRoZSBGb3JtYXQoKSBmdW5jdGlvbi4gLS0+DQo8IS0tIC0tLS0tLS0tLS0gLS
```

GET PowerShell Sample

```
# get awe-tasks
Write-Output "AWE-TASKS"
Write-Output "-----"
$aweList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks" -
Method 'GET' -Headers $authHeader
Write-Output $aweList | ConvertTo-Json
foreach ($awetask in $aweList.data)
{
    Write-Output ("Task ID $($aweTask.id): Name ->
$($aweTask.attributes.name)")
}

$aweID = $aweList.data[0].id
$awsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks/$aweID"
-Method 'GET' -Headers $authHeader
Write-Output $awsList | ConvertTo-Json
```

PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks/{{aweID}}

PATCH Body Sample

```
### UPDATE A SPECIFIC AWE TASK (this is using the environment var code)
### for your site you may have to adjust the index number
@aweID = {{aweList.response.body.$.data[14].id}}
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks/{{aweID}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
  "data": {
    "type": "aweTask",
    "attributes": {
      "logDirectory": "C\\temp\\"
    }
  }
}
```

PATCH Response Body Sample

```
{"data":{"id":"8405b064-aa9a-46b8-b0b5-
4c60eb3d3e09","type":"aweTask","attributes":{"name":"MyNewAWEScript","cre
ated":"2022-08-01","description":"Hello,
world!","useTimeout":true,"timeout":10,"logLevel":"Verbose","logDirectory
":"C\\temp\\","retainSuccessfulLogs":false,"retainFailedLogs":true,"code"
:"PEFNVEFTSz4NCjxBTVRBU0tIRUFEPg0KICAgIDxUQVNLSU5GTyBUQVNLVkVSU0lPTj0iMTA
uNy4xMDAuMSIgLz4NCjwvQU1UQVNLSEVBRD4NCg0KPEFNRlVOQ1RJT04gTkFNRT0iTWFpbiIg
QUNDRVNTPSJwcml2YXRlIiBSRVRVUk5UWVBFPSJ2YXJpYWJsZSI+DQo8IS0tIC0tLS0tLS0tL
S0gLS0+DQo8IS0tIEdFTkVSQUwgVEFTSyBOT1RFUzogIFRoaXMgdGFzayB3aWxsIG91dHB1dC
BhbGwgc3VwcG9ydGVkIGVudmlyb25tZW50IHZhcmlhYmxlcyB0byBhIHRleHQgZmlsZS4gWW9
1IGNhbiB1c2UgdGhlc2UgdmFyaWFibGVzIGluc2lkZSBvZiBBV0UgdGFza3Mgd2hlcmUgbmVl
ZGVkLg0KDQpUaGUgZmlsZSB3aWxsIGJlIGNhbGVkZCAiZW52aXJvbm1lbnQgdmFyaWFibGVzI
```

PATCH PowerShell Sample

```
##PATCH SITE AWE TASKS
Write-Output "Site specific AWE TASK PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "aweTask",
    "attributes": {
      "description": "Well Hello",
      "useTimeout": true,
      "timeout": 10,
      "logLevel": "Verbose",
      "retainSuccessfulLogs": true,
      "retainFailedLogs": true
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchAWETASKReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-
tasks/$aweID" -Method 'PATCH' -Headers $authHeader -Body $update
```

DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks/{{aweID}}

DELETE Body Sample

```
### DELETE A SPECIFIC AWE TASK
@aweID = {{aweList.response.body.$.data[0].id}}

DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks/{{aweID}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content

Date: Mon, 01 Aug 2022 21:16:00 GMT

Content-Length: 0

Cache-Control: no-cache, no-store, must-revalidate

Expires: -1

X-Content-Type-Options: nosniff

X-XSS-Protection: 1; mode=block

Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval'
data:;
```

DELETE PowerShell Sample

```
$aweID = $aweList.data[0].id
$awsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks/$aweID"
-Method 'DELETE' -Headers $authHeader
Write-Output $awsList | ConvertTo-Json
```

Endpoint:/sites/{siteId}/awe-tasks-folders

Note: Available as of EFT 8.1.0

(GET) Returns advanced workflow folder list from specified site and (POST) create advanced workflow folder for specified site

- GET
- POST

Method & URL

GET https://{{host}}/admin/v2/sites/{{Siteid}}/awe-task-folders

GET Body Sample

```
### GET ALL AWE TASK FOLDERS ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-folders HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":[{"id":"00000000-0000-0000-0000-
000000000000","type":"aweFolder","attributes":{"name":"<root>","tasks":[{
"id":"92ac1ba6-5fcc-5b85-b5da-e1b3c4d4a300","name":"Sample - Environment
Variables","created":"2022-08-01","description":""},{"id":"432c99b4-fcf3-
5472-981d-416946a9b95a","name":"Sample - Excel Actions","created":"2022-
08-01","description":""},{"id":"df5f43bb-1f34-57be-9e59-
75bd272ba408","name":"Sample - Expectations","created":"2022-08-
01","description":""},{"id":"cf774000-2d0a-5962-a060-
111aa4d94d1c","name":"Sample - File Date Operations","created":"2022-08-
01","description":""},{"id":"40536ad8-327e-53d6-8566-
0d7722156b42","name":"Sample - FTP Actions","created":"2022-08-
01","description":""},{"id":"a6a79232-f9e5-5a14-b944-
```

GET PowerShell Sample

```
####Added 2/01/2020
### Returns all site auto ban list###
Write-Output "AWE Task Folders"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks-
folders" -Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

POST https://{{host}}/admin/v2/sites/{{Siteid}}/awe-task-folders

POST Body Sample

```
### create a folder named AWE_Folder_Three
###

###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-folders
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "aweTaskFolder",
    "attributes": {
      "name": "AWE_Folder_Three"
    }
  }
}
"""
```

POST Response Body Sample

```
{"data":{"id":"dc70ad42-47ad-4239-a81a-
fe9d650820aa","type":"aweFolder","attributes":{"name":"AWE_Folder_Three",
"tasks":[]},"links":{"self":"/admin/v2/sites/b65892e9-399d-4447-a3b1-
477c08f0abca/awe-tasks-folders/dc70ad42-47ad-4239-a81a-
fe9d650820aa"}},"links":{"self":"/admin/v2/sites/b65892e9-399d-4447-a3b1-
477c08f0abca/awe-tasks-folders"}}
```

POST PowerShell Sample

```
####Added 2/01/2020
### Returns all site auto ban list###
Write-Output "AWE Task Folders"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks-
folders" -Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json

##POST SITE AWE TASKS FOLDERS
Write-Output "Site AWE TASK FOLDER POST"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "aweTaskFolder",
    "attributes": {
      "name": "AWE_Folder_FIVE-O"
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$postAWETASKReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-
tasks-folders" -Method 'POST' -Headers $authHeader -Body $update
Write-Output $postAWETASKReturn | ConvertTo-Json
```

Endpoint:/sites/{sitetId}/awe-tasks-folders/{aweTaskFolderID}

Note: Available as of EFT 8.1.0

(GET) Returns advanced workflow folder from specified site, (PATCH) Updates advanced workflow folder settings, and (DELETE) Delete advanced workflow folder.

- GET
- PATCH
- DELETE

Method & URL

GET https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-folders/{{AWEId}}

GET Body Sample

```
### GET A SPECIFIC AWE TASK FOLDERS ###
@AWEId = 859dc605-0600-42ab-a1fa-f23f5a19e8cd
GET https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-folders/{{AWEId}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":{"id":"859dc605-0600-42ab-a1fa-
f23f5a19e8cd","type":"aweFolder","attributes":{"name":"AWE _
Folder_One","tasks":[]},"links":{"self":"/admin/v2/sites/b65892e9-399d-
4447-a3b1-477c08f0abca/awe-tasks-folders/859dc605-0600-42ab-a1fa-
f23f5a19e8cd"}},"links":{"self":"/admin/v2/sites/b65892e9-399d-4447-a3b1-
477c08f0abca/awe-tasks-folders"}}
```

GET PowerShell Sample

```
####Added 2/01/2020
### Returns awe folder information###
Write-Output "AWE Task Folders"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks-
folders" -Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json

$aweID = $siteList.data[1].id
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-tasks-
folders/$aweID" -Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-folders/{{AWEId}}

PATCH Body Sample

```
@AWEId = dc70ad42-47ad-4239-a81a-fe9d650820aa
### PATCH AWE TASK FOLDERS ###
PATCH  https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-
folders/{{AWEId}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "id": "dc70ad42-47ad-4239-a81a-fe9d650820aa",
    "attributes": {
      "name": "NewFirstFolder"
    },
    "type": "aweTaskFolder"
  }
}
```

PATCH Response Body Sample

```
{"data":{"id":"dc70ad42-47ad-4239-a81a-
fe9d650820aa","type":"aweFolder","attributes":{"name":"NewFirstFolder","t
asks":[{"id":"432c99b4-fcf3-5472-981d-416946a9b95a","name":"Sample -
Excel Actions","created":"2022-08-01","description":"My
Description"},{"id":"40536ad8-327e-53d6-8566-0d7722156b42","name":"Sample
- FTP Actions","created":"2022-08-
01","description":""}]},"links":{"self":"/admin/v2/sites/b65892e9-399d-
4447-a3b1-477c08f0abca/awe-tasks-folders/dc70ad42-47ad-4239-a81a-
fe9d650820aa"}},"links":{"self":"/admin/v2/sites/b65892e9-399d-4447-a3b1-
```

PATCH PowerShell Sample

```
$folderID = $aweFolderList.data[1].id

##POST SITE AWE TASKS FOLDERS
Write-Output "Site AWE TASK FOLDER PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "aweTaskFolder",
    "attributes": {
      "name": "TEST FOLDER TWO"
    }
  }
}'

$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchAWETASKReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-
tasks-folders/$folderID" -Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $patchAWETASKReturn | ConvertTo-Json
```

DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/awe-tasks-folders/{{AWEId}}

NOTE: The folder should be emptied before it is deleted else the EFT will return a HTTP 500 Server error.

DELETE Body Sample

```
### AWEId will be the id of a folder to be deleted.
@AWEId = e1d62bf1-8bdd-4e82-8efd-395467069306
### DELETE AWE TASK FOLDERS ###
DELETE  https://{{host}}/admin/v2/sites/{{Siteid}}/awe-task-
folders/{{AWEId}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content
Date: Tue, 02 Aug 2022 17:47:50 GMT
Content-Length: 0
Cache-Control: no-cache, no-store, must-revalidate
Expires: -1
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval'
data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
$folderID = $aweFolderList.data[1].id

$deleteAWETASKReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/awe-
tasks-folders/$folderID" -Method 'DELETE' -Headers $authHeader
Write-Output $deleteAWETASKReturn | ConvertTo-Json
```

Endpoint: /sites/{siteId}/connection-profiles

Note: Available as of EFT 8.1.0

(GET) Returns connection profiles list from the specified site, (POST) create a connection profile for a site.

- GET
- POST

Method & URL

```
GET https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles
```

GET Body Sample

```
### GET a list of current connection profiles####
GET https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Thu, 04 Aug 2022 14:17:24 GMT Location:
/admin/v2/sites/76882fc7-3f55-4040-ae5f-2e645c12647b/connection-profiles
Content-Type: application/vnd.api+json Content-Length: 1383 Cache-Control:
no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-Options:
nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy: default-src
'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer {
"data": [ { "id": "c9ea36ec-8a9e-440f-8c21-920eb3927732", "type":
"connectionProfile", "attributes": { "type": "Protocol", "name": "POST
REQUEST Create connection Profile", "description": "Connection Profile
Details", "connectionSettings": { "protocol": "FTPSAuthTLS", "user": "AFAR",
"password": "UpKUJq1sD4Yfb1Hz8ZFiRw==", "autoLogin": true, "testPath":
```

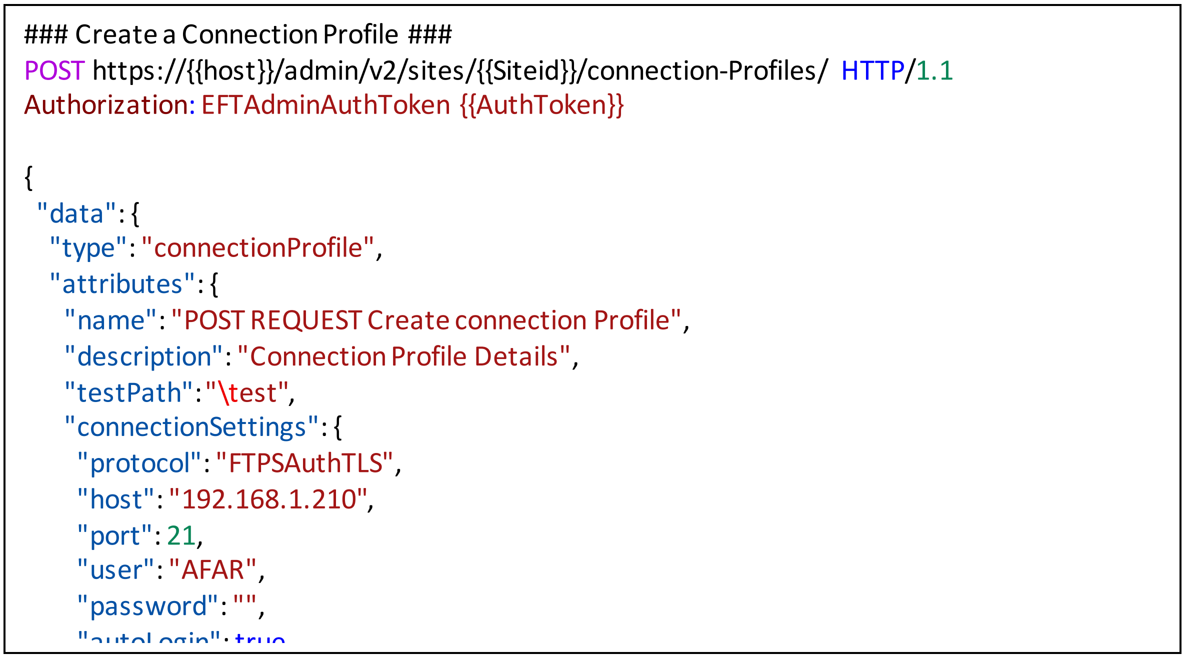
GET PowerShell Sample

```
####################################
###GET list of Connection Profiles###
Write-Output "GET list of Connection Profiles"
Write-Output "------------------------------"
#$siteID = "Enter site ID"
$CPlist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/connection-
Profiles" -Method 'GET' -Headers $authHeader
Write-Output $CPlist | ConvertTo-Json -Depth 25

$profileId = $CPList.data[0].id
```

Method & URL

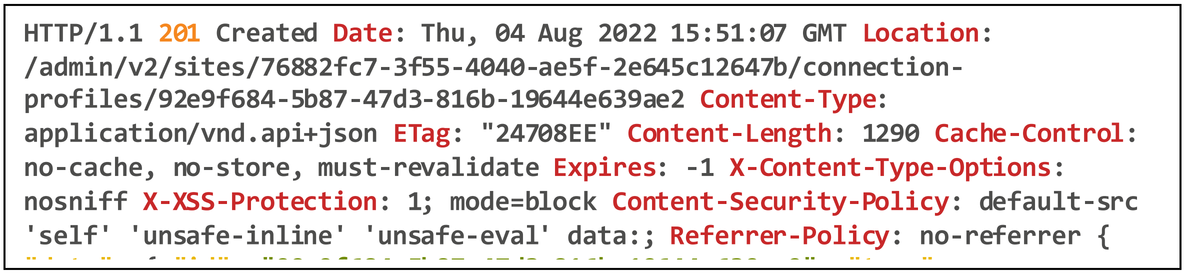
```
POST https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles
```

POST Body Sample



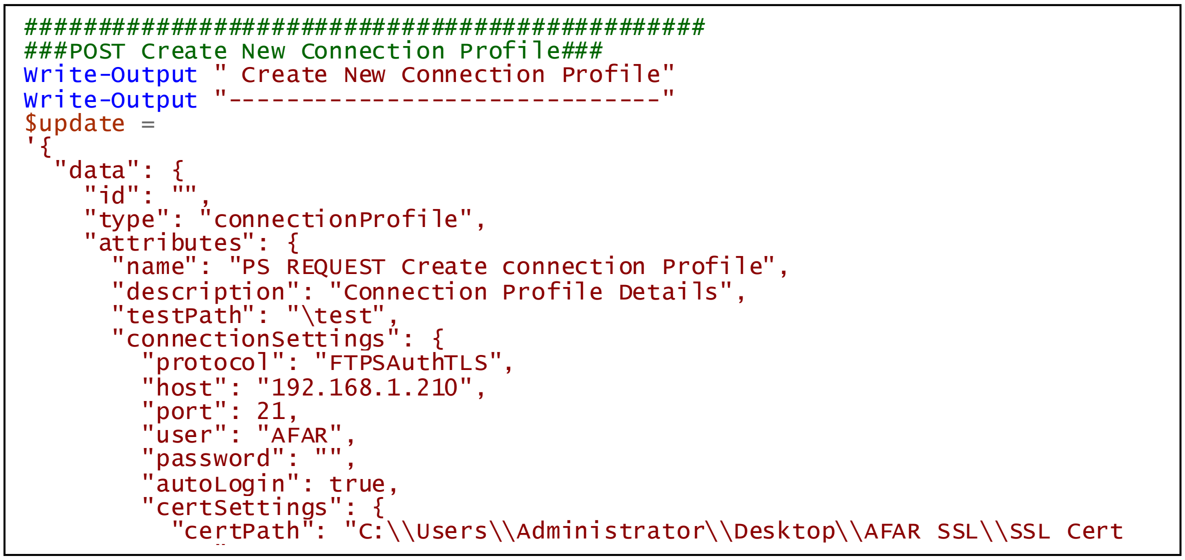
```
### Create a Connection Profile ###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles/ HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
 "data": {
  "type": "connectionProfile",
  "attributes": {
   "name": "POST REQUEST Create connection Profile",
   "description": "Connection Profile Details",
   "testPath": "\test",
   "connectionSettings": {
    "protocol": "FTPSAuthTLS",
    "host": "192.168.1.210",
    "port": 21,
    "user": "AFAR",
    "password": "",
    "autoLogin": true
```

POST Response Body Sample



```
HTTP/1.1 201 Created Date: Thu, 04 Aug 2022 15:51:07 GMT Location:
/admin/v2/sites/76882fc7-3f55-4040-ae5f-2e645c12647b/connection-
profiles/92e9f684-5b87-47d3-816b-19644e639ae2 Content-Type:
application/vnd.api+json ETag: "24708EE" Content-Length: 1290 Cache-Control:
no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-Options:
nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy: default-src
'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer {
```

POST PowerShell Sample



```
###############################################
###POST Create New Connection Profile###
Write-Output " Create New Connection Profile"
Write-Output "------------------------------"
$update =
'{
  "data": {
    "id": "",
    "type": "connectionProfile",
    "attributes": {
      "name": "PS REQUEST Create connection Profile",
      "description": "Connection Profile Details",
      "testPath": "\test",
      "connectionSettings": {
        "protocol": "FTPSAuthTLS",
        "host": "192.168.1.210",
        "port": 21,
        "user": "AFAR",
        "password": "",
        "autoLogin": true,
        "certSettings": {
          "certPath": "C:\\Users\\Administrator\\Desktop\\AFAR SSL\\SSL Cert
```

Endpoint: /sites/{siteId}/connection-profiles/{connectionProfileID}

Note: Available as of EFT 8.1.0

(GET) GET Details for specific connection profile, (PATCH) update details for a specific connection profile, (DELETE) remove a specific connection profile.

- GET
- PATCH
- DELETE

Method & URL

```
GET https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles
```

GET Body Sample

```
### GET Details for specific connection profile####
###You can enter the Connection profile ID below or pass the value from a previous
request###
##@ConnectionProfileID = 31334739-2f41-4561-9d27-22e17c9b9362
GET https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles/{{ConnectionProfileID}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Thu, 04 Aug 2022 16:11:35 GMT Location:
/admin/v2/sites/76882fc7-3f55-4040-ae5f-2e645c12647b/connection-
profiles/c9ea36ec-8a9e-440f-8c21-920eb3927732 Content-Type:
application/vnd.api+json ETag: "A50CA65F" Content-Length: 1289 Cache-Control:
no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-Options:
nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy: default-src
'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer
{ "data": { "id": "c9ea36ec-8a9e-440f-8c21-920eb3927732", "type":
"connectionProfile", "attributes": { "type": "Protocol", "name": "POST REQUEST
Create connection Profile", "description": "Connection Profile Details",
"connectionSettings": { "protocol": "FTPSAuthTLS", "user": "AFAR", "password":
"UpKUJq1sD4Yfb1Hz8ZFiRw==", "autoLogin": true, "testPath": "\test", "host":
"192.168.1.210", "port": 21, "SSLCertificate": { "certPath": "", "keyPath": "",
"passphrase": "UpKUJq1sD4Yfb1Hz8ZFiRw==" }, "trustedPubKey": "", "proxy": {
"use": false, "type": "FTP", "hostName": "", "username": "", "password":
"UpKUJq1sD4Yfb1Hz8ZFiRw==", "port": 21, "subType": 0, "separator": "@",
"customCommands": [] }, "socks": { "use": false, "type": "None", "hostName": "",
"port": 1080, "auth": false, "username": "", "password":
"UpKUJq1sD4Yfb1Hz8ZFiRw==" }, "advanced": { "numberOfConnectionRetries": 0,
"delayBetweenRetries": 30, "maxConcurrentThreads": 1, "connectionTimeout": 30,
```

GET PowerShell Sample

```
###Returns the deatails of the specified connection profile ###
Write-Output "Details of Connection Profile $profileId"
Write-Output "---------------------------------------------------------------
----"
#$ConnectionProfileID = "Enter site ID"
$CPlist = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/connection-
Profiles/$ConnectionProfileID" -Method 'GET' -Headers $authHeader
Write-Output $CPlist | ConvertTo-Json -Depth 25
```

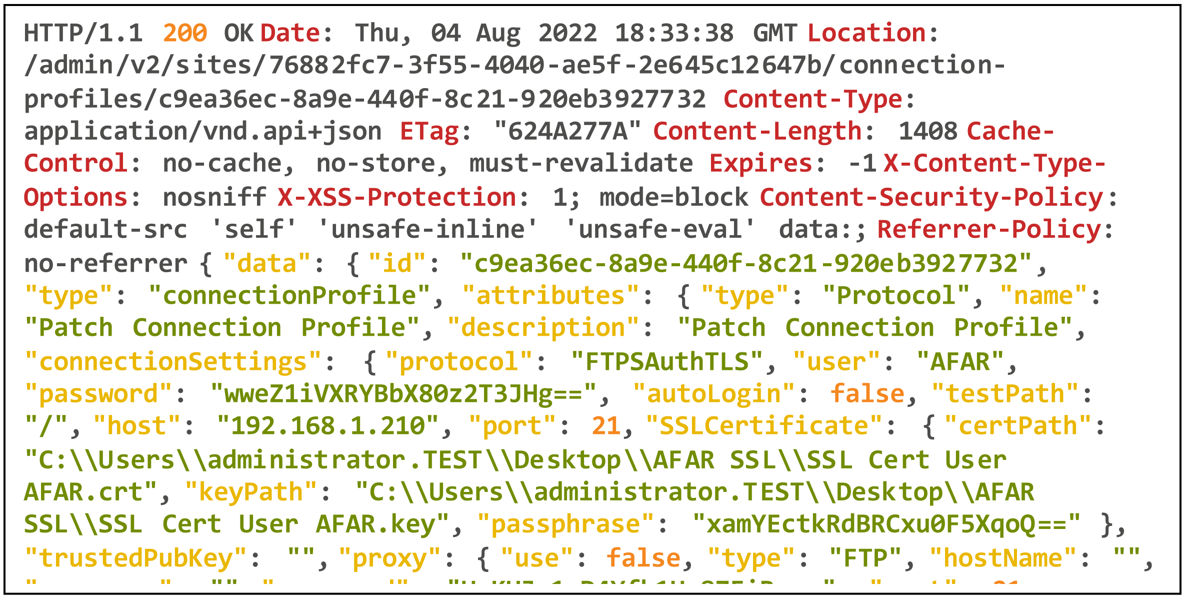
Method & URL

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/connection-
Profiles/{[ConnectionProfileID}}
```

PATCH Body Sample

```
### Update a Connection Profile ####
###You can enter the Connection profile ID below or pass the value from a previous
request###
##@ConnectionProfileID = 31334739-2f41-4561-9d27-22e17c9b9362
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles/{{ConnectionProfileID}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
 "data": {
    "type": "connectionProfile",
    "attributes":{
     "name": "Patch Connection Profile",
     "description": "Patch Connection Profile",
     "testPath": "/",
     "connectionSettings": {
       "protocol": "FTPSAuthTLS",
       "host": "192.168.1.210",
       "port": 21,
       "user": "AFAR",
       "password": "pass",
       "autoLogin": false
       }
     }
```

PATCH Response Body Sample



```
HTTP/1.1 200 OK Date: Thu, 04 Aug 2022 18:33:38 GMT Location:
/admin/v2/sites/76882fc7-3f55-4040-ae5f-2e645c12647b/connection-
profiles/c9ea36ec-8a9e-440f-8c21-920eb3927732 Content-Type:
application/vnd.api+json ETag: "624A277A" Content-Length: 1408 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "id": "c9ea36ec-8a9e-440f-8c21-920eb3927732",
"type": "connectionProfile", "attributes": { "type": "Protocol", "name":
"Patch Connection Profile", "description": "Patch Connection Profile",
"connectionSettings": { "protocol": "FTPSAuthTLS", "user": "AFAR",
"password": "wweZ1iVXRYBbX80z2T3JHg==", "autoLogin": false, "testPath":
"/", "host": "192.168.1.210", "port": 21, "SSLCertificate": { "certPath":
"C:\\Users\\administrator.TEST\\Desktop\\AFAR SSL\\SSL Cert User
AFAR.crt", "keyPath": "C:\\Users\\administrator.TEST\\Desktop\\AFAR
SSL\\SSL Cert User AFAR.key", "passphrase": "xamYEctkRdBRCxu0F5XqoQ==" },
"trustedPubKey": "", "proxy": { "use": false, "type": "FTP", "hostName": "",
```

PATCH PowerShell Sample

```
####Added 8/4/2022
### PATCH Connection Profile###
### User must obtain the SiteID and Connection Profile ID ###
Write-Output "PATCH Site Connection Profile"
Write-Output "------------------------------"
$update =
'{
  "data": {
      "type": "connectionProfile",
      "attributes": {
        "name": "PS_Patch Connection Profile",
        "description": "PS_Patch Connection Profile",
        "testPath": "/",
        "connectionSettings": {
          "protocol": "FTPSAuthTLS",
          "host": "192.168.1.210",
          "port": 21,
          "user": "AFAR",
          "password": "pass",
          "autoLogin": false
          }
        }
```

Method & URL

DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/connection-Profiles/{{ConnectionProfileID}}

DELETE Body Sample

```
### DELETE specific connection profile####
DELETE  https://{{host}}/admin/v2/sites/{{Siteid}}/connection-
Profiles/{{ConnectionProfileID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Thu, 04 Aug 2022 18:52:32 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
```

DELETE PowerShell Sample

```
####Added 8/4/2022 ###
###### DELETE Connection Profile######
Write-Output "### DELETE specific Connection Profile###"
Write-Output "----------------------------------------"
$profileName = "PowerShell Example"
$Delete_Profile = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/connection-Profiles/$ConnectionProfileID" -Method
'DELETE' -Headers $authHeader
```

Endpoint: /experimental/site/templates/{templateName}

Returns the contents of the specified site template

Note: Available as of EFT 8.1.0

- GET
- PATCH

Method & URL

GET https://{{host}}/admin/v2/sites/{{Siteid}}/templates/{templateName}

GET Body Sample

```
#GET Site Templates -- ##
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/templates/WorkspacesAdminAddedParticipantMsg.html HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Fri, 05 Aug 2022 13:01:21 GMT Content-Type:
application/vnd.api+json Content-Length: 370 Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer {
"data": "<html>\r\n<body>\r\n <p>Hello %USER.LOGIN%,</p>\r\n <p>The
following <b>Enhanced File Transfer</b> Workspace has been shared with
you: %FOLDER_NAME% (owner: %WS_OWNER_NAME%).</p>\r\n <p>Please use the
following URL to access this Workspace:</p>\r\n <a
href=\"%LINK%\">%LINK%</a><br/><br/>\r\n \r\n %WS_MESSAGE%<br/><br/>\r\n
<hr/>\r\n</body>\r\n</html>" }
```

GET PowerShell Sample

```
#####################################
#GET Site Template Details -- ##
Write-Output "GET Site Template Detail"
Write-Output "-------------------------------"
#$siteID = "Enter site ID"
#$TemplateName= "WorkspacesAdminAddedParticipantMsg.html"
$TemplateDetails = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/templates/WorkspacesAdminAddedParticipantMsg.html"
-Method 'GET' -Headers $authHeader
Write-Output $TemplateDetails | ConvertTo-Json -Depth 25
```

Method & URL

PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/templates/WorkspacesAdminAddedParticipantMsg.html

PATCH Body Sample

```
#Modify Site Template - Will append the site name to the file name should only be used for
this site ##
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/templates/WorkspacesAdminAddedParticipantMsg.h
tml HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": "<html>\r\n<body>\r\n  <p>Update Template Example
%USER.LOGIN%,</p>\r\n  <p>The following <b>Enhanced File Transfer</b> Workspace has
been shared with you:  %FOLDER_NAME% (owner:
%WS_OWNER_NAME%).</p>\r\n  <p>Please use the following URL to access this
Workspace:</p>\r\n  <a
href=\"%LINK%\">%LINK%</a><br/><br/>\r\n  \r\n  %WS_MESSAGE%<br/><br/>\r\n  <hr/>\r\
n</body>\r\n</html>"
}
```

PATCH Response Body Sample

```
HTTP/1.1 204 No Content Date: Fri, 05 Aug 2022 13:48:49 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

PATCH PowerShell Sample

```
####Added 8/4/2022
### Update Content of Template###
### Modify Site Template - Will append the site name to the file name should
only be used for this site  ###
Write-Output "PATCH Site Connection Profile"
Write-Output "------------------------------"
$update =
'{
  "data": "<html>\r\n<body>\r\n  <p>PS1. Update Template Example
%USER.LOGIN%,</p>\r\n  <p>The following <b>Enhanced File Transfer</b>
Workspace has been shared with you: %FOLDER_NAME% (owner:
%WS_OWNER_NAME%).</p>\r\n  <p>Please use the following URL to access this
Workspace:</p>\r\n  <a href=\"%LINK%\">%LINK%</a><br/><br/>\r\n  \r\n
%WS_MESSAGE%<br/><br/>\r\n  <hr/>\r\n</body>\r\n</html>"
}'
```

Endpoint:/sites/{siteID}/dmz

Note: Available as of EFT 8.1.0

(GET) Returns DMZ gateway settings from specified site and (PATCH) updates DMZ gateway settings

- GET
- PATCH

Method & URL

GET https://{{host}}/admin/v2/sites/{{Siteid}}/dmz

GET Body Sample

```
### GET SITE DMZ INFO ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/dmz HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":{"type":"DMZGateway","attributes":{"enabled":false,"hosts":[],"po
rt":44500,"securePNC":{"enabled":false,"publicKeyPath":""},"ftp":{"enable
d":false,"port":21},"ftpsExpl":{"enabled":false,"port":21},"ftpsImpl":{"e
nabled":false,"port":990},"sftp":{"enabled":false,"port":22},"http":{"ena
bled":false,"port":80},"https":{"enabled":false,"port":443},"internalDoma
in":{"HTTPDomain":"","emailDomainPatterns":"","allowADorLDAPAuth":true,"i
sExclusiveADorLDAP":false,"isRequired2FAforInternalUsers":false,"SSLCerti
ficate":{"certPath":"","keyPath":"","passphrase":"UpKUJq1sD4Yfb1Hz8ZFiRw=
="}},"pasv":{"enabled":false,"listenIP":"0.0.0.0","portMin":28000,"portMa
x":30000},"remoteAdmin":{"enabled":false,"port":1100},"acceleration":{"en
```

GET PowerShell Sample

```
$siteID = $siteList.data[0].id

# get DMZ info
Write-Output "DMZ INFO"
Write-Output "-----"
$dmzList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/dmz" -Method
'GET' -Headers $authHeader
Write-Output $dmzList | ConvertTo-Json
```

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/dmz
```

PATCH Body Sample

```
### PATCH (UPDATE) DMZ INFO ###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/dmz HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "DMZGateway",
    "attributes": {
      "hosts":
      [
        "10.91.160.102"
      ],
      "port":"44500",
      "enabled": true,
      "ftpEnabled": false,
      "ftpsEnabled": true,
      "ftpsExpEnabled": true,
      "sftpEnabled": false,
      "ftpPort": 21,
      "ftpsPort": 990,
      "httpsEnabled": true,
      "adminEnabled": true,
      "adminPort": 1100
    }
  }
}
###
```

PATCH Response Body Sample

```
{"data":{"type":"DMZGateway","attributes":{"enabled":true,"hosts":["10.91
.160.102"],"port":44500,"securePNC":{"enabled":false,"publicKeyPath":""},
"ftp":{"enabled":false,"port":21},"ftpsExpl":{"enabled":false,"port":21},
"ftpsImpl":{"enabled":false,"port":990},"sftp":{"enabled":false,"port":22
},"http":{"enabled":false,"port":80},"https":{"enabled":false,"port":443}
,"internalDomain":{"HTTPDomain":"","emailDomainPatterns":"","allowADorLDA
PAuth":true,"isExclusiveADorLDAP":false,"isRequired2FAforInternalUsers":f
alse,"SSLCertificate":{"certPath":"","keyPath":"","passphrase":"UpKUJq1sD
4Yfb1Hz8ZFiRw=="}},"pasv":{"enabled":false,"listenIP":"0.0.0.0","portMin"
:28000,"portMax":30000},"remoteAdmin":{"enabled":false,"port":1100},"acce
```

PATCH PowerShell Sample

```
##Patch SITE DMZ settings
Write-Output "Site DMZ PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "DMZGateway",
    "attributes": {
      "hosts":
      [
        "10.91.160.102"
      ],
      "port":"44500",
      "enabled": true,
      "ftpEnabled": false,
      "ftpsEnabled": true,
      "ftpsExpEnabled": true,
      "sftpEnabled": false,
      "ftpPort": 21,
      "ftpsPort": 990,
      "httpsEnabled": true,
      "adminEnabled": true,
      "adminPort": 1100
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchDMZReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/dmz" -
Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $patchDMZReturn | ConvertTo-Json
```

Security REST API

Endpoint: /lists/security/tls/algorithms

Returns a list of tls algorithms

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/lists/security/tls/algorithms

GET Body Sample

```
###
##Get Server TLS Security Settings
GET https://{{host}}/admin/v2/lists/security/tls/algorithms HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { "resultingCipherList": [ "ECDHE-ECDSA-AES256-
GCM-SHA384", "ECDHE-RSA-AES256-GCM-SHA384", "DHE-RSA-AES256-GCM-SHA384",
"ECDHE-ECDSA-CHACHA20-POLY1305", "ECDHE-RSA-CHACHA20-POLY1305", "DHE-RSA-
CHACHA20-POLY1305", "ECDHE-ECDSA-AES128-GCM-SHA256", "ECDHE-RSA-AES128-
GCM-SHA256", "DHE-RSA-AES128-GCM-SHA256", "ECDHE-ECDSA-AES256-SHA384",
```

GET PowerShell Sample

```
###
#GET TLS ALOGRITHMS
Write-Output "Get EFT TLS ALGORITHMS"
$serverNodeID = "DTY2019WinServe"

Write-Output "Server Node: $serverNodeID"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/lists/security/tls/algorithms" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Endpoint: /lists/security/ssh/algorithms

Returns a list of ssh algorithms

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/lists/security/ssh/algorithms

GET Body Sample

```
###
##Get Server SSH Security Settings
GET https://{{host}}/admin/v2/lists/security/ssh/algorithms HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { "algorithms": { "ciphers": [ "aes256-
gcm@openssh.com", "aes256-ctr", "aes256-cbc", "rijndael-
cbc@lysator.liu.se", "twofish256-cbc", "twofish-cbc", "aes192-ctr",
"aes192-cbc", "aes128-gcm@openssh.com", "aes128-ctr", "aes128-cbc",
"twofish128-cbc",
```

GET PowerShell Sample

```
###
#GET SSH ALOGRITHMS
Write-Output "Get EFT SSH ALGORITHMS"
$serverNodeID = "DTY2019WinServe"

Write-Output "Server Node: $serverNodeID"
Write-Output "-----"
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/lists/security/ssh/algorithms" -Method 'GET' -Headers $authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Endpoint: /lists/security/tls/ciphers/{cipherString}

Returns a list of resulting ciphers

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/lists/security/tls/ciphers/{cipherString}

GET Body Sample

```
###
##Get Server TLS Ciphers Settings
GET https://{{host}}/admin/v2/lists/security/tls/ciphers/"aes-ctr"
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { "resultingCipherList": [ "ECDHE-ECDSA-AES256-
GCM-SHA384", "ECDHE-RSA-AES256-GCM-SHA384", "DHE-RSA-AES256-GCM-SHA384",
"ECDHE-ECDSA-CHACHA20-POLY1305", "ECDHE-RSA-CHACHA20-POLY1305", "DHE-RSA-
CHACHA20-POLY1305", "ECDHE-ECDSA-AES128-GCM-SHA256", "ECDHE-RSA-AES128-
GCM-SHA256", "DHE-RSA-AES128-GCM-SHA256", "ECDHE-ECDSA-AES256-SHA384",
```

GET PowerShell Sample

```
###
#GET TLS CIPHERS
Write-Output "Get EFT TLS CIPHERS"
$serverNodeID = "DTY2019WinServe"
$cipherId = "aes-ctr"

Write-Output "Server Node: $serverNodeID"
Write-Output "-----""
$serverAdminDetails = Invoke-RestMethod -Uri
"$baseURL/v2/lists/security/tls/ciphers/$cipherID -Method 'GET' -Headers
$authHeader
Write-Output $serverAdminDetails | ConvertTo-Json -Depth 10
```

Login/Logout REST API

Endpoint: /logout

Logs out of current session.

- POST

Method & URL

POST https://[server URL]:[port]/admin/v2/logout

POST Body Sample

```
POST https://{{host}}/admin/v2/logout HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

POST Response Body Sample

```
HTTP/1.1 200 OK Date: Wed, 10 Aug 2022 20:10:28 GMT Content-Length: 0
Set-Cookie: websessionid=xxxxxx; expires=Tuesday, 10-Aug-21 20:10:28 GMT;
path=/; secure; HttpOnly; SameSite=Strict Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval' data:: Referrer-Policv: no-referrer
```

POST PowerShell Sample

```
# POST - logout
$auth = Invoke-RestMethod -Uri "$baseURL/v2/logout" -Method 'POST' -Body
$authBody -Headers $authHeader
#
```

Keep-Alive REST API

Endpoint: /keep-alive

Note: Available as of EFT 8.1.0

Extends current session.

- HEAD

Method & URL

HEAD https://[server URL]:[port]/admin/v2/keep-alive

HEAD Body Sample

```
##Get Server keep alive
HEAD https://{{host}}/admin/v2/keep-alive HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

HEAD Response Body Sample

```
HTTP/1.1 200 OK Date: Mon, 21 Feb 2022 22:17:46 GMT Content-Length: 0
Cache-Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-
Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-
Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-
Policy: no-referrer
```

HEAD PowerShell Sample

```
# keep-alive
Write-Output "Keep Alive"
Write-Output "----"
$keepAlive = Invoke-RestMethod -Uri "$baseURL/v2/keep-alive" -Method 'HEAD' -
Headers $authHeader
```

Users REST API

Endpoint: /sites/{siteId}/users

Note: Available as of EFT 8.1.0

Returns a list of users for specific site, create user(s) for a specific site.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/users

GET Body Sample

```
### GET SITE USERS ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": [ { "attributes": { "accountEnabled": "inherit",
"agreementToTermsOfService": "unknown", "as2": { "inbound": { "enabled":
"inherit" }, "outbound": { "enabled": "inherit" } }, "changePassword": {
"enabled": "inherit", "value": {
"changeAdminProvidedPasswordUponFirstUse": "inherit",
```

GET PowerShell Sample

```
# get sites
Write-Output "Site"
Write-Output "----"
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites" -Method 'GET' -Headers
$authHeader
Write-Output $siteList | ConvertTo-Json
foreach ($site in $siteList.data)
{
    Write-Output ("Site  $($site.id): $($site.attributes.name)")
}

$siteID = $siteList.data[0].id

# get users
Write-Output "Users"
Write-Output "-----"
$userList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/users" -Method
'GET' -Headers $authHeader
Write-Output $userList | ConvertTo-Json
foreach ($user in $userList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

POST https://[server URL]:[port]/admin/v2/ sites/{siteId}/users

POST Body Sample

```
### ADD SITE USER ###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "user",
    "attributes": {
      "loginName": "TestUser1",
      "password": {
        "type": "Disabled",
        "value": "Alaska12#"
      },
      "accountEnabled": "inherit",
      "expiration": {
        "enabled": false,
        "value": {
          "date": {}
        }
      },
      "personal": {
        "name": "Test User",
        "description": null,
        "email": null,
```

POST Response Body Sample

```
"data": { "attributes": { "accountEnabled": "inherit",
"agreementToTermsOfService": "unknown", "as2": { "inbound": { "enabled":
"inherit" }, "outbound": { "enabled": "inherit" } }, "changePassword": {
"enabled": "inherit", "value": {
"changeAdminProvidedPasswordUponFirstUse": "inherit",
"mustChangePassword": false, "passwordExpiration": { "enabled": true,
"value": { "maxAgeDays": 90, "emailUpon": true, "remindPrior": {
"enabled": false, "value": { "daysBefore": 5, "emailUser": true } } } } }
}, "connectionTimeout": { "enabled": "inherit", "value": { "maxSec": 0 }
},
```

POST PowerShell Sample

```
Write-Output "Add User POST"
Write-Output "-----"
$update =
'{
  "data": {
    "type": "user",
    "attributes": {
      "loginName": "TestUser2",
      "password": {
        "type": "Disabled",
        "value": "Alaska12#"
      },
      "accountEnabled": "inherit",
      "expiration": {
        "enabled": false,
        "value": {
          "date": {}
        }
      },
      "personal": {
        "name": "Test User",
        "description": null,
        "email": null,
        "DUNS": null,
        "phone": null,
        "mobile": null,
        "pager": null,
        "fax": null,
        "company": null,
        "partnerId": null,
        "custom1": null,
        "custom2": null,
        "custom3": null,
        "comments": null
      },
      "isEuDataSubject": "unknown",
      "consentToPrivacyPolicy": "unknown",
      "agreementToTermsOfService": "unknown",
      "homeFolder": {
        "enabled": "yes",
        "value": {
          "path": null
        }
      },
      "hasHomeFolderAsRoot": "inherit",
      "externalAuthentication": "inherit",
      "ftp": {
        "enabled": "inherit",
        "value": {
          "enableFxp": "inherit",
          "enableNoop": "inherit",
          "enableComb": "inherit",
          "enableXcrc": "inherit",
          "enableZlib": "inherit",
          "banner": {
            "message": "",
            "usage": "useDefault"
          }
        }
      },
      "ftps": {
```

Endpoint: /sites/{siteid}/users/{userid}

Returns, updates, and deletes user account details.

- GET
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}

GET Body Sample

```
### GET SITE SPECIFIC USER ###
# @name usersList
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###

@Usersid = {{usersList.response.body.$.data[0].id}}
###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": { "attributes": { "accountEnabled": "inherit",
"agreementToTermsOfService": "unknown", "as2": { "inbound": { "enabled":
"inherit" }, "outbound": { "enabled": "inherit" } }, "changePassword": {
"enabled": "inherit", "value": {
"changeAdminProvidedPasswordUponFirstUse": "inherit",
"mustChangePassword": false, "passwordExpiration": { "enabled": true,
"value": { "maxAgeDays": 90, "emailUpon": true, "remindPrior": {
"enabled": false, "value": { "daysBefore": 5, "emailUser": true
```

GET PowerShell Sample

```
# get a specific users information
Write-Output "Users"
Write-Output "-----"
$userList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/users" -Method
'GET' -Headers $authHeader
Write-Output $userList | ConvertTo-Json
foreach ($user in $userList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}

$userid = $userList.data[0].id

# get user’s data
Write-Output "Users"
Write-Output "-----"
$userList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/users/$userid" -
Method 'GET' -Headers $authHeader
Write-Output $userList | ConvertTo-Json
foreach ($user in $userList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/

PATCH Body Sample

```
@Usersid = {{usersList.response.body.$.data[0].id}}
###
### UPDATE A USERS PERSONAL INFORMATION ###
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}}/
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "user",
    "attributes": {
      "personal": {
        "name": "MR JONES",
        "description": “MANAGER",
        "email": "user@example.com",
        "duns": "",
        "phone": "",
        "mobile": "2107221301",
        "pager": "",
        "fax": "",
        "company": "",
        "partnerId": "",
        "custom1": "1234",
        "custom2": "5678",
        "custom3": "abcd",
        "comments": "Does this work?"
      },
      "isEuDataSubject": "yes",
      "consentToPrivacyPolicy": "grantedExplicit",
      "agreementToTermsOfService": "agreedExplicit"
     }
  }
```

PATCH Response Body Sample

```
{ "data": { "attributes": { "accountEnabled": "inherit",
"agreementToTermsOfService": "agreedExplicit", "as2": { "inbound": {
"enabled": "inherit" }, "outbound": { "enabled": "inherit" } },
"changePassword": { "enabled": "inherit", "value": {
"changeAdminProvidedPasswordUponFirstUse": "inherit",
"mustChangePassword": false, "passwordExpiration": { "enabled": true,
"value": { "maxAgeDays": 90, "emailUpon": true, "remindPrior": {
"enabled": false, "value": { "daysBefore": 5, "emailUser": true } } } } }
```

PATCH PowerShell Sample

```
##UPDATE USERS
Write-Output "Update user data"
Write-Output "-----"
$update =
'{
    "data": {
       "type": "user",
       "attributes": {
       "accountEnabled": "no"

       }
      }

}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$PatchReturn = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/users/$userid" -Method 'PATCH' -Headers $authHeader
-Body $update
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/

DELETE Body Sample

```
### REMOVE THE USER ### (userOnly, userAndHomeFolder)
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}}
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "parameters": {
        "scope": "userAndHomeFolder"
    }
}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Fri, 04 Feb 2022 21:00:36 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
```

DELETE PowerShell Sample

```
$update =
'{
    "parameters": {
        "scope": "userAndHomeFolder"
    }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
# delete
Write-Output "Delete User"
Write-Output "-----"
$deleteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/users/$userid"
-Method 'DELETE' -Headers $authHeader -Body $update
Write-Output $deleteList | ConvertTo-Json
foreach ($user in $deletList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteid}/users/{userid}/as2inbound

Returns and updates user inbound AS2 parameters

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/as2inbound

GET Body Sample

```
# @name usersList
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###

@Usersid = {{usersList.response.body.$.data[0].id}}
###
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}}/as2inbound
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { }, "relationships": {
"transactionFailedCommand": { "data": null },
"transactionSuccessCommand": { "data": null } }, "type": "AS2Inbound" } }
```

GET PowerShell Sample

```
$userid = $userList.data[0].id

# get users
Write-Output "Specific User AS2inbound information"
Write-Output "-----"
$userList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/users/$userid/as2inbound" -Method 'GET' -Headers
$authHeader
Write-Output $userList | ConvertTo-Json
foreach ($user in $userList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/as2inbound

PATCH Body Sample

```
###
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}}/as2inbound
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "AS2Inbound",
    "attributes": {
      "myId": "AS2_1",
      "partnerId": "AS2_2"
    }
```

PATCH Response Body Sample

```
{
  "data": {
    "attributes": {
      "allowedEncryptionAlgorithms": [
      ],
      "allowedSignatureAlgorithms": [
      ],
      "authenticationType": "usernamePassword",
      "contentCollision": "reject",
      "doNotSend100Continue": false,
      "folderToMoveDataTo": "",
      "mdnSendAttemptDelaySeconds": 30,
```

PATCH PowerShell Sample

```
$update =
'{
  "data": {
    "type": "AS2Inbound",
    "attributes": {
      "myId": "AS2_1",
      "partnerId": "AS2_4",
      "partnerCertificateFilePath":
"C:\\Users\\DYelacic1951\\Desktop\\certs\\ssl.crt"
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
# update / patch
Write-Output "Update User specific as2inbound information"
Write-Output "-----"
$as2InboundList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/users/$userid/as2inbound" -Method 'PATCH' -Headers
$authHeader -Body $update
Write-Output $as2InboundList | ConvertTo-Json
foreach ($user in $as2InboundList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteid}/users/{userid}/as2outbound

Returns and updates user outbound AS2 parameters

- GET
- PATCH

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/as2outbound

GET Body Sample

```
# @name usersList
GET https://{{host}}/admin/v2/sites/{{Siteid}}/users HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###

@Usersid = {{usersList.response.body.$.data[0].id}}
###
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}}/as2outbound
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{ "data": { "attributes": { }, "relationships": { }, "type":
"AS2Outbound" } }
```

GET PowerShell Sample

```
Write-Output "Users specific as2outbound information"
Write-Output "-----"
$as2outList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/users/$userid/as2outbound" -Method 'GET' -Headers
$authHeader
Write-Output $as2outList | ConvertTo-Json
foreach ($user in $as2outList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/as2outbound

PATCH Body Sample

```
###
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{Usersid}}/as2outbound
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "AS2outbound",
    "attributes": {
      "myId": "AS2_1",
      "partnerId": "AS2_4",
    "hotFolderPath": "C:\\Users\\DYelacic1951\\Desktop\\HOTBOX\\"
    }
  }
```

PATCH Response Body Sample

```
{ "data": { "attributes": { }, "relationships": { }, "type":
"AS2Outbound" } }
```

PATCH PowerShell Sample

```
$update =
'{
  "data": {
    "type": "AS2outbound",
    "attributes": {
      "myId": "AS2_1",
      "partnerId": "AS2_4",
         "hotFolderPath": "C:\\Users\\DYelacic1951\\Desktop\\HOTBOX\\"
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
# update / patch
Write-Output "Update User specific as2outbound information"
Write-Output "-----"
$as2outboundList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/users/$userid/as2outbound" -Method 'PATCH' -Headers
$authHeader -Body $update
Write-Output $as2outboundList | ConvertTo-Json
foreach ($user in $as2outboundList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteid}/users/{userid}/counters

Returns user account statistics (As of EFT 8.1 the response will include this data pair "timePasswordChanged")

- GET

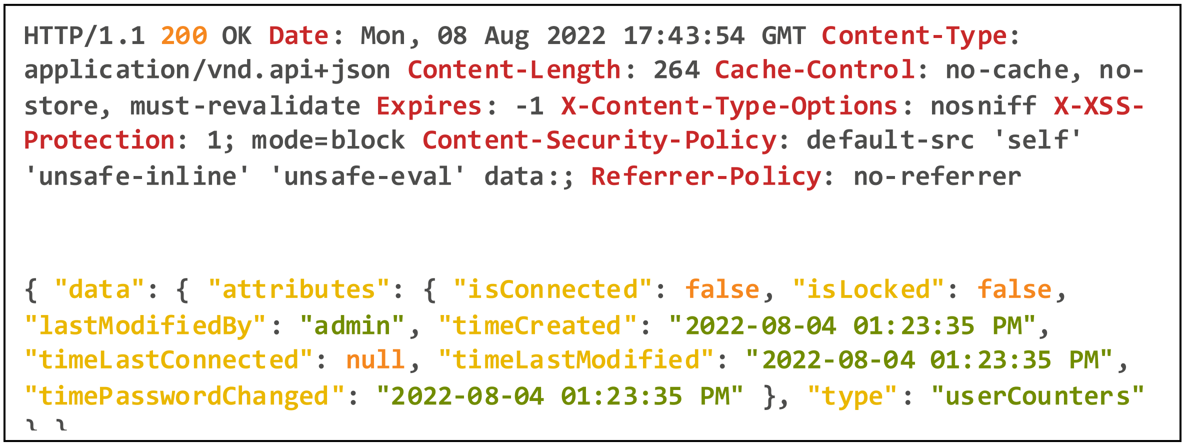
Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/users/{userid}/counters

GET Body Sample

```
#### GET USER DETAILS COUNTERS ####
## Must obtain User_ID##

GET https://{{host}}/admin/v2/sites/{{Siteid}}/users/{{UserID}}/counters HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample



```
HTTP/1.1 200 OK Date: Mon, 08 Aug 2022 17:43:54 GMT Content-Type:
application/vnd.api+json Content-Length: 264 Cache-Control: no-cache, no-
store, must-revalidate Expires: -1 X-Content-Type-Options: nosniff X-XSS-
Protection: 1; mode=block Content-Security-Policy: default-src 'self'
'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy: no-referrer


{ "data": { "attributes": { "isConnected": false, "isLocked": false,
"lastModifiedBy": "admin", "timeCreated": "2022-08-04 01:23:35 PM",
"timeLastConnected": null, "timeLastModified": "2022-08-04 01:23:35 PM",
"timePasswordChanged": "2022-08-04 01:23:35 PM" }, "type": "userCounters"
} }
```

GET PowerShell Sample

```
$userid = $userList.data[0].id

# get users counters
Write-Output "Users counters"
Write-Output "-----"
$userList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/users/$userid/counters" -Method 'GET' -Headers
$authHeader
Write-Output $userList | ConvertTo-Json
foreach ($user in $userList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Filesystem REST API

Endpoint: /sites/{siteId}/filesystem/folders

Returns folder (updated on EFT 8.1, it now includes the Upload Quota Settings)

- GET

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders

GET Body Sample

```
@Siteid = {{siteList.response.body.$.data[0].id}}
###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Mon, 08 Aug 2022 16:59:16 GMT Content-Type:
application/vnd.api+json ETag: "B7E6B8C3" Content-Length: 529 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer

{ "data": { "attributes": { "name": "", "subFolders": [ "anonymous", "Bin",
"Incoming", "Pub", "Usr" ], "target": { "type": "winFolder", "winFolder": {
"path": "C:\\InetPub\\EFTRoot\\MySite\\" } }, "uploadQuota": null }, "id":
"%2F", "links": { "self": "/admin/v2/sites/e82de253-b8ec-4b70-a72d-
d017c3317f01/filesystem/folders/%252F" }, "relationships": {
"inheritedPermissions": { "meta": { "name": "inherited permissions" } },
"permissions": { "links": { "related": "/admin/v2/sites/e82de253-b8ec-
```

GET PowerShell Sample

```
$userid = $userList.data[0].id

# get users counters
Write-Output "filesystem folders"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders" -Method 'GET' -Headers
$authHeader
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteId}/filesystem/folders/{folderid}

Returns folder settings, updates folder settings, creates folders, and deletes folders (updated on EFT 8.1, it now includes the Upload Quota Settings)

- GET
- POST
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}

GET Body Sample

```
##### GET the Folder structure and settings for a specifc User   #########
@User_Folder_ID = AFAR
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2F{{User_Folder_ID}}%2F
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Mon, 08 Aug 2022 17:04:14 GMT Content-Type:
application/vnd.api+json ETag: "95FFEAF1" Content-Length: 470 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer
{ "data": { "attributes": { "name": "AFAR", "subFolders": [], "target":
null, "uploadQuota": null }, "id": "%2Fusr%2FAFAR%2F", "links": { "self":
"/admin/v2/sites/e82de253-b8ec-4b70-a72d-
d017c3317f01/filesystem/folders/%252Fusr%252FAFAR%252F" },
"relationships": { "inheritedPermissions": { "meta": { "name": "inherited
permissions" } }, "permissions": { "links": { "related":
"/admin/v2/sites/e82de253-b8ec-4b70-a72d-
d017c3317f01/filesystem/folders/%252Fusr%252FAFAR%252F/permissions" } } }
```

GET PowerShell Sample

```
# get filesystem folders
Write-Output "filesystem folders"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2Fusr%2F" -Method 'GET' -
Headers $authHeader
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}

POST Body Sample

```
###
POST
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2fusr%2fdy
elacic%2f/folders HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
    "data": {
        "type": "folder",
        "id": "%2Fbiohazard%2F",
        "attributes": {
            "name": "biohazard",
            "target": "c:/temp"
        }
    }
}
```

POST Body Sample - Example II – including Upload Quotas

```
##### POST  REQUEST TO CREATE A FOLDER WITH UPLOAD QUOTA ENABLED #########
@User_Folder = afar
POST
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2F{{User_Folder}}%2F/fo
lders HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
 "data": {
   "type": "folder",
   "attributes": {
     "name": "Folder4",
     "subFolder": "Folder4",
   "uploadQuota": {
       "emailsToReportToOnExceed": [
         "gsqaaf1@gmail.com",
         "gsqaaf2@gmail.com"
       ],
       "enabled": true,
       "limitMBytes": 150,
       "usedMBytes": 0
     }
   }
 }
```

POST Response Body Sample

```
{ "data": { "attributes": { "name": "biohazard", "subFolders": [],
"target": "c:/temp" }, "id": "%2Fusr%2Fdyelacic%2Fbiohazard%2F", "links":
{ "self": "/Admin/v2/sites/f1e58e70-926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252Fbiohazard%252F"
}, "relationships": { "inheritedPermissions": { "meta": { "name":
"inherited permissions" } }, "permissions": { "links": { "related":
"/Admin/v2/sites/f1e58e70-926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252Fbiohazard%252F/
```

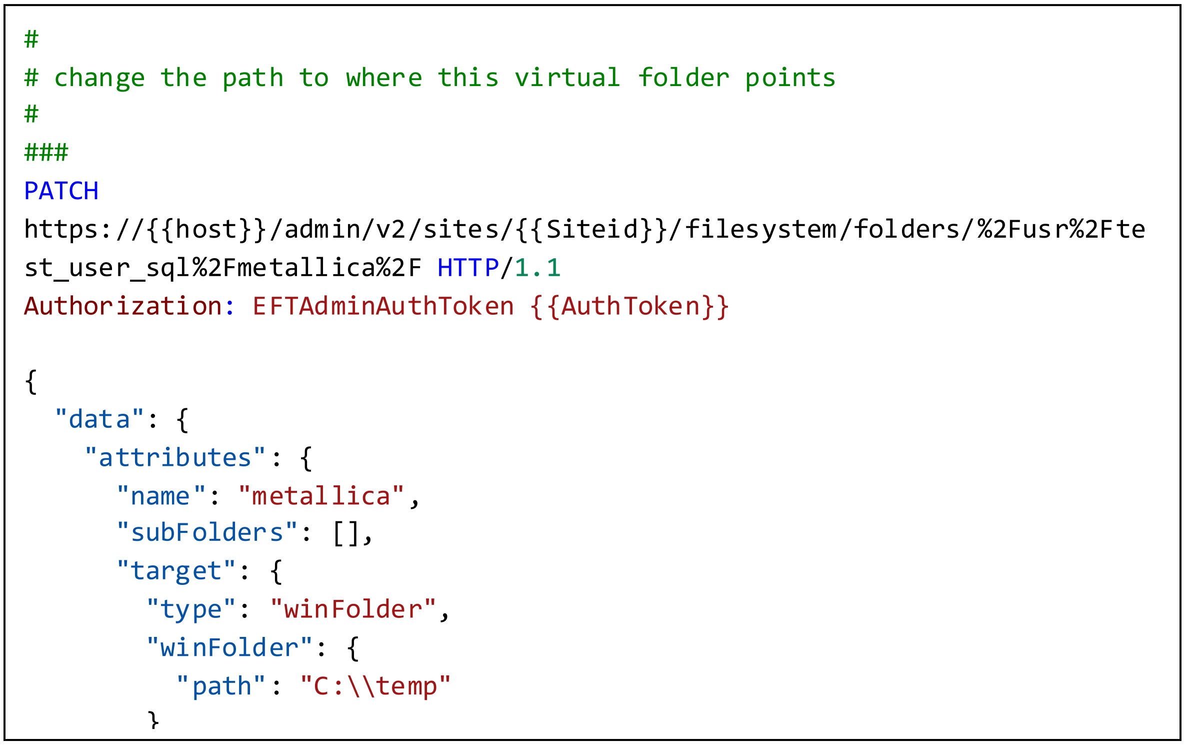
POST PowerShell Sample

```
$update =
'{
    "data": {
        "type": "folder",
        "id": "%2Fmetallica2%2F",
        "attributes": {
            "name": "metallica2",
            "target": "c:/util"
        }
    }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 20
# update filesystem folders
Write-Output "filesystem folders"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2fusr%2fFranklin%2f" -Method
'POST' -Headers $authHeader -Body $update
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}

PATCH Body Sample



```
#
# change the path to where this virtual folder points
#
###
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fte
st_user_sql%2Fmetallica%2F HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "attributes": {
      "name": "metallica",
      "subFolders": [],
      "target": {
        "type": "winFolder",
        "winFolder": {
          "path": "C:\\temp"
        }
```

PATCH Response Body Sample

```
HTTP/1.1 200 OK Date: Thu, 18 Aug 2022 13:31:59 GMT Content-Type:
application/vnd.api+json ETag: "B0A1B271" Content-Length: 590 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "attributes": { "name": "metallica",
"subFolders": [], "target": { "type": "winFolder", "winFolder": { "path":
"C:\\temp" } }, "uploadQuota": null }, "id":
"%2Fusr%2Ftest_user_sql%2Fmetallica%2F", "links": { "self":
```

PATCH PowerShell Sample

```
#
# renaming the virtual folder name
#
$update =
'{

  "data": {
    "attributes": {
      "name": "metallica",
      "subFolders": [],
      "target": {
        "type": "winFolder",
        "winFolder": {
          "path": "C:\\temp"
        }
      },
      "uploadQuota": null
    },
    "id": "%2Fusr%2Ftest_user_sql%2Fmetallica%2F",
    "type": "folder"
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 20
# update filesystem folders
Write-Output "patch filesystem folders"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2fusr%2fdyelacic%2fmetallica%2f
" -Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}

DELETE Body Sample

```
###
DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2FFranklin%2F HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Thu, 10 Feb 2022 14:49:46 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
Write-Output "delete filesystem folders"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2Fusr%2Fdyelacic%2F" -Method
'DELETE' -Headers $authHeader
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteId}/filesystem/folders/{folderid}/folders

Creates physical or virtual sub-folder

- POST

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}/folders

POST Body Sample

```
#
# creating a physical sub folder below the users home folder
#
###
POST
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fdy
elacic%2F/folders HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "folder",
    "attributes": {
      "name": "tuneroom",
      "subFolder": "tuneroom"
    }
  }
}
###
```

POST Response Body Sample

```
{ "data": { "attributes": { "name": "tuneroom", "subFolders": [],
"target": null }, "id": "%2Fusr%2Fdyelacic%2Ftuneroom%2F", "links": {
"self": "/Admin/v2/sites/f1e58e70-926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252Ftuneroom%252F"
}, "relationships": { "inheritedPermissions": { "meta": { "name":
"inherited permissions" } }, "permissions": { "links": { "related":
"/Admin/v2/sites/f1e58e70-926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252Ftuneroom%252F/p
ermissions" } } }, "type": "folder" } }
```

POST PowerShell Sample

```
#
# create physical sub folder below users home folder
#
$update =
'{
    "data": {
        "type": "folder",
        "attributes": {
            "name": "sunroom",
            "subFolder": "sunroom"
        }
    }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 20
# create filesystem folders
Write-Output "create sub folder"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2fusr%2fdyelacic%2f/folders" -
Method 'POST' -Headers $authHeader -Body $update
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteId}/filesystem/folders/{folderid}/permissions

Returns a list of permissions for a specific folder, adding existing permissions to a specific folder.

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}/permissions

GET Body Sample

```
# get folder permissions
#
###
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fdy
elacic%2F/permissions HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": [ { "attributes": { "canAppendFile": true,
"canCreateSubfolder": true, "canDeleteFile": true, "canDeleteSubfolder":
true, "canDownloadFile": true, "canList": true,
"canRenameFileOrSubfolder": true, "canSeeHiddenFileOrSubfolder": true,
"canSeeInParentList": true, "canSeeReadOnlyFileOrSubfolder": true,
"canUploadFile": true }, "links": { "self": "/Admin/v2/sites/f1e58e70-
926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252F/permissions/us
er-04dcfd20-fc73-41d0-a0d8-62d1706e6281" }, "relationships": { "trustee":
{ "data": { "id": "04dcfd20-fc73-41d0-a0d8-62d1706e6281", "meta": {
"name": "dyelacic" }, "type": "user" }, "links": { "related":
"/Admin/v2/sites/f1e58e70-926f-4786-877b-cf62abfdf0db/users/04dcfd20-
```

GET PowerShell Sample

```
#
# get filesystem folders permissions
Write-Output "filesystem folders"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2Fusr%2Fdyelacic%2f/permissions
" -Method 'GET' -Headers $authHeader
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}/permissions

POST Body Sample

```
###
#Add / Link folder to an existing user permissions group.

POST
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fte
st_user_sql%2Fspam%2F/permissions HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data":
    {
      "type": "folderPermissions",
      "attributes": {
        "canAppendFile": true,
        "canCreateSubfolder": true,
        "canDeleteFile": true,
        "canDeleteSubfolder": true,
        "canDownloadFile": true,
        "canList": true,
        "canRenameFileOrSubfolder": true,
        "canSeeHiddenFileOrSubfolder": true,
        "canSeeInParentList": true,
        "canSeeReadOnlyFileOrSubfolder": true,
        "canUploadFile": false
      },
      "relationships": {
        "trustee": {
          "data": {
            "type": "userGroup",
            "id": " b8448b77-01d8-47d5-8ee7-9f407c95e39f",
            "meta": {
              "name": "Guests"
            }
          }
        }
      }
    }
 }
###
```

POST Response Body Sample

```
HTTP/1.1 201 Created Date: Thu, 18 Aug 2022 19:32:43 GMT Location:
/admin/v2/sites/7c2d15e3-e151-4029-8aa2-
d4475296ba1c/filesystem/folders/%252Fusr%252Ftest_user_sql%252Fspam%252F/
permissions/group-b8448b77-01d8-47d5-8ee7-9f407c95e39f Content-Type:
application/vnd.api+json ETag: "C5020903" Content-Length: 774 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "attributes": { "canAppendFile": true,
"canCreateSubfolder": true, "canDeleteFile": true, "canDeleteSubfolder":
true, "canDownloadFile": true, "canList": true,
"canRenameFileOrSubfolder": true, "canSeeHiddenFileOrSubfolder": true,
"canSeeInParentList": true, "canSeeReadOnlyFileOrSubfolder": true,
"canUploadFile": false }, "links": { "self": "/admin/v2/sites/7c2d15e3-
e151-4029-8aa2-
d4475296ba1c/filesystem/folders/%252Fusr%252Ftest_user_sql%252Fspam%252F/
permissions/group-b8448b77-01d8-47d5-8ee7-9f407c95e39f" },
"relationships": { "trustee": { "data": { "type": "userGroup", "id":
"b8448b77-01d8-47d5-8ee7-9f407c95e39f", "meta": { "name": "Guests" } },
"links": { "related": "/admin/v2/sites/7c2d15e3-e151-4029-8aa2-
```

POST PowerShell Sample

```
#
# Add an existing user permissions group to a folder
$update =
'{

  "data":
    {
      "type": "folderPermissions",
      "attributes": {
        "canAppendFile": true,
        "canCreateSubfolder": true,
        "canDeleteFile": true,
        "canDeleteSubfolder": true,
        "canDownloadFile": true,
        "canList": true,
        "canRenameFileOrSubfolder": true,
        "canSeeHiddenFileOrSubfolder": true,
        "canSeeInParentList": true,
        "canSeeReadOnlyFileOrSubfolder": true,
        "canUploadFile": false
      },
      "relationships": {
        "trustee": {
          "data": {
            "type": "userGroup",
            "id": "b8448b77-01d8-47d5-8ee7-9f407c95e39f",
            "meta": {
              "name": "Guests"
            }
          }
        }
      }
    }
 }’
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 30
# add sub folder permissions
Write-Output "add sub folder permissions"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2fusr%2fdyelacic%2ftuneroom%2f/
permissions" -Method 'POST' -Headers $authHeader -Body $update
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Endpoint: /sites/{siteId}/filesystem/folders/{folderid}/permissions/{folderpermid}

Returns, updates, and deletes permissions on folder

- GET

- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}/permissions/{folderpermid}

GET Body Sample

```
# get the permissions for a specific folder
#
###
GET
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fdy
elacic%2Fsunroom%2F/permissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{ "data": { "attributes": { "canAppendFile": true, "canCreateSubfolder":
true, "canDeleteFile": true, "canDeleteSubfolder": true,
"canDownloadFile": false, "canList": true, "canRenameFileOrSubfolder":
true, "canSeeHiddenFileOrSubfolder": true, "canSeeInParentList": true,
"canSeeReadOnlyFileOrSubfolder": true, "canUploadFile": true }, "links":
{ "self": "/Admin/v2/sites/f1e58e70-926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252Fsunroom%252F/pe
rmissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281" }, "relationships":
{ "trustee": { "data": { "id": "04dcfd20-fc73-41d0-a0d8-62d1706e6281",
"meta": { "name": "dyelacic" }, "type": "user" }, "links": { "related":
"/Admin/v2/sites/f1e58e70-926f-4786-877b-cf62abfdf0db/users/04dcfd20-
fc73-41d0-a0d8-62d1706e6281" } } }, "type": "folderPermissions" } }
```

GET PowerShell Sample

```
# get the folders permission
# get filesystem folders permissions
Write-Output "filesystem folders permissions"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2Fusr%2Fdyelacic%2fsunroom%2f/p
ermissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281" -Method 'GET' -Headers
$authHeader
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}/permissions/{folderpermid}

PATCH Body Sample

```
# change / update folder permissions
###
PATCH
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fdy
elacic%2Fsunroom%2F/permissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
    "type": "folderPermissions",
    "attributes": {
      "canUploadFile": false,
      "canDownloadFile": true,
      "canDeleteFile": false,
      "canRenameFileOrSubfolder": true,
      "canAppendFile": true,
      "canDeleteSubfolder": false,
      "canCreateSubfolder": true,
      "canSeeHiddenFileOrSubfolder": true,
      "canSeeReadOnlyFileOrSubfolder": true,
      "canSeeInParentList": true,
      "canList": true
    }
  }
```

PATCH Response Body Sample

```
{ "data": { "attributes": { "canAppendFile": true, "canCreateSubfolder":
true, "canDeleteFile": false, "canDeleteSubfolder": false,
"canDownloadFile": true, "canList": true, "canRenameFileOrSubfolder":
true, "canSeeHiddenFileOrSubfolder": true, "canSeeInParentList": true,
"canSeeReadOnlyFileOrSubfolder": true, "canUploadFile": false }, "links":
{ "self": "/Admin/v2/sites/f1e58e70-926f-4786-877b-
cf62abfdf0db/filesystem/folders/%252Fusr%252Fdyelacic%252Fsunroom%252F/pe
rmissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281" }, "relationships":
{ "trustee": { "data": { "id": "04dcfd20-fc73-41d0-a0d8-62d1706e6281",
"meta": { "name": "dyelacic" }, "type": "user" }, "links": { "related":
"/Admin/v2/sites/f1e58e70-926f-4786-877b-cf62abfdf0db/users/04dcfd20-
fc73-41d0-a0d8-62d1706e6281" } } }, "type": "folderPermissions" } }
```

PATCH PowerShell Sample

```
# change folder permissions
$update =
'{
  "data": {
    "type": "folderPermissions",
    "attributes": {
      "canUploadFile": false,
      "canDownloadFile": true,
      "canDeleteFile": false,
      "canRenameFileOrSubfolder": true,
      "canAppendFile": true,
      "canDeleteSubfolder": false,
      "canCreateSubfolder": true,
      "canSeeHiddenFileOrSubfolder": true,
      "canSeeReadOnlyFileOrSubfolder": true,
      "canSeeInParentList": true,
      "canList": true
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 30
# change folder permissions
Write-Output "change folder permissions"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2fusr%2fdyelacic%2ftuneroom%2f/
permissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281" -Method 'PATCH' -
Headers $authHeader -Body $update
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/filesystem/folders/{folderid}/permissions/{folderpermid}

DELETE Body Sample

```
# delete folder permissions

###
DELETE
https://{{host}}/admin/v2/sites/{{Siteid}}/filesystem/folders/%2Fusr%2Fdy
elacic%2Fsunroom%2F/permissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Fri, 11 Feb 2022 22:06:08 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
#
# delete folder permissions

Write-Output "delete folders permissions"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri
"$baseURL/v2/sites/$siteID/filesystem/folders/%2Fusr%2Fdyelacic%2ftuneroom%2f/
permissions/user-04dcfd20-fc73-41d0-a0d8-62d1706e6281" -Method 'DELETE' -
Headers $authHeader
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Event Rules REST API

Endpoint://sites/{siteid/event-rules

Get or create site event rules

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rules

GET Body Sample

```
### GET SITE EVENT RULES ###
GET https://{{host}}/admin/v2/sites/{{Siteid}}/event-rules HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
{"data":[{"type":"eventRule","id":"8d11ec4f-3c6c-4ab9-8045-
fb5c94b38ca0","attributes":{"info":{"Description":"Backup and
Cleanup","Enabled":true,"Folder":"00000000-0000-0000-0000-
000000000000","Name":"Backup and Cleanup","Next":"a220530d-c190-4fa8-
ae4a-
52240f6fee94","Remote":false,"Type":"Timer","apiVersion":"v2","eventRuleV
ersion":"416"},"statements":{"StatementsList":[{"ActionStatement":{"Actio
n":{"CleanupAction":{"DaysToKeepFiles":30,"FileMask":"*.bak","Folder":"C:
\\ProgramData\\Globalscape\\EFT
Server\\Backup","Include":true,"Recursive":false,"TimestampType":"Both"}}
,"IfFailedActions":[]},"ConditionStatement":{"Conditions":[],"ElseAttache
dStatements":[],"IfAttachedStatements":[],"Operator":"Or"},"LoopDatasetSt
atement":{"Params":{"DatasetName":"","EndLoopRow":0,"OrderingColumn":"","
```

GET PowerShell Sample

```
### Returns event rule settings ###
### User must obtain the SiteID from Get Sites
Write-Output "ER"
Write-Output "----"
#$siteID = Enter SiteID here
$ERList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rules" -
Method 'GET' -Headers $authHeader
Write-Output $ERList | ConvertTo-Json
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rules

POST Body Sample

```
### CREATE EVENT RULES ###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/event-rules HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
      "type": "eventRule",
      "id": "a220530d-c190-4fa8-ae4a-52240f6fee94",
      "attributes": {
        "info": {
          "Description": "Execute a specified action one time or repeat
```

POST Response Body Sample

```
{"data":{"data":[{"type":"eventRule","id":"c2320fbb-e7cc-4fde-8b41-
ffa655e71e50","attributes":{"info":{"Description":"Execute a specified
action one time or repeat at a specified
interval.","Enabled":true,"Folder":"00000000-0000-0000-0000-
000000000000","Name":"SendToDrobinson","Next":"00000000-0000-0000-0000-
000000000000","Remote":false,"Type":"Timer","apiVersion":"v2","eventRuleV
ersion":"416"},"statements":{"StatementsList":[{"ActionStatement":{"Actio
n":null,"IfFailedActions":[]},"ConditionStatement":{"Conditions":[{"Conte
xtVariable":"","Negate":false,"Operator":"Equals","PropertyID":1000,"Valu
e":{"Flag":false,"Value":"Yes"}}],"ElseAttachedStatements":[],"IfAttached
Statements":[{"ActionStatement":{"Action":{"UploadAction":{"ConnectionPro
fileGUID":"00000000-0000-0000-0000-
000000000000","LocalPath":"C:\\Users\\Administrator\\Desktop\\EFT.exe","R
emotePath":"/%SOURCE.FILE_NAME%","RemoveSource":false,"TransferSettings":
{"Connection":{"CertificateSettings":{"CertId":"00000000-0000-0000-0000-
```

POST PowerShell Sample

```
### CREATE EVENT RULES ###
##POST SITE EVENT RULE settings
Write-Output "Site Event POST"
Write-Output "-----"
$update =

'{
  "data": {
      "type": "eventRule",
      "attributes": {
        "info": {
          "Description": "Execute a specified action one time or repeat at a
specified interval.",
          "Enabled": true,
          "Folder": "00000000-0000-0000-0000-000000000000",
          "Name": "SecondSendToDrobinson",
          "Next": "00000000-0000-0000-0000-000000000000",
          "Remote": false,
```

Endpoint: /sites/{siteid}/event-rules/{ruleid}

Get, create, modify, or delete a specific site event rule

- GET
- POST
- PATCH
- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rules/{ruleid}

GET Body Sample

```
### GET SPECIFIC EVENT RULE ###
@eventRuleID = {{eventRuleList.response.body.$.data[0].id}}
GET https://{{host}}/admin/v2/sites/{{Siteid}}/event-
rules/{{eventRuleID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{"data":{"type":"eventRule","id":"8d11ec4f-3c6c-4ab9-8045-
fb5c94b38ca0","attributes":{"info":{"Description":"Backup and
Cleanup","Enabled":true,"Folder":"00000000-0000-0000-0000-
000000000000","Name":"Backup and Cleanup","Next":"c2320fbb-e7cc-4fde-
8b41-
ffa655e71e50","Remote":false,"Type":"Timer","apiVersion":"v2","eventRuleV
ersion":"416"},"statements":{"StatementsList":[{"ActionStatement":{"Actio
n":{"CleanupAction":{"DaysToKeepFiles":30,"FileMask":"*.bak","Folder":"C:
\\ProgramData\\Globalscape\\EFT
Server\\Backup","Include":true,"Recursive":false,"TimestampType":"Both"}}
,"IfFailedActions":[]},"ConditionStatement":{"Conditions":[],"ElseAttache
dStatements":[],"IfAttachedStatements":[],"Operator":"Or"},"LoopDatasetSt
atement":{"Params":{"DatasetName":"","EndLoopRow":0,"OrderingColumn":"","
```

GET PowerShell Sample

```
### Returns event rule settings ###
### User must obtain the SiteID from Get Sites
Write-Output "ER"
Write-Output "----"
#$siteID = Enter SiteID here
$ERList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rules" -
Method 'GET' -Headers $authHeader
Write-Output $ERList | ConvertTo-Json

$ERID = $ERList.data[0].id

$ERsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-
rules/$ERID" -Method 'GET' -Headers $authHeader
Write-Output $ERsList | ConvertTo-Json
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rules/{ruleid}

POST Body Sample

```
###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/event-rules/2d083578-61de-426f-b357-
96b24d1d45ae HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
{
 "parameters": {
  "operation": "run",
  "parameters": {
   "variables": [
    {
     "name": "On Scheduler (Timer) Event Rule",
     "value": "true"
    }
   ]
  }
 }
}
```

POST Response Body Sample

```
{ "result": { "result": "success", "variables": [ { "name":
"EVENT.DATESTAMP", "value": "20220211" }, { "name":
"EVENT.DATE_TIME_ISO8601", "value": "2022-02-11T22:33:34Z" }, { "name":
"EVENT.EVENTNAME", "value": "On Scheduler (Timer) Event Rule" }, {
"name": "EVENT.NAME", "value": "Scheduler (Timer) Event" }, { "name":
"EVENT.REASON", "value": "Action completed successfully" }, { "name":
"EVENT.TIME", "value": "11 Feb 22 22:33:34" }, { "name":
"EVENT.TIMESTAMP", "value": "223334" }, { "name":
"EVENT.TIMESTAMP_PRECISE", "value": "223334784" }, { "name":
"EVENT.TRANSACTION_ID", "value": "A555F2DC-8B8A-11ec-A822-0022482F4EB1"
}, { "name": "On Scheduler (Timer) Event Rule", "value": "true" }, {
"name": "SERVER.INSTALL_DIRECTORY", "value": "C:\\Program Files
(x86)\\Globalscape\\EFT Server" }, { "name": "SERVER.LOG_LOCATION",
"value": "C:\\ProgramData\\Globalscape\\EFT Server\\Logs\\" }, { "name":
"SERVER.LOG_TYPE", "value": "W3C Extended Log File Format" }, { "name":
"SERVER.NODE_NAME", "value": "DTY2016WinServe" }, { "name":
"SERVER.STATUS", "value": "Yes" }, { "name":
```

POST PowerShell Sample

```
# RPC to execute an event rule

$update =
'{
  "parameters": {
    "operation": "run",
    "parameters": {
      "variables": [
        {
          "name": "On Scheduler (Timer) Event Rule",
          "value": "true"
        }
      ]
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 30
# run an event rule
Write-Output "rpc to run an event rule"
Write-Output "-----"
$folderList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID//event-
rules/2d083578-61de-426f-b357-96b24d1d45ae" -Method 'POST' -Headers
$authHeader -Body $update
Write-Output $folderList | ConvertTo-Json
foreach ($user in $folderList.data)
{
    Write-Output ("User $($user.id): $($user.attributes.loginName)")
}
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rules/{ruleid}

PATCH Body Sample

```
###

PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/event-
rules/{{eventRuleID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data": {
      "type": "eventRule",
      "attributes": {
        "info": {
          "Description": "Execute a specified action one time or repeat
at a specified interval.",
          "Name": "Backups Revisited"
        }
    }
  }
}
```

PATCH Response Body Sample

```
{"data":{"type":"eventRule","id":"8d11ec4f-3c6c-4ab9-8045-
fb5c94b38ca0","attributes":{"info":{"Description":"Execute a specified
action one time or repeat at a specified
interval.","Enabled":true,"Folder":"00000000-0000-0000-0000-
000000000000","Name":"Backups Revisited","Next":"c2320fbb-e7cc-4fde-8b41-
ffa655e71e50","Remote":false,"Type":"Timer","apiVersion":"v2","eventRuleV
ersion":"416"},"statements":{"StatementsList":[{"ActionStatement":{"Actio
n":{"CleanupAction":{"DaysToKeepFiles":30,"FileMask":"*.bak","Folder":"C:
\\ProgramData\\Globalscape\\EFT
Server\\Backup","Include":true,"Recursive":false,"TimestampType":"Both"}}
,"IfFailedActions":[]},"ConditionStatement":{"Conditions":[],"ElseAttache
dStatements":[],"IfAttachedStatements":[],"Operator":"Or"},"LoopDatasetSt
atement":{"Params":{"DatasetName":"","EndLoopRow":0,"OrderingColumn":"","
SortOrder":"Original","StartLoopRow":0,"UseEndLoopRow":false,"UseStartLoo
pRow":false},"Statements":[]},"Type":"Action"},{"ActionStatement":{"Actio
n":{"CleanupAction":{"DaysToKeepFiles":30,"FileMask":"*.*","Folder":"C:\\
ProgramData\\Globalscape\\EFT
Server\\Logs","Include":true,"Recursive":false,"TimestampType":"Both"}},"
IfFailedActions":[]},"ConditionStatement":{"Conditions":[],"ElseAttachedS
tatements":[],"IfAttachedStatements":[],"Operator":"Or"},"LoopDatasetStat
```

PATCH PowerShell Sample

```
##Patch SITE EVENT RULE settings
Write-Output "Site Event Rule PATCH"
Write-Output "-----"
$update =
'{
  "data": {
      "type": "eventRule",
      "attributes": {
        "info": {
          "Description": "Execute a specified action one time or repeat at a
specified interval.",
          "Name": "Backups Revisited"
        }
    }
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-
rules/$ERID" -Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $patchReturn | ConvertTo-Json
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rules/{ruleid}

DELETE Body Sample

```
### GET SPECIFIC EVENT RULE ###
@eventRuleID = {{eventRuleList.response.body.$.data[0].id}}
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/event-
rules/{{eventRuleID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content
Date: Tue, 09 Aug 2022 12:24:53 GMT
Content-Length: 0
Cache-Control: no-cache, no-store, must-revalidate
Expires: -1
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Content-Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
Referrer-Policy: no-referrer
```

DELETE PowerShell Sample

```
$ERID = $ERList.data[1].id

$ERsList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-
rules/$ERID" -Method 'DELETE' -Headers $authHeader
Write-Output $ERsList | ConvertTo-Json
```

Endpoint:/sites/{siteid}/event-rule-folders

Get or create site event-rule-folders

- GET
- POST

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rule-folders

GET Body Sample

```
### GET EVENT RULE FOLDER LIST ###

GET https://{{host}}/admin/v2/sites/{{Siteid}}/event-rule-folders HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
```

GET Response Body Sample

```
{"data":[{"attributes":{"name":"<root>"},"id":"00000000-0000-0000-0000-000000000000","link":{"self":"/admin/v2/sites/6203305a-d7c6-499e-a59b-81500765f8f2/event-rule-folders/00000000-0000-0000-0000-000000000000"},"type":"eventRuleFolder"}],"links":{"self":"/admin/v2/sites/6203305a-d7c6-499e-a59b-81500765f8f2/event-rule-folders"}}
```

GET PowerShell Sample

```
### Returns all site settings for the specified Site ID ###
### User must obtain the SiteID from Get Sites
Write-Output "ER Folders"
Write-Output "----"
#$siteID = Enter SiteID here
$siteList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rule-folders" -Method 'GET' -Headers $authHeader
Write-Output $siteList | ConvertTo-Json
```

Method & URL

POST https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rule-folders

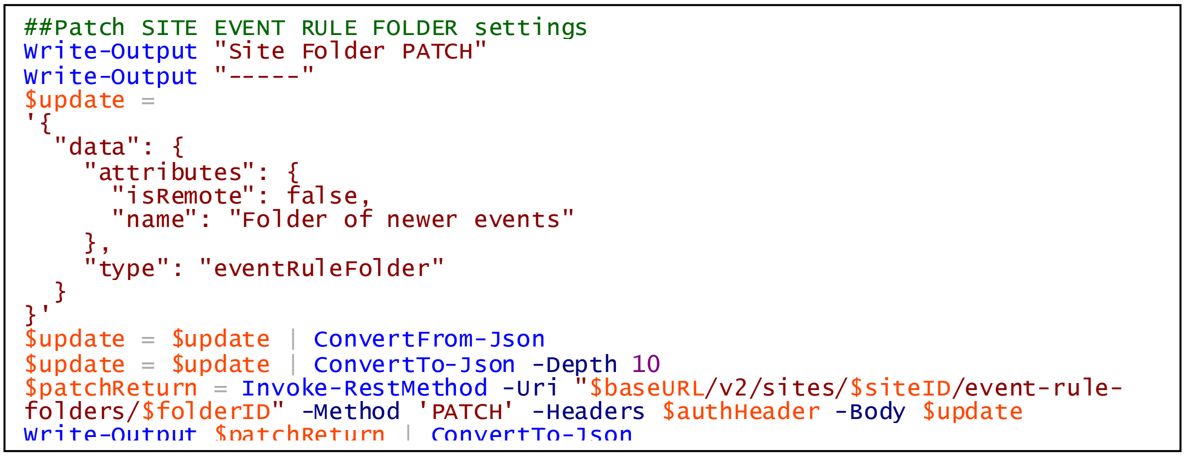
POST Body Sample

```
###
POST https://{{host}}/admin/v2/sites/{{Siteid}}/event-rule-folders
HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}

{
  "data":
    {
      "attributes": {
        "name": "Folder of user events"
      },
      "type": "eventRuleFolder"
    }
}
###
```

POST Response Body Sample

```
{"data":{"attributes":{"isRemote":false,"name":"Folder of user
events"},"id":"38d270a9-6b40-4732-9942-
5bdac9731d93","type":"eventRuleFolder"},"links":{"self":"/admin/v2/sites/
6203305a-d7c6-499e-a59b-81500765f8f2/event-rule-folders/38d270a9-6b40-
4732-9942-5bdac9731d93"}}
```

POST PowerShell Sample



```
##Patch SITE EVENT RULE FOLDER settings
Write-Output "Site Folder PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes": {
      "isRemote": false,
      "name": "Folder of newer events"
    },
    "type": "eventRuleFolder"
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rule-
folders/$folderID" -Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $patchReturn | ConvertTo-Json
```

Endpoint:/sites/{siteid}/event-rule-folders/{folderid}

Get, update, or delete a specific site event rule folder data

- GET
- PATCH

- DELETE

Method & URL

GET https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rule-folders/{folderid}

GET Body Sample

```
###
@folderID = {{foldersList.response.body.$.data[0].id}}
GET https://{{host}}/admin/v2/sites/{{Siteid}}/event-rule-
folders/{{folderID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

GET Response Body Sample

```
HTTP/1.1 200 OK Date: Tue, 09 Aug 2022 17:23:43 GMT Content-Type:
application/vnd.api+json ETag: "98C84003" Content-Length: 255 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "attributes": { "isRemote": false, "name":
"<root>" }, "id": "00000000-0000-0000-0000-000000000000", "type":
"eventRuleFolder" }, "links": { "self": "/admin/v2/sites/6203305a-d7c6-
```

GET PowerShell Sample

```
Write-Output "ER Folders"
Write-Output "----"
#$siteID = Enter SiteID here
$folderList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rule-
folders" -Method 'GET' -Headers $authHeader
Write-Output $folderList | ConvertTo-Json

$folderID = $folderList.data[1].id

$foldersList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rule-
folders/$folderID" -Method 'GET' -Headers $authHeader
Write-Output $foldersList | ConvertTo-Json
```

Method & URL

PATCH https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rule-folders/{folderid}

PATCH Body Sample

```
PATCH https://{{host}}/admin/v2/sites/{{Siteid}}/event-rule-
folders/{{folderID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}


{
  "data": {
    "attributes": {
      "isRemote": false,
      "name": "Folder of new events"
    },
    "type": "eventRuleFolder"
  }
}
###
```

PATCH Response Body Sample

```
HTTP/1.1 200 OK Date: Tue, 09 Aug 2022 18:07:40 GMT Content-Type:
application/vnd.api+json ETag: "1268DCC6" Content-Length: 269 Cache-
Control: no-cache, no-store, must-revalidate Expires: -1 X-Content-Type-
Options: nosniff X-XSS-Protection: 1; mode=block Content-Security-Policy:
default-src 'self' 'unsafe-inline' 'unsafe-eval' data:; Referrer-Policy:
no-referrer { "data": { "attributes": { "isRemote": false, "name":
"Folder of new events" }, "id": "38d270a9-6b40-4732-9942-5bdac9731d93",
"type": "eventRuleFolder" }, "links": { "self":
```

PATCH PowerShell Sample

```
##Patch SITE EVENT RULE FOLDER settings
Write-Output "Site Folder PATCH"
Write-Output "-----"
$update =
'{
  "data": {
    "attributes": {
      "isRemote": false,
      "name": "Folder of newer events"
    },
    "type": "eventRuleFolder"
  }
}'
$update = $update | ConvertFrom-Json
$update = $update | ConvertTo-Json -Depth 10
$patchReturn = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rule-
folders/$folderID" -Method 'PATCH' -Headers $authHeader -Body $update
Write-Output $patchReturn | ConvertTo-Json
```

Method & URL

DELETE https://[server URL]:[port]/admin/v2/sites/{siteId}/event-rule-folders/{folderid}

DELETE Body Sample

```
###
@folderID = {{foldersList.response.body.$.data[0].id}}
DELETE https://{{host}}/admin/v2/sites/{{Siteid}}/event-rule-
folders/{{folderID}} HTTP/1.1
Authorization: EFTAdminAuthToken {{AuthToken}}
###
```

DELETE Response Body Sample

```
HTTP/1.1 204 No Content Date: Tue, 09 Aug 2022 18:12:00 GMT Content-
Length: 0 Cache-Control: no-cache, no-store, must-revalidate Expires: -1
X-Content-Type-Options: nosniff X-XSS-Protection: 1; mode=block Content-
Security-Policy: default-src 'self' 'unsafe-inline' 'unsafe-eval' data:;
```

DELETE PowerShell Sample

```
$folderID = $folderList.data[1].id

$folderList = Invoke-RestMethod -Uri "$baseURL/v2/sites/$siteID/event-rule-
folders/$folderID" -Method 'DELETE' -Headers $authHeader
Write-Output $folderList | ConvertTo-Json
```