

# How can Sub-resources be Transformed?

Last Modified on 02/20/2019 7:41 am EST

Transformations can be mapped to sub-resources (children) of objects, but it is important to know that transformed sub-resources are not added to the API Docs. To accomplish this task, you have to create a common resource and transformation to the sub-object within the root object. For example, in Connectwise CRM there is a sub-resource `time-entries` within `incidents` (e.g., `/incidents/{id}/time-entries`). Create a common resource that includes the fields you require, and then map a Connectwise Transformation to this resource for the `incidentsTimeEntries` object.

Since this transformed sub-resource is not added to Swagger you will need to make API calls through `GET /{objectName}/{id}/{childObjectName}` where the `objectName` is the original resource and the `childObjectName` is the transformed sub-resource. Here is a Connectwise CRM example that shows mapping of the sub/child resource `incidentsTimeEntries` to a common resource called "myIncidentTime".

The screenshot shows the configuration interface for a common resource named "myIncidentTime". At the top, there is a text input field containing "myIncidentTime" and an "Edit" button. Below this is a "Sort Fields By:" dropdown menu set to "alphabetical". To the right, the interface is titled "ConnectWise for Docs (Connectwise CRM REST Beta) /incidentsTimeEntries" and includes navigation buttons (back, forward, settings, delete, save) and a "Load metadata by id" toggle switch.

The main area is divided into three sections for field management:

- Organization Level Fields:** A table with columns for Field Name, Type, and Display Name, with a plus sign to add fields.
- Account Level Fields:** A table with columns for Field Name, Type, and Display Name. It contains two rows:
  - Field Name: "Hours-Billed", Type: "string", Display Name: "hoursBilled".
  - Field Name: "Notes", Type: "string", Display Name: "notes".
- Instance Level Fields:** A table with columns for Field Name, Type, and Display Name, with a plus sign to add fields.

With this transformation in mind and knowing that you are trying to reach the sub-resource/child object (in this example `incidentsTimeEntries`), here is an example of how you must make the API call to:

```
/{objectName}/{id}/{childObjectName}
```

```
/incidents/{id}/myIncidentTime
```

GET /{objectName}/{objectId}/{childObjectName} Search for {childObjectName}

Parameters Cancel

Name	Description
<b>Authorization</b> * required string (header)	The authorization tokens. The format for the header value is 'Element <token>, User <user secret>' <input type="text" value=""/>
<b>objectName</b> * required string (path)	The name of the object <input type="text" value="incidents"/>
<b>objectId</b> * required string (path)	The {objectName} ID <input type="text" value="3"/>
<b>childObjectName</b> * required string (path)	The name of the childObjectName <input type="text" value="myIncidentTime"/>
<b>where</b> string (query)	The CEQL search expression. <input type="text" value="where - The CEQL search expression."/>