Synchronization GE SceneAPI v1 Specification

DATE: 14 September 2015

This version:

http://realxtend.github.io/fiware-synchronization/api/v1.0

Latest version:

http://realxtend.github.io/fiware-synchronization/api/latest

Editors

1. Lasse Öörni, LudoCraft Ltd.

Copyright

Abstract

The SceneAPI in the Synchronization GE provides a RESTful API for querying and modifying Entity-Component-Attribute formatted scene data. The scene data is communicated via XML.

For a description of the Entity-Component-Attribute data model, see [Scene and EC model] (https://github.com/realXtend/tundra/wiki/Scene-and-EC-Model).

REST API

Status of this document

Final version of the SceneAPI.

Table of Contents

API Specification

Scene

The resources include scene entities, which contain components, which contain attributes.

Entities [/entities]

The collection of all entities in the scene.

Retrieve Data of All Entities - GET /entities

The entities' data, including their components and attributes is returned in XML format.

Response 200 (application/xml)

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entities/retrieve-data-of-all-entities)

Retrieve an Entity by Name - GET /entities?name={entity_name}

Using the query syntax, search for an entity by name. The entity's data and the components it contains + their attributes are returned as XML.

Response 200 (application/xml)

Response 404

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entities/retrieve-an-entity-by-name)

Create a New Entity - POST /entities

The POST request body may contain XML-serialized data to be put inside the entity (components & their attribute values). If omitted, an empty entity is created. A free ID is assigned to the new entity and its data is returned as XML.

Response 200 (application/xml)

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entities/create-a-new-entity)

Entity [/entities/{entity_id}]

Represents a single entity in the scene. In addition to the components it contains, each entity has an integer ID and a temporary flag, which means whether or not to persist it when the scene is saved on the server.

Retrieve Data of an Entity - GET /entities/{entity_id}

The entity's data and the components it contains + their attributes are returned as XML.

Response 200 (application/xml)

Response 404

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entity/retrieve-data-of-an-entity)

Create a New Entity with a Specified ID - POST /entities/{entity_id}

The POST request body may contain XML data to be put inside the entity (components & their attribute values). If omitted, an empty entity is created. If there is an ID conflict with an existing entity, another free ID is chosen. The entity's data is returned as XML.

Response 200 (application/xml)

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entity/create-a-new-entity-with-a-specified-id)

Set Entity's Component Data - PUT /entities/{entity_id}

The entity with the specified ID must already exist in the scene. Its new components and their attribute data are to be contained in the PUT request body as XML. All existing components in the entity will be deleted and replaced with those contained in the request body. The entity's new data is returned as XML.

Response 200 (application/xml)

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entity/set-entity's-component-data)

Delete an Entity - DELETE /entities/{entity_id}

Response 200

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/entity/delete-an-entity)

Component [/entities/{entity_id}/{component_type}]

Components within an entity are identified by their type name, for example Placeable, Name or Script. They may also be temporary, which is indicated by a boolean.

Retrieve Data of a Component - GET /entities/{entity_id}/{component_type}

Returns the component's data as XML.

Response 200 (application/xml)

Response 404

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/component/retrieve-data-of-a-component)

Create a New Component Into an Entity - POST /entities/{entity_id}/{component_type}

The POST request body may contain initial attribute values as XML to be put into the created component. If omitted, the attributes will have default values. The component's new data is returned as XML.

Response 200 (application/xml)

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/component/create-a-new-component-into-an-entity)

Set New Attribute Values Into a Component - PUT /entities/{entity_id}/{component_type}

The attribute values are to be contained inside the PUT request body as XML. The component's new data is returned as XML.

Response 200 (application/xml)

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/component/set-new-attribute-values-into-a-component)

Set New Attribute Value Into a Component - PUT /entities/{entity_id}/{component_type}?{attribute_name}={new_attribute_value}

Set a single attribute inside a component to a new value using the query syntax. The component's new data is returned as XML.

Response 200

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/component/set-new-attribute-value-into-a-component)

Delete a Component From an Entity - DELETE /entities/{entity_id}/{component_type}

Response 200

Response 400

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/component/delete-a-component-from-an-entity)

Attribute [/entities/{entity_id}/{component_type}/{attribute_name}]

Attributes are typed values (for example integer, boolean, 3-dimensional vector) inside the component, identified by their names.

Retrieve the Value of an Attribute - GET /entities/{entity_id}/{component_type}/{attribute_name}

The value is returned as text.

Response 200 (text/plain)

Response 404

Go to example

View in Apiary (http://docs.sceneapi.apiary.io/#reference/scene/attribute/retrieve-the-value-of-an-attribute)

Examples

Scene

Entities [/entities]

Retrieve Data of All Entities - GET /entities

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<scene>
 <entity temporary="false" id="1" sync="true">
  <component temporary="false" typeId="5" type="Script" sync="true">
   <attribute value="avatarmenu.js" type="AssetReferenceList" id="scriptRef" name="Script ref"/>
   <attribute value="true" type="bool" id="runOnLoad" name="Run on load"/>
   <attribute value="0" type="int" id="runMode" name="Run mode"/>
   <attribute value="" type="string" id="applicationName" name="Script application name"/>
   <attribute value="" type="string" id="className" name="Script class name"/>
  </component>
  <component temporary="false" typeId="26" type="Name" sync="true">
```

```
   <attribute value="AvatarMenu" type="string" id="name" name="Name"/>
   <attribute value="" type="string" id="description" name="Description"/>
   <attribute value="" type="string" id="group" name="Group"/>
  </component>
 </entity>
 <entity temporary="false" id="2" sync="true">
  <component temporary="false" typeId="5" type="Script" sync="true">
   <attribute value="avatarapplication.js;simpleavatar.js;exampleavataraddon.js" type="AssetReferenceList" id="scriptRef" name="Script ref"/>
   <attribute value="true" type="bool" id="runOnLoad" name="Run on load"/>
   <attribute value="0" type="int" id="runMode" name="Run mode"/>
   <attribute value="AvatarApp" type="string" id="applicationName" name="Script application name"/>
   <attribute value="" type="string" id="className" name="Script class name"/>
  </component>
  <component temporary="false" typeId="26" type="Name" sync="true">
   <attribute value="AvatarApp" type="string" id="name" name="Name"/>
   <attribute value="" type="string" id="description" name="Description"/>
   <attribute value="" type="string" id="group" name="Group"/>
  </component>
 </entity>
</scene>
```

Go to specification

Retrieve an Entity by Name - GET /entities?name={entity_name}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<entity temporary="false" id="1" sync="true">
 <component temporary="false" typeId="5" type="Script" sync="true">
  <attribute value="avatarmenu.js" type="AssetReferenceList" id="scriptRef" name="Script ref"/>
  <attribute value="true" type="bool" id="runOnLoad" name="Run on load"/>
  <attribute value="0" type="int" id="runMode" name="Run mode"/>
  <attribute value="" type="string" id="applicationName" name="Script application name"/>
  <attribute value="" type="string" id="className" name="Script class name"/>
 </component>
 <component temporary="false" typeId="26" type="Name" sync="true">
  <attribute value="AvatarMenu" type="string" id="name" name="Name"/>
  <attribute value="" type="string" id="description" name="Description"/>
  <attribute value="" type="string" id="group" name="Group"/>
 </component>
</entity>
```

Response 404

Go to specification

Create a New Entity - POST /entities

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<entity temporary="false" id="1" sync="true"/>
```

Response 400

Go to specification

Entity [/entities/{entity_id}]

Retrieve Data of an Entity - GET /entities/{entity_id}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<entity temporary="false" id="1" sync="true">
 <component temporary="false" typeId="5" type="Script" sync="true">
  <attribute value="avatarmenu.js" type="AssetReferenceList" id="scriptRef" name="Script ref"/>
  <attribute value="true" type="bool" id="runOnLoad" name="Run on load"/>
  <attribute value="0" type="int" id="runMode" name="Run mode"/>
  <attribute value="" type="string" id="applicationName" name="Script application name"/>
  <attribute value="" type="string" id="className" name="Script class name"/>
 </component>
 <component temporary="false" typeId="26" type="Name" sync="true">
  <attribute value="AvatarMenu" type="string" id="name" name="Name"/>
  <attribute value="" type="string" id="description" name="Description"/>
  <attribute value="" type="string" id="group" name="Group"/>
 </component>
</entity>
```

Response 404

Go to specification

Create a New Entity with a Specified ID - POST /entities/{entity_id}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<entity temporary="false" id="10" sync="true"/>
```

Response 400

Go to specification

Set Entity's Component Data - PUT /entities/{entity_id}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<entity temporary="false" id="1" sync="true">
 <component temporary="false" typeId="20" type="Placeable" sync="true">
  <attribute value="0.000000,50.000000,0.000000,0.000000,0.000000,0.000000,1.000000,1.000000,1.000000" type="Transform" id="transform" name="Transform"/>
  <attribute value="false" type="bool" id="drawDebug" name="Show bounding box"/>
  <attribute value="true" type="bool" id="visible" name="Visible"/>
  <attribute value="1" type="int" id="selectionLayer" name="Selection layer"/>
  <attribute value="" type="EntityReference" id="parentRef" name="Parent entity ref"/>
  <attribute value="" type="string" id="parentBone" name="Parent bone name"/>
 </component>
 <component temporary="false" typeId="26" type="Name" sync="true">
  <attribute value="NewName" type="string" id="name" name="Name"/>
  <attribute value="" type="string" id="description" name="Description"/>
  <attribute value="" type="string" id="group" name="Group"/>
 </component>
```

```
</entity>
```

Response 400

Go to specification

Delete an Entity - DELETE /entities/{entity_id}

Response 200

Response 400

Go to specification

Component [/entities/{entity_id}/{component_type}]

Retrieve Data of a Component - GET /entities/{entity_id}/{component_type}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<component temporary="false" typeId="5" type="Script" sync="true">
 <attribute value="avatarmenu.js" type="AssetReferenceList" id="scriptRef" name="Script ref"/>
 <attribute value="true" type="bool" id="runOnLoad" name="Run on load"/>
 <attribute value="0" type="int" id="runMode" name="Run mode"/>
 <attribute value="" type="string" id="applicationName" name="Script application name"/>
 <attribute value="" type="string" id="className" name="Script class name"/>
</component>
```

Response 404

Go to specification

Create a New Component Into an Entity - POST /entities/{entity_id}/{component_type}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<component temporary="false" typeId="20" type="Placeable" sync="true">
 <attribute value="0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,1.000000,1.000000,1.000000" type="Transform" id="transform" name="Transform"/>
 <attribute value="false" type="bool" id="drawDebug" name="Show bounding box"/>
 <attribute value="true" type="bool" id="visible" name="Visible"/>
 <attribute value="1" type="int" id="selectionLayer" name="Selection layer"/>
 <attribute value="" type="EntityReference" id="parentRef" name="Parent entity ref"/>
 <attribute value="" type="string" id="parentBone" name="Parent bone name"/>
</component>
```

Response 400

Go to specification

Set New Attribute Values Into a Component - PUT /entities/{entity_id}/{component_type}

Response 200 (application/xml)

Headers

```
Content-Type: application/xml
```

Body

```
<component temporary="false" typeId="20" type="Placeable" sync="true">
 <attribute value="0.000000,50.000000,0.000000,0.000000,0.000000,0.000000,1.000000,1.000000,1.000000" type="Transform" id="transform" name="Transform"/>
 <attribute value="false" type="bool" id="drawDebug" name="Show bounding box"/>
 <attribute value="true" type="bool" id="visible" name="Visible"/>
 <attribute value="1" type="int" id="selectionLayer" name="Selection layer"/>
 <attribute value="" type="EntityReference" id="parentRef" name="Parent entity ref"/>
 <attribute value="" type="string" id="parentBone" name="Parent bone name"/>
</component>
```

Response 400

Go to specification

Set New Attribute Value Into a Component - PUT /entities/{entity_id}/{component_type}?{attribute_name}={new_attribute_value}

Response 200

Body

```
<component temporary="false" typeId="20" type="Placeable" sync="true">
 <attribute 
value="0.000000,50.000000,0.000000,0.000000,0.000000,0.000000,1.000000,1.000000,1.0
00000" type="Transform" id="transform" name="Transform"/>
 <attribute value="false" type="bool" id="drawDebug" name="Show bounding box"/>
 <attribute value="true" type="bool" id="visible" name="Visible"/>
 <attribute value="1" type="int" id="selectionLayer" name="Selection layer"/>
 <attribute value="" type="EntityReference" id="parentRef" name="Parent entity ref"/
>
 <attribute value="" type="string" id="parentBone" name="Parent bone name"/>
</component>
```

Response 400

Go to specification

Delete a Component From an Entity - DELETE /entities/{entity_id}/{component_type}

Response 200

Response 400

Go to specification

Attribute [/entities/{entity_id}/{component_type}/{attribute_name}]

Retrieve the Value of an Attribute - GET /entities/{entity_id}/{component_type}/{attribute_name}

Response 200 (text/plain)

Headers

```
Content-Type: text/plain
```

Body

```
0.000000,50.000000,0.000000,0.000000,0.000000,0.000000,1.000000,1.000000,1.000000
```

Response 404

Go to specification