EVOTIX

API Guide

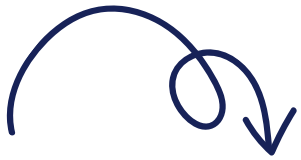
Assure Customer API

User Guide: Manage the Org Structure via API

Evotix Ltd. Revision 2.0 (July 2024)

US +1 (872) 215 5913 **UK** +44 (0) 161 521 8490 **AU** +61 3 8595 5909 **Visit** evotix.com

EVOTIX



The Assure Customer API is the means by which Evotix provides integration 'capability to Assure for customers.

The Customer API is available to customers via the public internet and takes the form of a RESTful API. Using the Customer API you can automate processes such as managing users, org unit structure and exporting data for analysis. Making use of the Customer API requires a level of technical expertise so this is typically something that a company's IT function would handle.

This guide focusses on how to set up the Customer API to manage the organizational structure in Assure. A separate guide is available for setting up the Customer API.

EVOTIX

Contents

EVOTIX

Managing Organisational Units via the Customer API

Another of the interactive methods available via the Customer API is the ability to manage the org unit hierarchy in Assure.

This allows you to create, update and disable an organisational unit.

Limitations

The following lists the current limitations of the API for creating and updating organisational units. It is expected that these will be addressed in future Assure releases:

1. Cannot add any branding to Organisational Units. This includes report images and banner images.

Pre-configuration for existing Organisational Units

Where Organisational Units have been manually created in Assure (i.e. not via the Customer API) there is some pre-configuration required in order to be able then use the Customer API methods to create/update/disable Organisational Units. This is required for the API methods to be able to correctly assign Organisational Units their correct position in the hierarchy (setting Parent Organisational Unit).

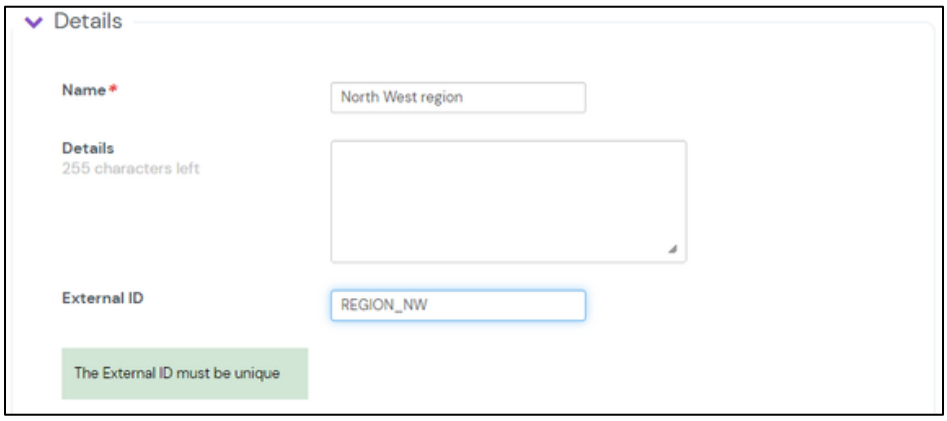
This pre-configuration is only needed for Organisational Units which have been manually added and which do not have an External ID set.

The Assure Organisational Unit hierarchy has a new attribute for each unit called 'External ID'. You will find this in the 'Edit' page of an Organisational Unit and its purpose is to allow an unique external identifier to be associated with each unit. This is necessary because:

1. The existing Organisational Unit names are not unique and therefore cannot be used with an API method to target a specific Organisational Unit.

2. Organisational Unit names can be changed by Assure administrative users so they are not guaranteed to align with the customers IT systems (where user details are being obtained from by the customers integration workflow).

The external IDs for the Organisational Units can be configured manually using the Assure UI (screenshot below shows an example setting the Organisational Unit external ID for the "North West region" to be REGION_NW). This is fine for testing but for ensuring that the Organisational Unit hierarchy is in sync with the customers user management system we have a bulk import/update tool which can be used.



Organisational Unit JSON object

When creating/updating an Organisational Unit the Organisational Unit's details need to be provided in the form of a JSON object. JSON is the most commonly used syntax for describing data objects in RESTful APIs. The OpenAPI schema for the Customer API contains the formal definition of the JSON structure i.e. the orgUnitPOSTRequest object. The OpenAPI schema is the master definition of the API methods and data objects, it should always be consulted to understand the required fields, field types, max data lengths and string patterns, plus descriptions of the behaviour associated with the use of each field (or its omission). Some software tools and platforms can consume the OpenAPI schema to automate the process of generating the correct JSON and if this is available it should be used. To aid understanding of how the JSON fields in the Organisational Unit JSON object relate to the resulting Organisational Unit setup in Assure the following sections contain some worked examples.

The text encoding to be used for all interactions with the Customer API is UTF-8. This is pretty much the standard today for software tools and platforms however it is important to check that you are using UTF-8 as if not then when you get foreign characters in the data or other symbols like emoticons these will not appear correctly in Assure.

As JSON objects are described using plain text it is possible to hand craft these objects for initial testing. However, it is VERY IMPORTANT to ensure that when generating JSON objects to send to the Customer API you use a proper JSON library or a tool with native JSON support. This is because JSON relies on 'escaping' for certain data values, if this escaping is not done correctly it will lead to API errors and can also be a source of security vulnerabilities.

Minimum Organisational Unit data

This example shows the minimum possible Organisational Unit's data which can be used to create/update an Organisational Unit.

```
{
    "externalId": "example.orgExternalId",
    "name": "example.orgName",
    "parentExternalId" : "example.orgParentExternalId"
}
```

Further organisation unit details omitted for brevity as they will all be the defaults.

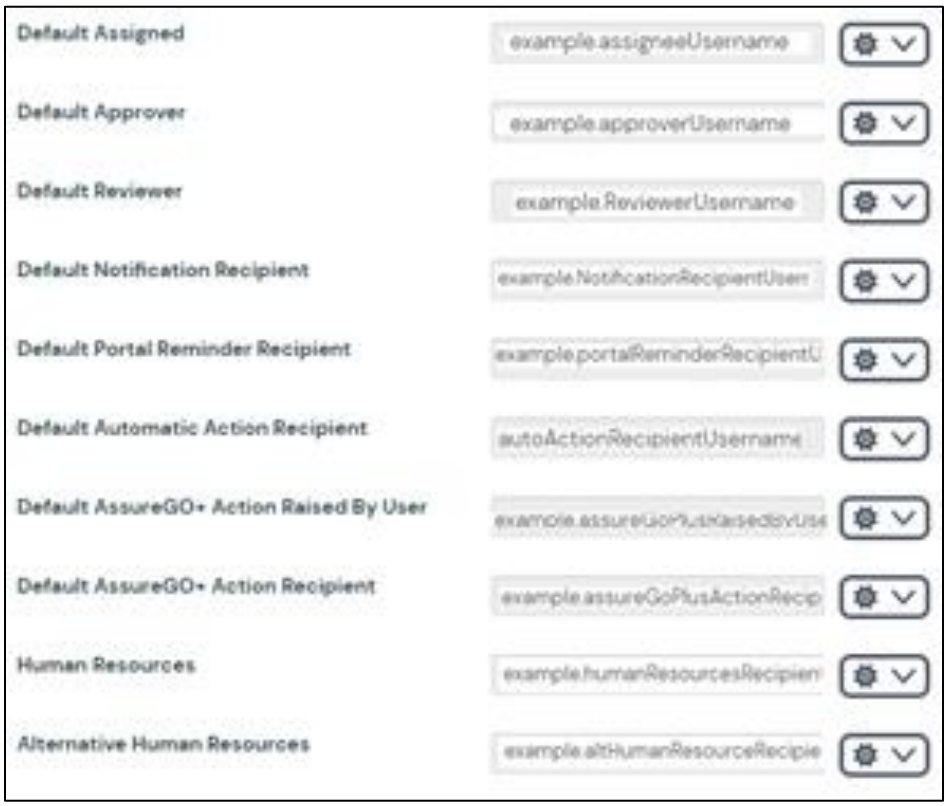
EVOTIX

Organisational Unit with default users

This example shows the same Organisational Unit with default users assigned to it.

```
{
    "externalId": "example.orgExternalId",
    "name": "example.orgName",
    "parentExternalId" : "example.orgParentExternalId",
    "defaultUsers":
    {
      "approverUsername": "example.approverUsername",
      "assigneeUsername" : "example.assigneeUsername",
      "reviewerUsername": "example.reviewerUsername",
      "notificationRecipientUsername" :
"example.notificationRecipientUsername",
      "portalReminderRecipientUsername":
"example.portalReminderRecipientUsername",
      "autoActionRecipientUsername": "example.autoActionRecipientUsername",
      "assureGoPlusActionRaisedByUsername":
"example.assureGoPlusRaisedByUsername",
      "assureGoPlusActionRecipientUsername":
"example.assureGoPlusActionRecipientUsername",
      "humanResourcesRecipientUsername" :
"example.humanResourcesRecipientUsername",
      "alternativeHumanResourcesRecipientUsername" :
"example.altHumanResourceRecipientUsername"
    }
}
```

The Organisational Unit's details will be the same as the 'minimum Organisational Units data' screen shot above. The difference in this case is that the default users are present in the Permissions section.



Organisational Units with Module overrides

This example shows the same Organisational Unit with multiple module overrides (including a module override which applies to all children).

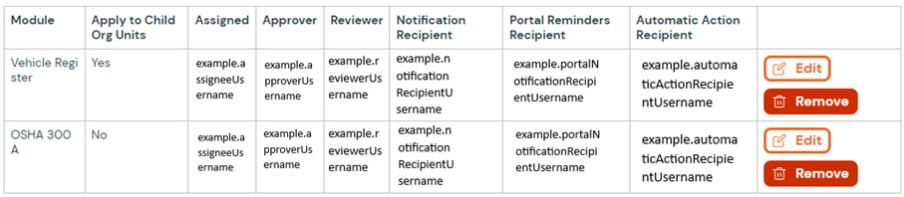
EVOTIX


```
{
    "externalId": "example.orgExternalId",
    "name": "example.orgName",
    "parentExternalId" : "example.orgParentExternalId",
    "defaultUsers": {
      "moduleOverrides" :
      [{
        "module" : "vehicle-register",
        "applyToChildUnits": "true" ,
        "assigneeUsername": "example.assigneeUsername",
        "approverUsername": "example.approverUsername",
        "reviewerUsername": "example.reviewerUsername",
        "notificationRecipientUsername":
"example.notificationRecipientUsername",
        "portalReminderRecipientUsername":
"example.portalReminderRecipientUsername",

"automaticActionRecipientUsername":"example.autoActionRecipientUsername"
      },
      {
        "module" : "osha300a",
        "applyToChildUnits": "false" ,
        "assigneeUsername": "example.assigneeUsername1",
        "approverUsername": "example.approverUsername1",
        "reviewerUsername": "example.reviewerUsername1",
        "notificationRecipientUsername":
"example.notificationRecipientUsername1",
        "portalReminderRecipientUsername":
"example.portalReminderRecipientUsername1",

"automaticActionRecipientUsername":"example.autoActionRecipientUsername1"
      }]
    }
}
```


The Organisational Unit's details will be the same as the 'minimum Organisational Unit's data' screen shot above. The difference in this case is that the two module overrides are present in the 'User Roles' section (with the vehicle register only override including children).



Module	Apply to Child Org Units	Assigned	Approver	Reviewer	Notification Recipient	Portal Reminders Recipient	Automatic Action Recipient	
Vehicle Register	Yes	example.assigneeUsername	example.approverUsername	example.reviewerUsername	example.notificationRecipientUsername	example.portalNotificationRecipientUsername	example.automaticActionRecipientUsername	Edit Remove
OSHA 300A	No	example.assigneeUsername	example.approverUsername	example.reviewerUsername	example.notificationRecipientUsername	example.portalNotificationRecipientUsername	example.automaticActionRecipientUsername	Edit Remove

EVOTIX

Organisational Unit with everything defined

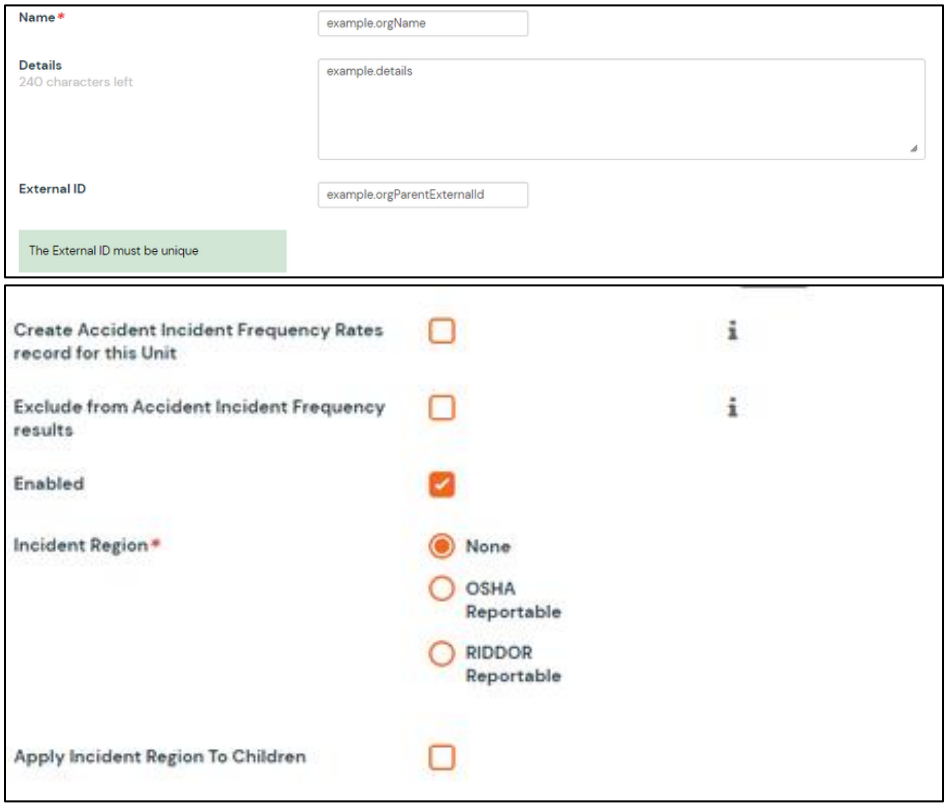
This example shows the same Organisational Unit with multiple module overrides (including a Organisational unit which applies to all children) and all the possible fields defined. Note that as per the OpenAPI schema the applyIncidentRegionToChildUnits field only takes effect when updating a Organisational Unit (you can include it when creating an Organisational Unit but it will be ignored).

```
{
    "externalId": "example.orgExternalId",
    "name": "example.orgName",
    "details" : "example.details",
    "incidentRegion" : "NONE",
    "applyIncidentRegionToChildUnits" : "false",
    "frequencyRateRecording": "INCLUDE",
    "parentExternalId" : "example.orgParentExternalId",
    "defaultUsers":
    {
      "approverUsername": "example.approverUsername",
      "assigneeUsername" : "example.assigneeUsername",
      "reviewerUsername": "example.reviewerUsername",
      "notificationRecipientUsername" :
"example.notificationRecipientUsername",
      "portalReminderRecipientUsername":
"example.portalReminderRecipientUsername",
      "autoActionRecipientUsername": "example.autoActionRecipientUsername",
      "assureGoPlusActionRaisedByUsername":
"example.assureGoPlusRaisedByUsername",
      "assureGoPlusActionRecipientUsername":
"example.assureGoPlusActionRecipientUsername",
      "humanResourcesRecipientUsername" :
"example.humanResourcesRecipientUsername",
      "alternativeHumanResourcesRecipientUsername" :
"example.altHumanResourceRecipientUsername",
      "moduleOverrides" :
      [{
        "module" : "vehicle-incident",
        "applyToChildUnits": "true" ,
        "assigneeUsername": "example.assigneeUsername",
        "approverUsername": "example.approverUsername",
        "reviewerUsername": "example.reviewerUsername",
        "notificationRecipientUsername":
"example.notificationRecipientUsername",
        "portalReminderRecipientUsername":
"example.portalReminderRecipientUsername",

"automaticActionRecipientUsername":"example.autoActionRecipientUsername"
      }]
    }
}
```

It is not necessary to set all the fields when creating/updating a Organisational field. Any of the fields which are not required can be omitted in which case the default value for the field will be applied. See the description against each field in the OpenAPI schema for details of the default behaviour.

The Organisational Unit's parent, default user's and module overrides will be the same as the above screenshots.



EVOTIX

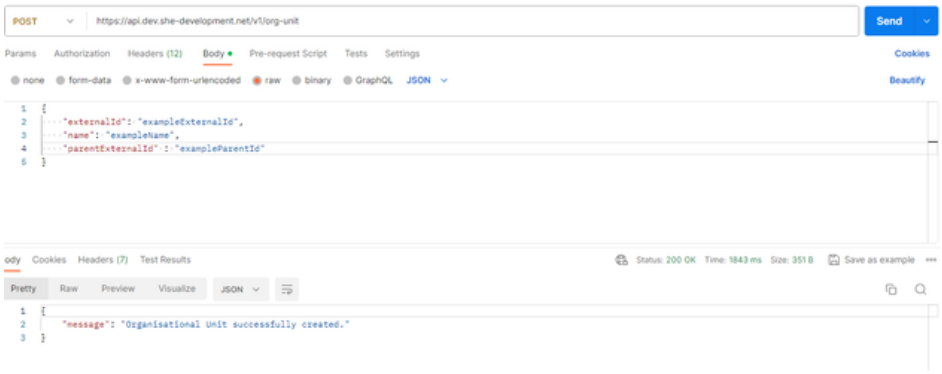
Create an Organisational Unit

The Customer API allows the creation of Organisational Unit in Assure using the /v1/org-unit API method with the POST verb. The Organisational Unit details are supplied in the body of the API request using the Organisational Unit JSON object (see section above). Organisational Units are uniquely identified by their External ID, which is the value in the externalId field of the Organisational Unit JSON object. If there is no Organisational Unit record for the external Id then the /v1/org-unit API method will create the Organisational Unit using the details in the Organisational Unit JSON object.

If there is an existing Organisational Unit record for the external ID then the /v1/org-unit API method will update the existing Organisational Unit record to match the supplied details (see the 'Update a Organisational Unit section below for details on the behaviour when updating an existing Organisational Unit).

Before attempting to use the /v1/org-unit API method make sure you understand the Organisational Unit JSON object (see the section above) and that you have the required details to access the API (i.e. the URL prefix, API key, etc), follow the Getting Started guide if you don't have these details already. The following table sections show how to create an Organisational Unit using the /v1/org-unit API method in a variety of software tools / platforms:

Any HTTP response code is the range 200-299 should be treated as success (do not use the contents of the response body for identifying success or failure as these may change). Implement effective error handling as per the guide for error handling.

Postman API platform	The screenshot below shows a successful 'create Organisational Unit's request for a customer on the uk2 stack. The API key is supplied via the 'Authorization' tab (see the Getting Started section in this document for how to setup the API key). The response section at the bottom shows the result Status: 200 OK which indicates that the Organisational Unit creation was successful (also displayed is the response message from the Customer API confirming the successful creation). NB: With tools like this where the Organisational Unit JSON object is manually entered you must ensure the contents of any string values are properly escaped. 
Windows Powershell	The code block below shows the few lines of Powershell script required to setup the Organisational Unit Object JSON and to make the 'create Organisational Unit ' request to the Customer API for a customer on the uk2 stack. The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX is where the API key needs to be placed. `$OrgUnitObjectJSON = @{` `"externalId" = "example.externalId"` `"name" = "example name"` `"parentId" = "example.parentExternalId"` `} | ConvertTo-Json` `Invoke-WebRequest `` `-Headers @{'x-api-key' = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'} `` `-Uri https://api.uk2.sheassure.net/v1/org-unit`` `-Method Post `` `-ContentType 'application/json' `` `-Body $OrgUnitObjectJSON`` The following shows the output from the above Powershell script code being run where a successful response is generated. The StatusCode: 200 indicates that the create Organisational Unit request was successful. Some response lines have been removed for brevity. `StatusCode : 200` `StatusDescription : OK` `Content : {"message":"Organisational Unit successfully created."}`

Python	The code block below shows the few lines of Python code required to setup the Organisational Unit Object JSON and to make the 'create Organisational Unit ' request to the Customer API for a customer on the uk2 stack. The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX is where the API key needs to be placed.
	
	`# pip install requests`
	`import requests`
	
	`orgUnit_object = {`
	`"externalId": "example.externalId",`
	`"name": "example name",`
	`"parentExternalId": "example.parentExternalId",`
	`}`
	
	`headers = { "x-api-key": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"}`
	`r = requests.post("https://api.uk2.sheassure.net/v1/org-unit", headers=headers, json = orgUnit_object)`
	
	`r.raise_for_status()`
	`print(f"StatusCode={r.status_code}")`
	`print(f"Body={r.content}")`
	
	The following shows the output from the above Python code being run where a successful response is generated. The StatusCode=200 indicates that the Organisational Unit creation was successful.
	
	`StatusCode=200`
	`Body=b'{"message": "Organisational Unit successfully created."}'`

EVOTIX

Update an Organisational Unit

The Customer API allows the creation of users in Assure using the /v1/org-unit API method with the POST verb. The Organisational Unit details are supplied in the body of the API request using the Organisational Unit JSON object (see section above). Organisational Units are uniquely identified by their External ID, which is the value in the externalId field of the Organisational Unit JSON object. If there is an existing Organisational Unit record for the external Id then the /v1/org-unit API method will update the existing Organisational Unit record to match the supplied details (see the 'Update an Organisational Unit's section below for details on the behaviour when updating an existing Organisational Unit). If the existing Organisational Unit record is disabled then the Organisational Unit will be re-enabled.

If there is no Organisational Unit record for the External Id then the /v1/org-unit API method will create the Organisational Unit using the details in the Organisational Unit JSON object (see the 'Create a Organisational Unit' section above for details on the behaviour when creating a Organisational Unit).

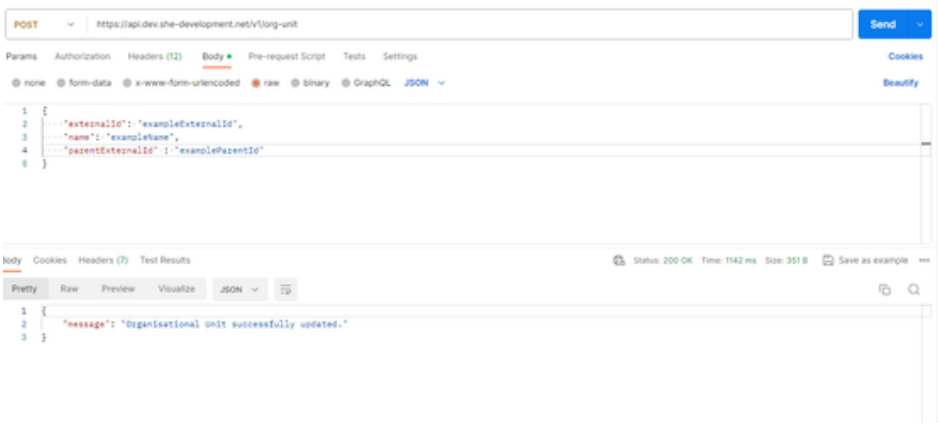
Before attempting to use the /v1/org-unit API method make sure you understand the Organisational Unit JSON object (see the section above) and that you have the required details to access the API (i.e. the URL, API key, etc), follow the Getting Started guide if you don't have these details already. The following table sections show how to update a Organisational Unit using the /v1/org-unit API method in a variety of software tools / platforms:

Partial updates are not supported i.e. you cannot update only selected Organisational Unit fields. The Organisational Unit JSON object must contain all the fields that are required to define the Organisational Unit data (same as if the Organisational Unit was being created). Assure will update the Organisational Unit record to match the details supplied, this includes applying the default values for any fields that are not provided in the Organisational Unit JSON object. See the OpenAPI schema for details of how the Organisational Unit data will be defaulted for each field.

Any HTTP response code is the range 200-299 should be treated as success (do not use the contents of the response body for identifying success or failure as these may change). Implement effective error handling as per the guide for error handling.

<table>
<tr>
<td>Postman API platform</td>
<td>The screenshot below shows a successful ‘update Organisational Unit’ request for a customer on the uk2 stack (NB: the Postman setup is identical to that used for creating a Organisational Unit). The API key is supplied via the ‘Authorization’ tab (see the Getting Started guide for how to setup the API key). The response section at the bottom shows the result Status: 200 OK which indicates that the Organisational Unit update was successful (also displayed is the response message from the Customer API confirming the successful update).

NB: With tools like this where the Organisational Unit JSON object is manually entered you must ensure the contents of any string values are properly escaped.

 
</td>
</tr>
<tr>
<td>Windows Powershell</td>
<td>The code block below shows the few lines of Powershell script required to setup the Organisational Unit Object JSON and to make the ‘update Organisational Unit’ request to the Customer API for a customer on the uk2 stack (NB: this is identical to the script for creating a Organisational Unit). The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX is where the API key needs to be placed.


```
$OrgUnitObjectJSON = @{
    "externalId" = "example.externalId"
    "name" = "example name"
    "parentId" = "example.parentExternalId"
} | ConvertTo-Json

Invoke-WebRequest `
    -Headers @{'x-api-key' = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'} `
    -Uri https://api.uk2.sheassure.net/v1/org-unit`
    -Method Post `
    -ContentType 'application/json' `
    -Body $OrgUnitObjectJSON`
```


The following shows the output from the above Powershell script code being run where a successful response is generated. The StatusCode: 200 indicates that the update Organisational Unit request was successful. Some response lines have been removed for brevity.


```
StatusCode        : 200
StatusDescription : OK
Content           : {"message":"Organisational Unit updated."}
```
</td>
</tr>
</table>

Python	The code block below shows the few lines of Python code required to setup the Organisational Unit Object JSON and to make the 'update Organisational Unit' request to the Customer API for a customer on the uk2 stack (NB: this is identical to the script for creating an Organisational Unit). The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX is where the API key needs to be placed.
	
	`# pip install requests`
	`import requests`
	
	`orgUnit_object = {`
	`"externalId": "example.externalId",`
	`"name": "example name",`
	`"parentExternalId": "example.parentExternalId",`
	`}`
	
	`headers = { "x-api-key": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"}`
	`r = requests.post("https://api.uk2.sheassure.net/v1/user", headers=headers, json = orgUnit_object)`
	
	`r.raise_for_status()`
	`print(f"StatusCode={r.status_code}")`
	`print(f"Body={r.content}")`
	
	The following shows the output from the above Python code being run where a successful response is generated. The StatusCode=200 indicates that the Organisational Unit update was successful.
	
	`StatusCode=200`
	`Body=b'{"message": "Organisational Unit updated."}'`

EVOTIX

Disable an Organisational Unit

The Customer API allows the disabling of Organisational Unit in Assure using the /v1/org-unit/{externalId} API method with the DELETE verb. The Organisational Unit to be disabled is identified by their external id supplied via the {externalId} path parameter. The Organisational Unit Object is NOT required for this method.

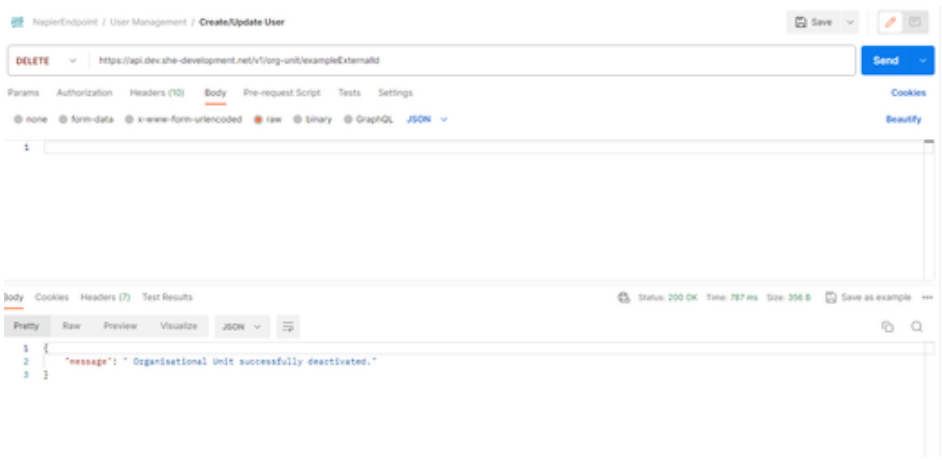
If there is no Organisational Unit record for the external Id or the Organisational Unit is already disabled then the /v1/org-unit/{externalId} API method will still return a successful response (i.e. HTTP status code in the range 200-299).

Before attempting to use the /v1/org-unit/{externalId} API method make sure you have the required details to access the API (i.e. the URL, API key, etc), follow the Getting Started guide if you don't have these details already. The following table sections show how to disable an Organisational Unit using the /v1/org-unit/{externalId} API method in a variety of software tools / platforms:

Complete deletion of a Organisational Unit record is not currently supported via the Customer API. Fully deleting an Organisational Unit record requires manual steps in Assure to associate the Organisational Unit data with another Organisational Unit (or archive the data).

Any HTTP response code is the range 200-299 should be treated as success (do not use the contents of the response body for identifying success or failure as these may change). Implement effective error handling as per the guide for error handling.

IMPORTANT - The external ID value needs to have URL escaping applied before inclusion in the URL.

Postman API platform	The screenshot below shows a successful 'disable Organisational Unit' request for a customer on the uk2 stack. Note the use of the 'Authorization' tab to configure the API key header. The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX value is where the API key would be placed. The response section at the bottom shows the result Status: 200 OK which indicates that the Organisational Unitdisable was successful (also displayed is the response message from the Customer API confirming the successful disable). NB: With tools like this where the externalId to be deleted is manually entered you must ensure that URL escaping is applied to the external Id. 

Windows Powershell	The code block below shows the few lines of Powershell script required to make the 'disable Organisational Unit' request to the Customer API for a customer on the uk2 stack. The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX is where the API key needs to be placed.
	
	IMPORTANT – Note the use of [System.Net.WebUtility]::UrlEncode to ensure that the external ID value is properly escaped for inclusion in the URL.
	
	`$URL = "https://api.uk2.sheassure.net/v1/org-unit/" + [System.Net.WebUtility]::UrlEncode("example?externalId") Invoke-WebRequest ` -Headers @{'x-api-key' = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'} ` -Uri $URLEscaped ` -Method Delete ``
	
	The following shows the output from the above Powershell script code being run where a successful response is generated. The StatusCode: 200 indicates that the disable Organisational Unit request was successful. Some response lines have been removed for brevity.
	
	`StatusCode : 200 StatusDescription : OK Content : {"message": "Organisational Unit successfully deactivated."}`

Python

The code block below shows the few lines of Python code required to make the 'disable Organisational Unit' request to the Customer API for a customer on the uk2 stack. The XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX is where the API key needs to be placed.

IMPORTANT - Note the use of urllib.parse.quote to ensure that the external Id is properly escaped for inclusion in the URL.

```
# pip install requests
# pip install urllib
import requests
import urllib

url = "https://api.uk2.sheassure.net/v1/org-unit/" +
urllib.parse.quote('example.externalId', safe='')
headers = { "x-api-key": "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"}
r = requests.delete(url, headers=headers)

r.raise_for_status()
print(f"StatusCode={r.status_code}")
print(f"Body={r.content}")
```

The following shows the output from the above Python code being run where a successful response is generated. The StatusCode=200 indicates that the Organisational Unit disable was successful.

```
StatusCode=200
Body=b'{"message": "Organisational Unit successfully deactivated."}'
```