



# Revolution

## Web API

### Disclaimers:

- All information within this document is subject to change until the final version is released.

© 2021 Intrado

# Web API

The Revolution Web API allows you to build applications that interact with Revolution in more complex ways than the integrations we provide out of the box.

## Authentication

Authenticate your Web API requests by providing a basic token, which identifies a single user.

In the Revolution Web UI, navigate to the Users page, and create a new User with the Role - WebApi. After successfully saving the user, you may fetch the WebApi Token value from the User page and provide it as Basic Authentication token in all api calls.

Role  
WebApi

---

WebApi Token  
Mp1jPP4kBEqKgHgshxYNaw==

For example, execute a cURL command to fetch all notifications:

```
curl -H "Authorization: Basic qlcO8uBOSk6JnLn8S2i9iA==" http://<Server-IP>/unifiedgui/webapi/Notifications/
```

WebApi user profile can be assigned a comma delimited set of IpAddresses from which all Http requests will be entertained. Leaving this blank will accept requests for all client IpAddresses (default behavior). Providing one or more IpAddress is a more restrictive usage of the webapi authentication token. If an Http request is received from the client address that is not declared in the webapi user profile then the server will return **HttpStatus code 401 Unauthorized**.

## Odata syntax

OData (Open Data Protocol) is an ISO/IEC approved, OASIS standard that defines a set of best practices for building and consuming RESTful APIs. Revolution uses Odata for its REST interface.

## Filtering with Odata

OData supports various kinds of query options for querying data. This section will help you go through the common scenarios for these query options.

### System Query Option \$filter

The \$filter system query option allows you to filter response data, and only items where the expression evaluates to true are included in the response.

There are several kinds of basic built-in functions for \$filter, including logical operators and arithmetic operators. The request below uses \$filter to get people with FirstName "Scott".

```
GET serviceRoot/People?$filter=FirstName eq 'Scott'
```

### Filter on Complex Type

\$filter can also work on complex types. The request below returns airports with "San Francisco" contained in its Address. And Address is property of complex type Location.

```
GET serviceRoot/Airports?$filter=contains(Location/Address, 'San Francisco')
```

You can use **and** and **or** in \$filter, for example:

```
$filter=contains(Site, 'Denver') and contains(Name, 'One')
```

## Redundancy

If you are using Revolution in a redundancy setup, there are two options for using the Revolution API: use a load balancer and send all API commands to the load balancer for routing to the active system or send all commands to both systems. If you choose to send all commands to both systems, you should expect commands to the primary server to timeout if that server is unavailable. You should expect the secondary server to respond with "Status": "ServerIsStandby" when the primary server is active.

## Setup Methods

## Resource Information

GET, POST, PUT, PATCH calls require the below Resource header for all REST calls.

All POST calls return an appropriate response object, returns HTTP Status 201 for successful creates and in some cases successful updates.

PATCH and PUT calls return HTTP Status 200 for success.

DELETE calls return a HTTP Status 404 for success.

All POST and PUT HTTP calls require *full object in the payload* (unless indicated otherwise).

PATCH calls are allowed partial objects are payloads.

For Authorization failures, the API returns HTTP Status 401 Unauthorized.

For all other errors and/or other failures in the server, the API call returns HTTP Status 500 or 400.

Response formats	JSON
Content-Type	application/json
Authorization (header)	Basic Authentication token

## System Settings

These are sets of global settings that apply to the overall operation of Revolution.

```
GET: http://<ServerIP>/unifiedgui/webapi/System/
```

### Example Response

```
{
  "@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#System",
  "value": [
    {
      "Name": "MulticastEnabled",
      "Description": "Only unicast is used when unchecked.",
      "Value": "True",
      "IsReadOnly": false,
```

```

        "AcceptableValues": [
            ]
        },
        {
            "Name": "MulticastThreshold",
            "Description": "Use multicast when the notification has
this many endpoints (ignored if multicast is not enabled).",
            "Value": "1",
            "IsReadOnly": false,
            "AcceptableValues": [
                ]
        },
        {
            "Name": "MulticastIPAddress",
            "Description": "Your network multicast IP address.",
            "Value": "239.10.10.10",
            "IsReadOnly": false,
            "AcceptableValues": [
                ]
        },
    },
}

```

For a complete set of all system settings, query the URL.

## Update System Settings

One or more system settings may be updated. Upon a successful POST, system responds with HttpStatus 200 OK and no content in the body.

```
POST: http:// <ServerIP>/unifiedgui/webapi/UpdateSystem/
```

### Example Request

```

{
    "settings": [
        {

```

```

        "Name": "MulticastEnabled",
        "Description": "Only unicast is used when unchecked.",
        "Value": "True",
        "IsReadOnly": false,
        "AcceptableValues": []
    },
    {
        "Name": "MulticastThreshold",
        "Description": "Use multicast when the notification has
this many endpoints (ignored if multicast is not enabled).",
        "Value": "1",
        "IsReadOnly": false,
        "AcceptableValues": []
    }
]
}

```

## Service Modules

To see a list of all registered modules within Revolution, issue the below GET. There is no update or delete API for service modules.

```
GET: http://<ServerIP>/unifiedgui/webapi/ServiceModules/
```

or to find a specific module such a SIPActivator

```
GET: http://localhost/unifiedgui/webapi/ServiceModules?$filter=Name eq
'SIPActivator'
```

### Example Response

```

{
  "@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#ServiceModules",
  "value": [
    {
      "Id": "70eaed1a-e1f3-45b2-9a2f-208cd5c7263c",

```

```

        "Name": "SIPActivator",
        "Description": "First, configure your SIP trunk and define
a range of SIP lines in your phone system communications manager. Then
add SIP lines here, which are made available to assign to notifications
to allow end users to send those notifications by dialing extension num-
bers on their IP phone.",
        "Version": "1.0.0.0",
        "IsActivator": true,
        "IsNotifier": false,
        "IsActive": true
    },
    {
        "Id": "129a11ce-0cad-4f05-9c9b-8eb8ca738308",
        "Name": "CiscoNotifier",
        "Description": "Most of the communication between this
application and your Cisco phone system is configured in your Cisco
call manager. Once configured, supported IP phones automatically
register.",
        "Version": "1.0.0.0",
        "IsActivator": false,
        "IsNotifier": true,
        "IsActive": true
    }
]
}

```

## Query Methods

Filter parameters are permissible on all GET queries. These are applicable for GETs on all service modules.

**\$filter** - Use this query parameter to include filters on the entity you are querying.

**\$count=true** - Use this query parameter to get a total count of entities that were received in the query. The count value is returned in the response.

**\$orderby=<fieldName> <asc/desc>** - Use to return query returns in specific order.

**\$top=n** - Use this query parameter to return only first 'n' number of records in the query.

**\$skip=n** - Use this query parameter to skip first 'n' number of records.

**\$top** and **\$skip** - Use these query parameters to construct pagination in collections of records.

Example

```
{
"@odata.count": 10,
"value": [ ... ]
}
```

## SIPActivator Settings

### Settings Object

<b>DigestRealm</b> (optional)	The digest realm setting for the trunk.
<b>Username</b> (optional)	Username setting for the trunk.
<b>Password</b> (optional)	Password setting for the trunk. Clients (e.g. PBX systems) are prompted for these credentials when connecting. If blank, auth is not required.
<b>PinTimeout</b> (optional)	Number of seconds before the security code times out.
<b>STUNServer</b> (optional)	Server used for STUN tunneling.
<b>OutboundProxyServers</b> (optional)	Proxy used for outbound SIP connections.
<b>SIPPort</b> (optional)	Default is 5060.
<b>TrustedServers</b> (optional)	Comma-separated list of trusted server IP or names.
<b>EnableTLS</b> (optional)	Set to True or False.

### Get SIPActivator Module Settings

```
GET: http://<ServerIP>/unifiedgui/webapi/SipActivator/
```

Example Response



```
{
"@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#SipActivator/$entity",
"Id": "70eaed1a-e1f3-45b2-9a2f-208cd5c7263c",
"Name": "SIPActivator",
"DigestRealm": "3344",
"Username": "someuser",
>Password": "somepassword",
"PinTimeoutSeconds": "15",
"STUNServer": "",
"SIPPort": "",
"OutboundProxyServers": "",
"TrustedServers": "",
"EnableTLS": false,
"DisableReinvites": false
}
```

## Update SIPActivator Settings

Upon successful update, system responds with HttpStatus 200 OK and no content in the body.

```
PUT: http://<ServerIP>/unifiedgui/webapi/SipActivator(70eaed1a-e1f3-
45b2-9a2f-208cd5c7263c)/
```

### Example Request body

```
{
"Id": "70eaed1a-e1f3-45b2-9a2f-208cd5c7263c",
"Name": "SIPActivator",
"DigestRealm": "3344",
"Username": "someuser",
>Password": "somepassword",
"PinTimeoutSeconds": "25",
"STUNServer": "",
"SIPPort": "",
"OutboundProxyServers": "",
"TrustedServers": "",
}
```

```
"EnableTLS": false,
"DisableReinvites": false
}
```

**Note:** Another option to Update SIPActivator settings is to issue a PATCH and only update those property values that need a change instead of the entire object.

```
PATCH: http://<ServerIP>/unifiedgui/webapi/SipActivator(70eaed1a-e1f3-45b2-9a2f-208cd5c7263c) /
```

#### Example Request

```
{
  "Id": "70eaed1a-e1f3-45b2-9a2f-208cd5c7263c",
  "PinTimeoutSeconds": "25",
  "EnableTLS": true
}
```

## SIPActivator Actions

### SIP Line Actions Object

<b>Name (required)</b>	The type of action you are creating.
<b>Extension (required)</b>	The extension number for the SIP line.
<b>SecurityCode (optional)</b>	A security code for the line.
<b>ActivatorTextTitle (optional)</b>	Title text to be passed from activator to notification.
<b>ActivatorTextBody (optional)</b>	Body text to be passed from activator to notification.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites.

### Cloud Trunk Registration Actions Object

<b>Name (required)</b>	The type of action you are creating.
<b>RegistrarUri (required)</b>	Registrar server URI. <i>For example: sip:domain.com</i>
<b>AddressOfRecord (optional)</b>	Registration address of record. <i>For example: user@domain.com</i>
<b>RegisterInterval (optional)</b>	Interval in seconds.

Realm (optional)	
Username	
Password	
Site	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites.

## Get All SIP Actions

```
GET: http://<ServerIP>/unifiedgui/webapi/SipActivator/Actions/
```

### Example Response

```
{
  "@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#SipActivator",
  "value": [
    {
      "Id": "bd19e208-983a-4409-bbd4-35081f7e9162",
      "Name": "sipaction1",
      "Extension": "2323",
      "SecurityCode": "5454",
      "ActivatorTextTitle": "title text updated",
      "ActivatorTextBody": "body text updated",
      "ActionTypeName": "SIP Line",
      "Site" : ""
    },
    {
      "Id": "827fbee7-ec74-4b23-a32e-0f3d31c66b16",
      "Name": "sipaction2",
      "Extension": "1232",
      "SecurityCode": "222222",
      "ActivatorTextTitle": "",
      "ActivatorTextBody": "",
      "ActionTypeName": "SIP Line",
      "Site" : "PDX"
    }
  ]
}
```

```
},  
.....  
}
```

## Get a Specific SIPActivator Action

```
http://<ServerIP>/unifiedgui/webapi/SipActivator/Action(actionId=  
d=00af4db7-1c23-4a1b-ad44-8b7303b56505) /
```

### Example Response

```
{  
"@odata.context": "http://localhost/unifiedgui/webapi/$metadata#sipacti-  
ons/$entity",  
"Id": "00af4db7-1c23-4a1b-ad44-8b7303b56505",  
"Name": "cloudreg",  
"RegistrarUri": "sip:192.168.11.71",  
"AddressOfRecord": "user@192.168.11.68",  
"RegisterInterval": "300",  
"Site" : "PDX",  
"Realm": "",  
"Username": "username",  
"Password": "bXN1YnJhbWE=",  
"ActionTypeName": "Cloud Trunk Registration"  
}
```

## Filter on SIPActivator Actions

Refer to the *Query Methods* sections for how to filter on GET calls.

```
http://<Server-  
IP>/unifiedgui/webapi/SipActivator/Actions/?$filter=contains (Name,  
'action')  
http://<Server-  
IP>/unifiedgui/webapi/SipActivator/Actions/?$filter=Extension ge '2000'
```

And / Or operators are allowed in queries.

```
http://<ServerIP>/unifiedgui/webapi/SipActivator/Actions/?$filter=Name
eq 'action1' and Extension ge '5000'
```

```
http://<ServerIP>/unifiedgui/webapi/SipActivator/Actions/?$filter=Name
eq 'action1' or Name eq 'action2'
```

*[Note: You can use OR operators to apply different filters on the same field. However, you cannot use OR operators to apply filters on two different fields.]*

If the queries result in no matching records, the response is an empty collection.

#### Example Response

```
{
"@odata.context": "http://localhost/unifiedgui/webapi/$metadata#SipActi-
vator",
"value": []
}
```

## Add or Update SIPActivator Actions

Revolution allows managing two types of SIP activator actions.

- Sip Lines
- Cloud Trunk Registration

## Manage SIP Line Actions

One or more SIP lines may be added using the URL below.

```
POST: http://<ServerIP>/unifiedgui/webapi/SipActivator/SipLineActions/
```

#### Example Request

```
{"actions": [{
  "Name": "siplineaction1",
  "Extension": "1010",
  "SecurityCode": "4321",
  "ActivatorTextTitle": "title text updated",
  "ActivatorTextBody": "body text updated",
  "ActionTypeName": "SIP Line",
```

```
"Site" : "PDX"
}]
}
```

## Manage Cloud Trunk Registration Action

One or more cloud trunk registrations may be added using the URL below.

```
POST: http://<ServerIP>/unifiedgui/webapi/SipActivator/CloudTrunkAction-
s/
```

### Example Request

```
{
  "actions": [{
    "Name": "cloudreg",
    "RegistrarUri": "sip:192.168.11.71",
    "AddressOfRecord": "user@192.168.11.68",
    "RegisterInterval": "300",
    "Realm": "",
    "Username": "someuser",
    "Password": "bXN1YnJhbWE=" ,
    "Site" : "PDX"
  }]
}
```

## Delete SIPActivator Actions

```
POST: http://<ServerIP>/unifiedgui/webapi/SipActivator/DeleteActions/
```

### Example Request

- This POST deletes actions with below IDs.

```
{ "ids" : ["68782574-051a-4000-b585-a056edbbf2fa", "7sdf42574-051a-
4000-b585-a056edbbf2fa"] }
```

## CiscoNotifier Settings

Cisco Notifier settings can be retrieved by issuing a GET on the below URL.

## Settings Object

```
GET: http://<ServerIP>/unifiedgui/webapi/CiscoNotifier/
```

<b>CacheUpdateIntervalMinutes</b>	How often (in minutes) to update the list of phones from Call Manager.
<b>CacheFullUpdateMultiple</b>	Full update every Nth update, otherwise partial update.
<b>PhoneAuthUsername</b>	Username used for phone authentication.
<b>PhoneAuthPassword</b>	Password used for phone authentication.
<b>AuthenticatorPort</b>	Authenticator Port.
<b>CustomAuthenticatorForwardUrl</b>	If not configured, the URL is dynamically set using the pattern http://<ip>:<port>/ccmcp/authenticate.jsp where <ip> is the first Call Manager.
<b>ClusteredCallManager</b>	If multiple call managers are used, set whether they are clustered or standalone.
<b>DXPort</b>	Listening port of Cisco DX series phones running the Android app.
<b>CallInterruptPriorityThreshold</b>	Notification priority must be higher than this value to interrupt an active phone call.
<b>ImportEndUsers</b>	If checked, contacts are created from Call Manger user accounts.

The above settings can be updated by a PATCH API call.

```
PATCH: http://<ServerIP>/unifiedgui/webapi/CiscoNotifier(129a11ce-0cad-4f05-9c9b-8eb8ca738308)
```

### Example Request

```
{  
  "Id": "129a11ce-0cad-4f05-9c9b-8eb8ca738308",  
}
```

```
"ClusteredCallManager": "False",  
"DXPort": "8008"  
}
```

## Cisco Actions

Cisco has 3 different types of Actions:

1. Phone Services
2. Phone Services Dialer
3. Dial Monitor
4. Call Manager

### Get Cisco Actions

Retrieve all Cisco actions by issuing a GET on the below URL

```
GET: http://<ServerIP>/unifiedgui/webapi/Cisco/Actions/
```

This will return a collection of Actions objects which could be a mix of the above-mentioned Action types.

```
GET: http://<ServerIP>/unifiedgui/webapi/Cisco/Action(actionId=  
d=b0c5b73e-39ff-e811-80cf-0050568f0d00)
```

The above will return a specific Action with Id.

[Note: Query filters may be applied on the Actions object. Refer to *Query Methods* portion of the document for more information]

**The following API calls are deprecated and redirected to a different URL as specified below**

```
POST:http://<ServerIP>/unifiedgui/webapi/CiscoNotifier/CiscoActions
```

```
Moved to POST: http://<ServerIP>/unifiedgui/webapi/Cisco/CallManager
```

```
GET: http://<ServerIP>/unifiedgui/webapi/CiscoNotifier/Actions
```

```
Moved to GET: http://<ServerIP>/unifiedgui/webapi/Cisco/Actions
```

```
GET: http://<ServerIP>/unifiedgui/webapi/CiscoNotifier/Action(actionId=  
d=b0c5b73e-39ff-e811-80cf-0050568f0d00)
```



```
Moved to GET: http://<ServerIP>/unifiedgui/webapi/Cisco/Action(actionId=
d=b0c5b73e-39ff-e811-80cf-0050568f0d00)
POST: http://<ServerIP>/unifiedgui/webapi/CiscoNotifier/DeleteActions
POST: http://<ServerIP>/unifiedgui/webapi/Cisco/DeleteActions
```

## Add or Update Cisco Actions

### Phone Service Actions Object

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>URL</b>	This is not a create or update property but will be provided on all GET of an action. This is a dynamically generated callback URL to trigger this action.

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Cisco/PhoneService/
```

#### Example Request

- This POST request adds one new PhoneService and updates another existing PhoneService (Note: the object with 'Id')

```
{ "actions" : [
  {
    "Name": "PhoneService_1",
    "Site": ""
  },
  {
    "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
    "Name": "PhoneService_2",
```

```
    "Site" : "San Jose site"
  }
]
}
```

**Example Response**

```
{
"Ok": true,
"Successes":[
{
"Id": "36a173f7-d026-4d76-8458-321e7f2eff47",
"Name": " PhoneService_1",
"Message": null
},
{
"Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
"Name": " PhoneService_2",
"Message": null
}
],
"Failures":[]
}
```

***Phone Service Dialer Actions Object***

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>Extension</b>	Dial an extension directly
<b>URL</b>	This is not a create or update property but will be provided on all GET of an actions. This is a dynamically generated callback URL to trigger this action.

Actions may be added/updated in collections of Actions objects.

POST: http://<ServerIP>/unifiedgui/webapi/Cisco/PhoneServiceDialer/

Follow the same example as provided for PhoneService

### Cisco Dial Monitor Actions Object

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>MonitoredNumber</b> (required)	Enter specific number to monitor or use ? and * wildcards (?11 matches 911 and 411; 2* matches any number dialed beginning with 2)
<b>CallingDNPattern</b>	Enter specific phone to call from or use ? and * wildcards (207? matches monitored number dialed from phones in 207x range; 20* matches monitored number dialed from phones beginning with 20). Leave blank to include all.
<b>CallingSearchSpace</b>	Only monitor numbers called from this search space. Leave blank to include all.
<b>Partition</b>	Only monitor numbers called from this partition. Leave blank to include all.
<b>DevicePool</b>	Only monitor numbers called from this device pool. Leave blank to include all.

Actions may be added/updated in collections of Actions objects.

POST: http://<ServerIP>/unifiedgui/webapi/Cisco/DialMonitor/

Follow the same example as provided for PhoneService

### Cisco Call Manager Action Objects

<b>Name</b> (required)	The type of action you are creating. (See below for list of valid actions.)
<b>IPAddress</b> (required)	IP address for the CUCM.

<b>Username</b> (required)	Username for the CUCM account.
<b>Password</b> (required)	Password for the CUCM account.
<b>Version</b> (required)	Call Manager Version number.
<b>PreferenceOrder</b> (optional)	Preference order for the CUCM.

One or more Cisco call managers may be added using the URL below.

POST: <http://<ServerIP>/unifiedgui/webapi/Cisco/CallManager/>

#### Example Request

- This POST request adds two Cisco Call Managers.

```
{ "actions" : [
  {
    "Name":"CiscoAction_1",
    "IPAddress" : "192.168.11.99",
    "Username" : "name",
    "Password" : "888888",
    "Version" : "10.5"
  },
  {
    "Name":"CiscoAction_2",
    "IPAddress" : "192.168.11.88",
    "Username" : "name",
    "Password" : "88888888",
    "Version" : "11.0"
  }
]
```

## Delete Cisco Actions

Delete Cisco Actions by sending a collection of Ids of actions that need to be deleted. The Ids could be a collection of mix of the above mentioned types of Cisco Actions.

POST: <http://<ServerIP>/unifiedgui/webapi/Cisco/DeleteActions/>

### Example Request

- This POST deletes actions with below IDs.

```
{ "ids" : ["68782574-051a-4000-b585-a056edbbf2fa", "7sdf42574-051a-4000-b585-a056edbbf2fa"] }
```

## Mitel Actions

Mitel has 2 different types of Actions:

1. Phone Services
2. Phone Services Dialer

### Get Mitel Actions

Retrieve all Mitel actions by issuing a GET on the below URL

```
GET: http://<ServerIP>/unifiedgui/webapi/Mitel/Actions/
```

This will return a collection of Actions object which could be a mix of the above-mentioned Action types.

```
GET: http://<ServerIP>/unifiedgui/webapi/Mitel/Action(actionId=d=b0c5b73e-39ff-e811-80cf-0050568f0d00)
```

The above will return a specific Action with Id.

[Note: Query filters may be applied on the Actions object. Refer to *Query Methods* portion of the document for more information]

### Phone Service Actions Object

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>URL</b>	This is not a create or update property but will be provided on all GET of an

actions. This is a dynamically generated callback URL to trigger this action.
---

## Add or Update Mitel Actions

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Mitel/PhoneService/
```

### Example Request

- This POST request adds one new PhoneService and updates another existing PhoneService (Note: the object with 'Id')

```
{ "actions" : [  
  {  
    "Name":"PhoneService_1",  
    "Site":""  
  },  
  {  
    "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",  
    "Name":"PhoneService_2",  
    "Site" : "San Jose site"  
  }  
]
```

### Example Response

```
{  
  "Ok": true,  
  "Successes": [  
    {  
      "Id": "36a173f7-d026-4d76-8458-321e7f2eff47",  
      "Name": " PhoneService_1",  
      "Message": null  
    },  
    {  
      "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
```

```
"Name": " PhoneService_2",
"Message": null
}
],
"Failures":[]
```

### Phone Service Dialer Actions Object

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>Extension</b>	Dial an extension directly
<b>URL</b>	This is not a create or update property but will be provided on all GET of an actions. This is a dynamically generated callback URL to trigger this action.

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Mitel/PhoneServiceDialer/
```

*Follow the same example as provided for PhoneService*

### Delete Mitel Actions

Delete Mitel Actions by sending a collection of Ids of actions that need to be deleted.

```
POST: http://<ServerIP>/unifiedgui/webapi/Mitel/DeleteActions/
```

#### Example Request

- This POST deletes actions with below IDs.

```
{ "ids" : ["68782574-051a-4000-b585-a056edbbf2fa", "7sdf42574-051a-4000-b585-a056edbbf2fa"] }
```

# Avaya Actions

Avaya has two different types of Actions:

- 1.Phone Services
- 2.Phone Services Dialer

## Get Avaya Actions

Retrieve all Avaya actions by issuing a GET on the below URL.

```
GET: http://<ServerIP>/unifiedgui/webapi/Avaya/Actions/
```

This will return a collection of Actions objects, which could be a mix of the above-mentioned Action types.

```
GET: http://<ServerIP>/unifiedgui/webapi/Avaya/Action(actionId=d=b0c5b73e-39ff-e811-80cf-0050568f0d00)
```

The above will return a specific Action with Id.

[Note: Query filters may be applied on the Actions object. Refer to *Query Methods* portion of the document for more information]

## Phone Service Actions Object

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>URL</b>	This is not a create or update property but will be provided on all GET of an actions. This is a dynamically generated callback URL to trigger this action.

## Add or Update Avaya Actions

Actions may be added/updated in collections of Actions objects.



```
POST: http://<ServerIP>/unifiedgui/webapi/Avaya/PhoneService/
```

#### Example Request

- This POST request adds one new PhoneService and updates another existing PhoneService (Note: the object with 'Id')

```
{  "actions" : [
    {
      "Name":"PhoneService_1",
      "Site":""
    },
    {
      "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
      "Name":"PhoneService_2",
      "Site" : "San Jose site"
    }
  ]
}
```

#### Example Response

```
{
  "Ok": true,
  "Successes":[
    {
      "Id": "36a173f7-d026-4d76-8458-321e7f2eff47",
      "Name": " PhoneService_1",
      "Message": null
    },
    {
      "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
      "Name": " PhoneService_2",
      "Message": null
    }
  ],
  "Failures":[]
}
```

## Phone Service Dialer Actions Object

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>Extension</b>	Dial an extension directly
<b>URL</b>	This is not a create or update property but will be provided on all GET of an actions. This is a dynamically generated callback URL to trigger this action.

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Avaya/PhoneServiceDialer/
```

*Follow the same example as provided for PhoneService*

## Delete Avaya Actions

Delete Avaya Actions by sending a collection of Ids of actions that need to be deleted.

```
POST: http://<ServerIP>/unifiedgui/webapi/Avaya/DeleteActions/
```

### Example Request

- This POST deletes actions with below IDs.

```
{ "ids" : ["68782574-051a-4000-b585-a056edbbf2fa", "7sdf42574-051a-4000-b585-a056edbbf2fa"] }
```

## Polycom Actions

Polycom has three different types of Actions:

1. Phone Services
2. Phone Services Dialer
3. Dial Monitor

## Get Polycom Actions

Retrieve all Polycom actions by issuing a GET on the below URL.

```
GET: http:// <ServerIP>/unifiedgui/webapi/Polycom/Actions/
```

This will return a collection of Actions objects which could be a mix of the above-mentioned Action types.

```
GET: http://<ServerIP>/unifiedgui/webapi/Polycom/Action(actionId=d=b0c5b73e-39ff-e811-80cf-0050568f0d00)
```

The above will return a specific Action with Id.

[Note: Query filters may be applied on the Actions object. Refer to *Query Methods* portion of the document for more information]

### *Phone Service Actions Object*

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>URL</b>	This is not a create or update property but will be provided on all GET of an actions. This is a dynamically generated callback URL to trigger this action.

## Add or Update Polycom Actions

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Polycom/PhoneService/
```

### Example Request

- This POST request adds one new PhoneService and updates another existing PhoneService (Note: the object with 'Id')

```
{ "actions" : [
```

```

{
  "Name": "PhoneService_1",
  "Site": ""
},
{
  "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
  "Name": "PhoneService_2",
  "Site": "San Jose site"
}
]
}

```

**Example Response**

```

{
  "Ok": true,
  "Successes": [
    {
      "Id": "36a173f7-d026-4d76-8458-321e7f2eff47",
      "Name": " PhoneService_1",
      "Message": null
    },
    {
      "Id": "0b739e18-22a2-eb11-84f2-54bf64846080",
      "Name": " PhoneService_2",
      "Message": null
    }
  ],
  "Failures": []
}

```

***Phone Service Dialer Actions Object***

<b>Name</b> (required)	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites, i.e. The action will be

	available to all Sites configured in Revolution.
<b>Extension</b>	Dial an extension directly
<b>URL</b>	This is not a create or update property but will be provided on all GET of an actions. This is a dynamically generated callback URL to trigger this action.

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Polycom/PhoneServiceDialer/
```

*Follow the same example as provided for PhoneService*

### ***Polycom Dial Monitor Actions Object***

<b>Name (required)</b>	The name of action you are creating.
<b>Site</b>	Provide a Site (name) to associate the action.  A null or empty string for Site defaults the action to AllSites, i.e. The action will be available to all Sites configured in Revolution.
<b>MonitoredNumber (required)</b>	Enter specific number to monitor or use ? and * wildcards (?11 matches 911 and 411; 2* matches any number dialed beginning with 2)

Actions may be added/updated in collections of Actions objects.

```
POST: http://<ServerIP>/unifiedgui/webapi/Polycom/DialMonitor/
```

*Follow the same example as provided for PhoneService*

## **Delete Polycom Actions**

Delete Polycom Actions by sending a collection of Ids of actions that need to be deleted.

```
POST: http://<ServerIP>/unifiedgui/webapi/Polycom/DeleteActions/
```

### **Example Request**

- This POST deletes actions with below IDs.

```
{ "ids" : ["68782574-051a-4000-b585-a056edbbf2fa", "7sdf42574-051a-4000-b585-a056edbbf2fa" ] }
```

# Triggers

Get a list of module triggers.

## Resource URL

```
GET: https://<ServerIP>/unifiedgui/webapi/Triggers
```

### Example Request with Filter

```
GET: https://<ServerIP>/unifiedgui/webapi/Triggers?$filter=ModuleName eq 'SIPActivator'
```

### Example Response

```
{
"@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#Triggers",
"value": [
{
"Id": "bd19e208-983a-4409-bbd4-35081f7e9162",
"Name": "sipaction",
"ModuleName": "SIPActivator",
"Site" : "PDX"
},
{
"Id": "827fbee7-ec74-4b23-a32e-0f3d31c66b16",
"Name": "sipaction2",
"ModuleName": "SIPActivator",
"Site" : "PDX"
}
]
}
```

## Delete a Trigger

```
DELETE: http://<ServerIP>/unifiedgui/webapi/Triggers(76ad0e0a-1a4f-  
e811-8c33-f48e3896e1a5)
```

If the delete is successful, a 204 No Content response is returned.

## Sites

Get a list of module sites.

## Resource URL

```
GET: https://<ServerIP>/unifiedgui/webapi/Sites
```

### Example Request with Filter

```
GET: https://<ServerIP>/unifiedgui/webapi/Sites?$filter=Name eq 'PDX'
```

### Example Response

```
[  
{"id": 1, "name": "Default"},  
{"id": 2, "name": "Company PDX"}  
]
```

## Contacts

Get a list of module sites.

## Resource URL

```
GET: http://<ServerIP>/unifiedgui/webapi/Contacts
```

### Example Request with Filter

```
GET: http://<ServerIP>/unifiedgui/webapi/Contacts?$filter=LastName eq  
'Robinson'
```

### Example Response

```
{
  "Id": "6f75556a-07c2-ea11-84cb-54bf64846080",
  "LastName": " Robinson",
  "FirstName": "Derek",
  "JobTitle": "developer",
  "Department": "Eng Dept.",
  "CompanyOrOrganization": "CattleProd USA, Inc.",
  "Address":{
    "AddressLine1": "5050 N 22nd Ave",
    "AddressLine2": "",
    "City": "LO",
    "StateOrProvince": "Arizona",
    "Country": "USA",
    "PostalCode": "85016",
    "Note": ""
  },
  "AssociatedEndpoints":[
    "32d88295-4cc8-e911-80ea-0050568f18f4",
    "34d88295-4cc8-e911-80ea-0050568f18f4"
  ],
  "EmailAddresses":[{"
    "Email": "derek@webapi.com",
    "Label": "Work",
    "PreferenceOrder": 0
  }],
  "Phones":[{"Number": "+16235940000", "Label": "Work", "CanReceiveSMS":
false, "PreferenceOrder": 0...}],
  "AllowMobile": true,
  "AllowSelfManagement": true,
  "Site": "Lincoln",
  "ProviderBy": "Web Api",
  "Category": "webapi Sales",
  "ContactState": "Active"
}
```



## Create a Contact

POST: <http://<ServerIP>/unifiedgui/webapi/Contacts>

### Example Request

```
{
  "LastName": "Doe",
  "FirstName": "John",
  "JobTitle": "Developer",
  "Department": "Eng Dept.",
  "CompanyOrOrganization": "CattleProd USA, Inc.",
  "Address": {
    "AddressLine1": "5050 N 22nd Ave",
    "City": "LO",
    "StateOrProvince": "Arizona",
    "Country": "USA",
    "PostalCode": "85016"
  },
  "AssociatedEndpoints": ["@Cisco:SEP58BC27757880", "@Cisco:SEP2834A283EE79"],
  "EmailAddresses": [
    {
      "Email": "smith@webapi.com",
      "Label": "Work",
      "PreferenceOrder": 0
    }
  ],
  "Phones": [
    {
      "Number": "+16235940000",
      "Label": "Work",
      "CanReceiveSMS": false,
      "PreferenceOrder": 0
    }
  ],
}
```

```
"AllowMobile": true,  
"AllowSelfManagement": true,  
"Site": "Lincoln",  
"Category": "webapi Sales",  
"ContactState": "Active"  
}
```

**Example Response**

HttpStatus - 201 Created. The API also returns the Contact object that was created along with Id. If there was an error during Contact creation, then a HttpStatus - 500 Internal Server Error along with exception message is returned.

## Update a Contact

```
PUT: http://<ServerIP>/unifiedgui/webapi/Contacts(6f75556a-07c2-ea11-84cb-54bf64846080)
```

OR

```
PATCH: http://<ServerIP>/unifiedgui/webapi/Contacts(6f75556a-07c2-ea11-84cb-54bf64846080)
```

Either PUT or PATCH can be used to update existing Contact record. A successful update will return a HttpStatus - 200 OK and no return content

## Delete a Contact

```
DELETE: http://<ServerIP>/unifiedgui/webapi/Contacts(6f75556a-07c2-ea11-84cb-54bf64846080)
```

A successful delete will return a HttpStatus 204 No Content. If the contact was not found in the system, the API will return HttpStatus 404 Not Found

## UserTags

Get a list of tags.

```
GET: http://<ServerIP>/unifiedgui/webapi/UserTags
```

#### Example Response

```
{
"@odata.context": "http://localhost/unifiedgui/webapi/$metadata#Tags",
"value": [{
  "Id": "932b76d5-3ba6-e811-8c36-f48e3896e1a5",
  "Name": "Test",
  "TagType": "Standard",
  "Site" : "PDX"
},
{
  "Id": "932b76d5-3ba6-e811-8c36-f48e3896e345",
  "Name": "IPDevices",
  "TagType": "Standard",
  "Site" : "PDX"
},
{
  "Id": "932b76d5-3xsa6-e811-8c36-f48e3896e1a5",
  "Name": "DynamicTag",
  "TagType": "DynamicEndpoint",
  "Site" : "PDX"
}
}]
```

## Create a Standard UserTag

### Standard UserTag Object

<b>Name</b> (required)	A unique name for the UserTag. Name can have maximum 64 characters, \$&#?() and single-quote are excluded.
<b>Endpoints</b>	List of URNs (strings).

<b>SubTags</b>	A list of IDs (Guid) of other usertags that could be nested under this usertag.
<b>Site</b>	Provide a Site (name) to associate the action. A null or empty string for Site defaults the action to AllSites.

### Example

```
POST: http://<ServerIP>/unifiedgui/webapi/UserTags/StandardUserTag/
```

### Example Request

```
{ "usertag" : {
  "Name": "ST_5",
  "Endpoints" : ["@AtlasSound:3222c18106b9:1"],
  "SubTags" : ["932b76d5-3ba6-e811-8c36-f48e3896e1a5"],
  "Site": "Default"
}
```

An update to an existing usertag also requires a POST to the same endpoint, but the payload requires an Id.

```
POST: http://<ServerIP>/unifiedgui/webapi/UserTags/StandardUserTag/
```

### Example: Update a Usertag Tag

```
{ "usertag": {
  "Id": "932b76d5-3xsa6-e811-8c36-f48e3896e1a5",
  "Name": "ST_5",
  "TagType": "Standard",
  "Site": "Default",
  "Endpoints": ["@AtlasSound:0002c18106b9:1"],
  "Contacts": [],
  "SubTags": [160]
}}
```

## Create a Dynamic UserTag

Dynamic Tags resolve to a set of endpoints or contacts at the time of execution that satisfy the conditions specified. The API allows creation of two different dynamic tags.

- Endpoint dynamic tag
- Contact dynamic tag

<b>Name</b> (required)	A unique name for the UserTag.
<b>Site</b>	Provide a Site (name) to associate the action. <i>A null or empty string for Site defaults the action to AllSites.</i>

## TagQuery Objects

These are used to define the dynamic conditions.

<b>Key</b> (required)	<p>List of Endpoint keys:</p> <p>"Category",</p> <p>"MAC",</p> <p>"Manufacturer",</p> <p>"Model",</p> <p>"IP Address",</p> <p>"MitelSite",</p> <p>"MitelUserGroup",</p> <p>"Cisco Calling Search Space",</p> <p>"Cisco Device Pool",</p> <p>"Cisco Partition",</p> <p>"Custom Name"</p> <p>"Destination Code"</p> <p>List of Contact keys, do GET:</p> <p>"Department",</p> <p>"CompanyOrOrganization",</p> <p>"JobTitle",</p>
-----------------------	--

	"DisplayName", "Category"
<b>Pattern</b> (required)	Any string to match against a key value such as "192.168" on key "IP Address".
<b>Comparator</b> (required)	Here is a list of comparators that could be used in a tag query: "Equal", "Not Equal", "Starts With", "Ends With", "Contains", "Does Not Contain"

### Example

```
POST: http://<serverIP>/unifiedgui/webapi/UserTags/DynamicEndpointUserTag
```

### Example Request

```
{ "usertag" : {
  "Name": "Endpoint_Created_2",
  "TagQueries": [{ "Key": "Category", "Pattern" : "Phone", "Comparator": "Contains" }],
  "Site": "Default"
}}
```

### Example

```
POST: http://<ServerIP>/unifiedgui/webapi/UserTags/DynamicContactUserTag
```

### Example Request

```
{ "usertag" : {
  "Name": "Contact_Created_2",
  "TagQueries": [{ "Key": " JobTitle", "Pattern" : "Engineer", "Comparator": "Contains" }],
  "Site": "Default"
}}
```

### Example Response

```
{
```

```
"Id": "932b76d5-3ba6-e811-8c36-f48e3896e1a5",
"Name": "Endpoint_Created_2",
"TagType": "DynamicEndpoint",
"Site": "Default",
"Endpoints": [],
"Contacts": [],
"SubTags": [
"8240ae67-3ca6-e811-8c36-f48e3896e1a5"
]}
```

## Updates to DynamicTags

```
POST: http://<serverIP>/unifiedgui/webapi/UserTags/DynamicEndpointUserTag
```

Use the same payload structure as creating a new tag, but also include 'ID' of the tag you want to update including the updates you want to make.

This is an example of how to add more tag patterns to an existing usertag, as well as make other changes.

- Name was modified
- Another tag search pattern was added

```
{
  "usertag" :{
    "Id": "932b76d5-3ba6-e811-8c36-f48e3896e1a5",
    "Name" : "Dy_test_1_chg",
    "TagQueries" : [{ "Key" : "Category", "Pattern" : "Phone", "Com-
parator" : "Contains"},
                    { "Key" : "IP Address", "Pattern" : "192", "Com-
parator" : "Starts With"}],
    "Site" : "Default"
  }
}
```

## Delete UserTag

To delete a usertag, issue a DELETE request with the ID of the tag.

**Example:**

```
DELETE: http://<ServerIP>/unifiedgui/webapi/UserTags(932b76d5-3ba6-  
e811-8c36-f48e3896e1a5)
```

Response: Http Status Code 204

## Endpoints

### Resource Information

<b>Name</b>	String
<b>Urn</b>	String
<b>Description</b>	String
<b>IPAddress</b>	String
<b>Port</b>	Int
<b>ModuleName</b>	String. Name of the module the endpoint belongs to.
<b>CustomName</b>	String. User editable.
<b>IsLicensed</b>	Boolean
<b>StatusLastChanged</b>	Datetime
<b>Staus</b>	String
<b>Type</b>	String. Could be endpoint, gateway, rebroadcaster, static stream.
<b>EndpointSettings</b>	Collection of key value pair. User editable. Allowed Keys are "TextLoop", "TextSpeed", "DefaultVolume", "MaxVolume", and "MinVolume" ONLY on IPDevice endpoints.
<b>DestinationCodes</b>	Collection of key value pairs. User editable. This property value is returned only on endpoints that have user assigned destination codes



<b>SystemDestinationCodes</b>	Collection of key value pairs. These are system assigned destination codes and cannot be edited by user.
<b>Metadata</b>	Read only collection of key value pairs. This property value is returned only on specific endpoint query.
<b>Site</b>	String
<b>Latitude</b>	Double. User editable.
<b>Longitude</b>	Double. User editable.
<b>Elevation</b>	Double. User editable.
<b>Location</b>	String. User editable.

## Get Endpoints

Get a list of all endpoints.

```
GET: http://<ServerIP>/unifiedgui/webapi/Endpoints/
```

### Example Response

```
{
  "value": [
    { "Name": "v3 Paging Relay - 192.168.203.19 - LineOut", "Description":
      "v3 Paging Relay - 192.168.203.19 - 00:20:f7:02:4c:a4",...},
    { "Name": "v3 Paging Relay - 192.168.203.19 - Relay", "Description":
      "v3 Paging Relay - 192.168.203.19 - 00:20:f7:02:4c:a4",...},
    { "Name": "v3 Paging Relay - 192.168.203.19 - Rebroadcaster", "Description":
      "v3 Paging Relay - 192.168.203.19 - 00:20:f7:02:4c:a4",...}
    ... ]
}
```

### To get a specific endpoint:

```
GET: http://<ServerIP>/unifiedgui/webapi/Endpoints
('@IPDevice:2046f901012d:1')
```

### Example response

```
{
  "Name": "Adaptive Micro Systems LLC - 192.168.10.128",
  "Description": "Adaptive Micro Systems LLC - 192.168.10.128 -
98:5D:AD:CA:CD:5B",
  "Urn": "@IPDevice:985DADCACD5B",
  "IPAddress": "192.168.10.128",
  ...
}
```

## Update Settings in an Endpoint

Use PATCH: to update endpoint settings. Not all settings are available for update.

Only the following fields can be updated:

- Latitude
- Longitude
- Elevation
- Location
- EndpointSettings - Valid keys allowed to be set are "TextLoop", "TextSpeed", "DefaultVolume", "MaxVolume", and "MinVolume"
- *[Note: User should be aware of the endpoint's capabilities before setting these settings. i.e. certain IPDevices support only audio, therefore TextLoop setting is not appropriate for the endpoint.]*
- CustomName
- DestinationCodes

```
PATCH: http://<ServerIP>/unifiedgui/webapi/Endpoints
('@IPDevice:2046f9030b60:1')
```

### Example Request

```
{
  "CustomName": "AND Updated",
  "Latitude": "41.51",
```

```
"Longitude": "80.55",
"DestinationCodes": [
  "3200"
],
"EndpointSettings": [
  {
    "Key": "DefaultVolume",
    "Value": "50"
  },
  {
    "Key": "TextLoop",
    "Value": "0"
  }
]
```

If the request above is successful, the call returns HttpStatus 200 OK.

If there were validation errors, the call returns HttpStatus 400 Bad Request, with a meaningful error message.

## Delete an Endpoint

```
DELETE: http://<ServerIP>/unifiedgui/webapi/Endpoints('@Cis-
co:SEP2834A283B753')
```

If the delete is successful, a 204 No Content response is returned.

## Directory

### Get All Directory Entries

```
GET: http://<ServerIP>/unifiedgui/webapi/DirectoryEntry
```

#### Example Response

```

{
  "@odata.context": "https://-
localhost/unifiedgui/webapi/$metadata#DirectoryEntry",
  "value": [
    {
      "Id": "63a12568-f976-eb11-80f8-0050568f18f2",
      "Name": "Hot Desk 1",
      "Code": "1704",
      "Location": "",
      "Latitude": 45.51823011705024,
      "Longitude": -122.6839846153478,
      "Elevation": null
    },
    {
      "Id": "64a81568-f976-eb11-80f8-0050568f18f4",
      "Name": "James Smith",
      "Code": "4136",
      "Location": "Chicago, IL office",
      "Latitude": null,
      "Longitude": null,
      "Elevation": 135
    },
    .....
  ]
}

```

## Delete Directory Entries

POST: <http://<ServerIP>/unifiedgui/webapi/DirectoryEntry/DeleteEntries>

### Example Request

This POST deletes directory entries with the below IDs.

```

{ "ids" : ["50075344-be35-ec11-810a-0050568f18f4", "a15cb8fa-8a1c-ec11-
8108-0050568f18f4"] }

```

# Add/Update Directory Entries

```
POST: http://<ServerIP>/unifiedgui/webapi/DirectoryEntry/AddUpdate
```

Bulk update endpoints and directory entries or add new directory entries.

Request body fields:

- Match
  - Set this to the name of the column on which to match existing values:
  - “DestinationCode”
  - “IpAddress” (matches endpoints only)
  - “MAC” (matches endpoints only)
  - “URN” (matches endpoints only)
- UpdateEndpoints (default false)
  - Set to true to update any matching endpoints
- UpdateDirectory (default false)
  - Set to true to update any matching directory entries
- AddDirectory (default false)
  - Set to true to create new directory entries if no matches are found
- Data
  - A list of values, each representing an endpoint/directory entry to update or create
  - The following fields can be included:
    - Name (Custom Name)
    - Destination Code
    - Location
    - Latitude
    - Longitude

- Elevation
- IpAddress (affects endpoints only)
- MAC (affects endpoints only)
- URN (affects endpoints only)
- Site (affects endpoints only)

#### Example Request

```
{
  "Match": "DestinationCode",
  "UpdateEndpoints": true,
  "UpdateDirectory": true,
  "AddDirectory": true,
  "Data": [
    {
      "Name": "John Doe",
      "DestinationCode": "1234",
      "Location": "Portland, OR office",
      "Elevation": "150"
    },
    {
      "DestinationCode": "400*1",
      "Location": "Building A Floor 5",
      "IpAddress": "192.168.10.50"
    },
    {
      "Name": "Hot Desk 3",
      "DestinationCode": "7",
      "Latitude": "45.51823011705024",
      "Longitude": "-122.6839846153478"
    }
  ]
}
```

# Media

## Resource Information

<b>Name</b>	
<b>Id</b>	Guid
<b>MimeType</b>	string
<b>ResourceType</b>	String value. Audio, image, or both.
<b>Data</b>	Byte array
<b>Attributes</b>	Key and value pairs of additional information on the media.

## Get All Media Files

```
GET: http://<ServerIP>/unifiedgui/webapi/MediaResources/
```

or with query

```
GET: http://<ServerIP>/unifiedgui/webapi/MediaResources/?$top=5
```

or by specific media by guid

```
GET: http://<ServerIP>/unifiedgui/webapi/MediaResources(900edc47-ee89-e811-8c33-f48e3896e1a5)
```

or

```
GET: http://<ServerIP>/unifiedgui/webapi/MediaResources/?$filter=Id eq 050f2c82-c939-e811-80dc-0050568f18f4
```

### Example response

```
{
  "value": [
    {
      "Id": "01410b38-2a89-e811-8c33-f48e3896e1a5",
      "Name": "Fight_Bell.mp3",
```

```
"MimeType": "audio/mpeg",
"ResourceType": "Audio",
"Data": "",
"Attributes":[{"Key": "duration", "Value": "8s" }]
},
{"Id": "01bc0f32-2a89-e811-8c33-f48e3896e1a5", "Name": "export.svg",
"MimeType": "image/svg+xml",...},
{"Id": "01bd0f32-2a89-e811-8c33-f48e3896e1a5", "Name": "sun.svg",
"MimeType": "image/svg+xml",...},
{"Id": "024873a8-3389-e811-8c33-f48e3896e1a5", "Name": "Closing_
Bell.mp3", "MimeType": "audio/mp3",...},
{"Id": "04410b38-2a89-e811-8c33-f48e3896e1a5", "Name": "Fight_
Bell.wav", "MimeType": "audio/x-wav",...}
]
}
```

*[Note: If a user attempts to upload the same media file twice, the second attempt returns the id (Guid) of the original (first) media file as the second attempt. This helps prevent uploading same media files with different names.]*

## Upload Media

```
POST: http://<ServerIP>/unifiedgui/webapi/MediaResources/UploadMedia
```

Content-Type: multipart/form-data

It is important to include the content\_type “multipart/form-data” to decipher the binary of the media file.

The call returns a Guid value, which is the Id of the media that was successfully uploaded.

### Example Response

```
{
"value": "1e62befb-ef89-e811-8c33-f48e3896e1a5"
}
```



## Download Media

```
POST: http://<ServerIP>/unifiedgui/webapi/MediaResources/DownloadMedia
```

### Example Response

```
{  
  "id" : "f792eda8-4489-e811-8c33-f48e3896e1a5"  
}
```

## Delete Media

```
DELETE: http://<ServerIP>/unifiedgui/webapi/MediaResources (f792eda8-  
4489-e811-8c33-f48e3896e1a5)
```

A successful delete returns a 404 No Content message. If there were errors during delete, please refer to Revolution main log for details.

## Notifications

### Resource Information

<b>Name</b> (required)	
<b>NotificationType</b> (required)	Type of notification: "Stored_Audio", "Text_And_Images", "Text_To_Speech", "One_Way", "Recorded", "Two_Way", "Direct_One_Way"
<b>Priority</b> (optional)	Defaults to 5 if not set.
<b>Volume</b> (optional)	Defaults to 7 if not set.
<b>ActivationType</b> (optional)	Defaults to Iteration.
<b>OpeningTone</b> (optional)	Id of audio file from the media library.

<b>ClosingTone</b> (optional)	Id of audio file from the media library.
<b>StoredAudio</b> (optional)	List of audio file ids from the media library.
<b>StoredImages</b> (optional)	List of image ids from the media library.
<b>Triggers</b> (optional)	List of notification trigger Ids (Guid).
<b>Interval</b> (optional)	Used for repeated or continuous notifications. Defaults to 60 seconds.
<b>Iterations</b> (optional)	Defaults to 1.
<b>ClearDisplay</b> (optional)	Defaults to false.
<b>Title</b> (required)	Notification title.
<b>Body</b> (required)	Notification body.
<b>Endpoints</b> (optional)	List of endpoint urns the notification will be sent to.
<b>UserTags</b> (optional)	List of user tag ids (GUID) the notification will be sent to.
<b>Site</b> (required)	Provide a Site (name) to associate the action. An empty string defaults the action to AllSites.

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/
```

or

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/?$top=10
```

or to query a specific notification

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications(02934c71-e5fe-47d3-95f5-7bf7fec54fc7)/
```

#### Example Response

```
{
```

```

"@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#Notifications",
"value": [
  {
    "Id": "02934c71-e5fe-47d3-95f5-7bf7fec54fc7",
    "Name": "Test Alert_84",
    "NotificationType": "Stored_Audio",
    "ActivationType": "Iteration",
    "Volume": 3,
    "Priority": 5,
    "Interval": 60,
    "Iterations": 1,
    "FontColor": "",
    "ClearDisplayAfterNotification": false,
    "Icon": "85c73929-e98b-47b9-ab90-5b8dada690d4",
    "OpeningTone": "00000000-0000-0000-0000-000000000000",
    "ClosingTone": "00000000-0000-0000-0000-000000000000",
    "StoredAudio": ["b1def37d-d162-4fae-90bb-023d52344685"],
    "StoredImages": ["85c73929-e98b-47b9-ab90-5b8dada690d4"],
    "Triggers": [],
    "TextTitle": "Test Alert Script",
    "TextBody": "JOY JOY JOY",
    "Endpoints": ["@AtlasSound:0002c18106b9:1"],
    "UserTags": [],
    "GeoFencingBoundary": "All",
    "GPORelayMode": "On_While_Active",
    "ContactMethods": [],
    "Site": "Company PDX"
  },
  ...
]}

```

## Add a New Notification

POST: <http://<ServerIP>/unifiedgui/webapi/Notifications>

## Example Response

```
{
  "Name": "API-Text",
  "NotificationType": "Stored_Audio",
  "ActivationType": "Iteration",
  "Volume": 6,
  "Priority": 5,
  "Interval": 30,
  "Iterations": 2,
  "FontColor": "#B63939",
  "ClearDisplayAfterNotification": true,
  "Icon": "00000000-0000-0000-0000-000000000000",
  "OpeningTone": "00000000-0000-0000-0000-000000000000",
  "ClosingTone": "00000000-0000-0000-0000-000000000000",
  "StoredAudio": ["51143622-0a8f-4e2b-a0ab-4c703085b136"],
  "StoredImages": ["3b8a61cc-280a-4427-bfa6-9a6c3b8d7f78"],
  "Triggers": ["bd19e208-983a-4409-bbd4-35081f7e9162"],
  "TextTitle": "Title",
  "TextBody": "Text body",
  "Endpoints":["@Stream:88-3eWtb_kyMJs13_tS2g-
g", "@AtlasSound:0002c18106b9:1"],
  "UserTags": ["sd33e208-983a-4409-ddw4-35dfeef7e9162"],
  "Site": "",
  "GeoFencingBoundary": "All",
  "GPORelayMode": "On_While_Active",
  "ContactMethods": []
}
```

## Update an Existing Notification

```
PUT http://localhost<ServerIP>/unifiedgui/webapi/Notifications
(02934c71-e5fe-47d3-95f5-7bf7fec54fc7) /
```

*PUT - Must provide a full payload of the existing Notification object with changes to properties as needed*

```
PATCH: http://localhost<ServerIP>/unifiedgui/webapi/Notifications  
(02934c71-e5fe-47d3-95f5-7bf7fec54fc7) /
```

*PATCH - can accept a valid JSON of a subset of Notification object. E.g. Name-Values pairs of just the properties that need to be modified.*

#### Example Request

```
{  
  "Id": "02934c71-e5fe-47d3-95f5-7bf7fec54fc7",  
  "Name": "Test Alert_84",  
  "Volume": 5,  
  "Priority": 5,  
  "Site": "Company PDX"  
}
```

## Activate a Notification

Notification can be activated using two different REST methods - GET and POST.

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/ActivateNotifica-  
tion(notificationId=<notificationID>,token='<web api user token>')
```

#### Optional Querystring Parameters for GET Activations

The following optional parameters can be included in addition to the GET method of activating a notification to override/append to the properties already configured in the notification.

- overridetitle
- overridebody
- appendwithaudio
- appendwithimage
- prioritygroupid
- location
- elevation

- latitude (required if longitude provided)
- longitude (required if latitude provided)

#### Example Request with optional parameters

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/ActivateNotification(notificationId=<notificationID>,token='<web api user token>')?overrideTitle=Title is overridden&overridebody=this is a new body text&appendwithimage=a128ac04-9e05-ea11-8c55-f48e3896e1a5 &prioritygroupid=a443443a-4644-ea11-80ed-0050568f18f4&location='Portland OR'&elevation=125
```

All parameters or a subset of them may be added to the activation request. Incorrect optional parameter (e.g. typos) will result in an error response of HttpStatus 400 Bad Request.

On a successful activation the API returns HttpStatus 200 OK with activation event response object.

#### Example Response

```
Id": "45012f59-fc5d-e811-80dd-0050568f18f4",
"NotificationInstanceIDs": [
"1eaa99f9-d5ff-473a-b8e7-633619ef23ec"
],
>Status": "Activated"
```

Prioritygroupid parameter applies only in a redundancy environment. It takes a guid value of a prioritygroup as parameter. This value is used to determine if the notification should be executed on a server based on the priority group ranking already setup in Revolution. To retrieve all priority groups set up in the system please refer to the Priority Groups sections of the document. If a server determines that it should not execute the notification on the basis of prioritygroupid, then it will return an error code 403 with a friendly error message. On the other hand, if priority group matches with the server, then the notification will be activated an activation event object will be returned similar to the example above.

#### Activate notification using POST and parameter overrides.

```
POST: http://<ServerIP>/unifiedgui/webapi/Notifications/Activate
```

### Example Request

```
{  "activate" : {  "Id": "4c77dd9c-417c-4ae4-a3c1-91a692efb11e"  } }
```

### Resource Information

<b>Id</b> (required)	Guid of the notification.
<b>OverrideTitle</b> (optional)	Defaults to text in the notification.
<b>OverrideBody</b> (optional)	Defaults to text body in the notification.
<b>AppendWithImage</b> (optional)	Guid of one image that is appended to existing image (if any) in the notification.
<b>AppendWithAudio</b> (optional)	Guid of one audio that is appended to existing stored audio (if any) in the notification.
<b>PriorityGroupId</b> (optional)	Guid of priority group that will be considered in determining if this notification should be executed or not in a redundancy environment.
<b>Location</b> (optional)	Description of the event location
<b>Elevation</b> (optional)	Elevation of the event
<b>Latitude</b> (optional)	Latitude of event. Required if Longitude is provided.
<b>Longitude</b> (optional)	Longitude of event. Required if Latitude is provided.

On a successful activation, a 200 Http OK response with notification instance details is returned to the user.

### Example Response

```
{  
  "Id": "d696d4dd-2a4f-e811-8c33-f48e3896e1a5",  
  "NotificationInstanceIDs": [ "d05e0de8-8da3-4331-a2fa-cc96adaef9ba" ],  
  "Status": "Activated"  
}
```

### Stop an activate notification using POST

```
http://<ServerIP>/unifiedgui/webapi/Notifications/StopActivation/
```

```
{
  "notificationInstanceId": "16f525c7-12e3-48a2-8dab-b4cd4801ed83 "
}
```

<b>notificationInstanceId</b> (required)	Guid of the NotificationInstanceIDs that is returned in the response body of the Activate call.
---	---

A response of 200 OK is returned if the activation was successfully cancelled. An invalid notificationInstanceId in the POST call will result in a 400 BadRequest error. Any other issues with regards to cancelling the activation will return a 500 InternalServerError.

## Delete a Notification

```
DELETE: http://<ServerIP>/unifiedgui/webapi/Notifications(4c77dd9c-417c-4ae4-a3c1-91a692efb11e)
```

If the delete is successful, a 204 No Content response is returned.

### *Get Notification Results*

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adaef9ba) /
```

#### Example Request

```
{
  "@odata.context": "http://192.168.11.64/unifiedgui/webapi/$metadata#NotificationResult/$entity",
  "NotificationID": "45012f59-fc5d-e811-80dd-0050568f18f4",
  "NotificationInstanceId": "c4d2bbcf-85a5-4591-8399-d6da1b83c7b5",
  "NotificationName": "_smithIPDevice",
  "NotificationType": "Stored_Audio",
  "ActivationType": "Iteration",
  "CallerID": "M Subrama",
  "Title": "Test",
  "Body": "This is a short text message",
}
```



```
"TimeStarted": "2018-11-27T19:51:29.8271065Z",
"TimeFinished": "2018-11-27T19:51:32.8962809Z",
"AllSites": true,
"Audio": [{"Guid": "0ce5fd99-a4f1-e811-80e3-0050568f40f6", "Name": "ana-
log1.mp3", "MimeType": "audio/mpeg"...},
"Images": [{"Guid": "169f93c0-7bf2-e811-80e3-0050568f40f6", "Name":
"bell.png", "MimeType": "image/png"...},
"IsComplete": false,
"Status": "Finished",
"StatusText": "",
"SentToTotals": {"Total": 1, "Errors": 0, "Warnings": 0, "Success": 1,
"Busy": 0...},
"AllTotals": {
"Total": 1,
"Errors": 0,
"Warnings": 0,
"Success": 1,
"Busy": 0
},
"EndpointTotals": {"Total": 1, "Errors": 0, "Warnings": 0, "Success": 1,
"Busy": 0...},
"EndpointsThatCanRespond": 0,
"ContactTotals": {"Total": 0, "Errors": 0, "Warnings": 0, "Success": 0,
"Busy": 0...},
"GatewayTotals": {"Total": 0, "Errors": 0, "Warnings": 0, "Success": 0,
"Busy": 0...},
"RebroadcasterTotals": {"Total": 0, "Errors": 0, "Warnings": 0, "Suc-
cess": 0, "Busy": 0...},
"StaticStreamTotals": {"Total": 0, "Errors": 0, "Warnings": 0, "Suc-
cess": 0, "Busy": 0...},
"ResponseTotals": []
}
```

*[Note: NotificationResults returns Title, Body, and a list of Audio and Image used at the time of the notification. Audios and images are available for download using this DownloadMedia API call:*

```
POST: http://<ServerIP>/unifiedgui/webapi/MediaResources/DownloadMedia
```

with JSON body containing the Guid. Refer to the [MediaResource](#) section for more information and examples.]

## Understanding NotificationResults Response

Status - This indicates the status of the notification after it completes. Refer to the [IPDevice](#) for a list of possible Status values.

### Get all endpoints for a specific notification that was activated

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationEndpointResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adae-f9ba)/
```

Collection of endpoints a notification was sent to is returned as a collection.

### Example Response

```
{ "value": [
  {
    "URN": "@IPDevice:2046f9030b60:1",
    "IPAddress": "192.168.11.100",
    "Status": "Success",
    "Message": "",
    "Description": "Advanced Network Devices - 192.168.11.100",
    "Response": null
  },
  {
    "URN": "@IPDevice:2046f9030b90:1",
    "IPAddress": "192.168.11.101",
    "Status": "Success",
    "Message": "",
    "Description": "Advanced Network Devices - 192.168.11.101",
    "Response": null
  }
] }
```

### Get all contacts for a specific notification that was activated

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationContactResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adaef9ba) /
```

#### Example Response

```
{
  "value": [{
    "ContactID": "2ac50c83-c939-e811-80dc-0050568f18f4",
    "AddressEntryName": "OnePlus3 - Success|Chromebook_596D - Success|OnePlus 3 - Success|name@domain.com - Success|",
    "Status": "Success",
    "Message": "",
    "Description": "John Smith",
    "Response": null
  }]
}
```

### Get all Rebroadcasters for a specific notification that was activated

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationRebroadcasterResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adaef9ba) /
```

### Get all Gateways for a specific notification that was activated

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationGatewayResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adaef9ba) /
```

### Get all StaticStream for a specific notification that was activated

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationStaticStreamResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adaef9ba) /
```

### Get all StaticStream RTCP for a specific notification that was activated

```
GET: http://<ServerIP>/unifiedgui/webapi/Notifications/NotificationStaticStreamRTCPResult(notificationInstanceId=d05e0de8-8da3-4331-a2fa-cc96adaef9ba, urn='@Stream:g7NxJLhQEeiA4ABQVo8Y9A') /
```

# Priority Group

## Resource Information

<b>Name</b>	Friendly name of the priority group
<b>Guid</b>	Unique identification of the group
<b>Members</b>	Collection of servers, respective ranks and member group id

## Resource URL

Retrieves a collection of priority groups setup in the system.

```
GET: http://<ServerIP>/unifiedgui/webapi/PriorityGroups
```

### Example response

```
{
  "Guid": "31578640-2df1-e911-80ec-0050568f18f4",
  "Name": "DEFAULT",
  "Members": [
    {
      "Rank": 0,
      "Server": {
        "Guid": "30578640-2df1-e911-80ec-0050568f18f4",
        "Hostname": "dev",
        "Id": 0,
        "FriendlyName": "dev"
      },
    },
    {
      "Guid": "32578640-2df1-e911-80ec-0050568f18f4"
    }
  ]
}
```

## IPDevice

Get a list of IP devices.

## Resource URL

```
GET: https://<ServerIP>/unifiedgui/webapi/IPDevices
```

### Example Request with Filter

```
GET: https://<ServerIP>/unifiedgui/webapi/IPDevices?$filter=Manufacturer eq 'Company'
```

### Example Response

```
{
  "value": [
    {
      "Id": "5675xxx-41c8-e451-8044-0050343418f4",
      "MAC": "00:02:f1:8a:07:x4",
      "IPAddress": "192.168.11.55",
      "Type": "IPDevice",
      "Manufacturer": "AtlasIED",
      "APIVersion": "1.1",
      "Model": "IP-SDMF",
      "Description": "AtlasIED IPX",
      "FirmwareVersion": "1.2.0",
      "IsRegistered": true
    },
    {
      "Id": "fb1a6ree-41c8-e911-85ea-0050458f18f4",
      "MAC": "00:20:f7:00:ab:f9",
      "IPAddress": "192.168.10.20",
      "Type": "IPSpeaker",
      "Manufacturer": "Company",
      "APIVersion": null,
      "Model": "Ceiling Speaker v2",
      "Description": null,
      "FirmwareVersion": "v6.2.2",
    }
  ]
}
```

```
"IsRegistered": true
}, ...
] }
```

#### Example Get IPDevice Request with Id

```
GET: http://<ServerIP>/unifiedgui/webapi/IPDevices(5675aed6-41c8-e911-80ea-0050568f18f4)?
```

```
{
"@odata.context": "http://-
localhost/unifiedgui/webapi/$metadata#IPDevices/$entity",
"@odata.type": "#dotnetmvc.Models.ApiModel.IPDeviceDetail",
"Id": "5675aed6-41c8-e911-80ea-0050568f18f4",
"MAC": "00:02:c1:8a:07:a4",
"IPAddress": "192.168.19.55",
"Type": "IPDevice",
"Manufacturer": "AtlasIED",
"APIVersion": "1.1",
"Model": "IP-SDMF",
>Description": "AtlasIED IPX",
"FirmwareVersion": "1.2.0",
"IsRegistered": true,
"Serial": null,
"CommandPort": 8082,
"Protocol": "TCP",
"AudioInputs": 1,
"AudioOutputs": 1,
"GPInputs": 2,
"GPOutputs": 1,
"HeartbeatDateTime": "2020-08-06T22:26:01.399445Z",
"RegisteredDateTime": "2020-08-06T22:26:01.399445Z",
"SupportsMulticastRelay": false,
"SupportsStoredAudio": false,
"SupportXMLCommands": true,
"SupportText": true,
```

```
"SupportVideo": false,  
"MulticastSubnets": "",  
"SupportImage": false  
}
```

### *Delete an IPDevice*

```
DELETE: http://<ServerIP>/unifiedgui/webapi/IPDevices(4c77dd9c-417c-  
4ae4-a3c1-91a692efb11e)
```

If the delete is successful, a 204 No Content response is returned.

## Appendix

Status	Created, Initialized, Activated, Deactivated, Busy, NotFound, NoEndpoints, InsufficientEndpoints, TooManyEndpoints, Error, Timeout, Finished, Complete, PriorityBusy, Endpoints, Stopping,
--------	---

	NotLicensed, AudioSourceBusy, Cancelled, Preempted, ActivatorAudioRequired, PagingRelayBusy, PagingRelayFault, PagingRelayNotLicensed, AllEndpointsError, ServerIsStandby
--	--