

Storing nodes and sub-nodes using SLI and AAIService (under construction)

AAI Database is implemented using graph database. To interact with any node, user needs to know the path to the node. The definitions of all nodes are provided in [aai_schema_v11.xsd](#) document.

All paths supported by A&AI service are defined in the [aai_swagger_v11.html](#) document that can be found on [A&AI REST API Documentation](#) wiki page. The general rule of using CRUD API of AAIService is described in the [Introduction to CCSDK-Adaptors/AAIService component](#) wiki page.

Step-by-step guide to work with subnodes

All DG node transactions are focused on interacting with a node itself, meaning when user wants to perform a transaction for a specific type, it needs to use its type name as a resource, and in case of <save> or <update>, it needs to set the resource value using node's type. Let's take as an example a task to insert the following service-instance data to AAI:

sample service-instance json data

```
{
  "service-instance-id": "MY_DOMAIN",
  "service-instance-name": "MY_DOMAIN",
  "resource-version": "1508232694103"
},
"metadata": {
  "metadatum": [
    {
      "metaname": "domain-id",
      "metaval": "9eec1c7c-4e71",
      "resource-version": "1508230566629"
    },
    {
      "metaname": "EmailID:user@gmail.com",
      "metaval": "3ef8cc1c-6ad6",
      "resource-version": "1508230567068"
    }
  ]
}
```

AAIService API does not allow to insert the whole JSON graph to AAI, but it allows to operate on each node individually.

Therefore user needs to write 3 DG nodes to deal with each node individually: service-instance node first, then 2 metadata nodes, second.

1. First step is to write a DG node that points to service-instance. Based on the [aai_swagger_v1.html](#) document, the path to service instance is:
 - a. /aai/v11/business/customers/customer/{global-customer-id}/service-subscriptions/service-subscription/{service-type}/service-instances/service-instance/{service-instance-id}
 - b. the <save> node requires user to provide name-value pairs that identify this path uniquely, they will contain following names:
 - i. customer.global-customer-id
 - ii. service-subscription.service-type
 - iii. service-instance.service-instance-id
 - c. the request key string will look like this, assuming that the target path is: /aai/v11/business/customers/customer/my-customer/service-subscriptions/service-subscription/SERVICE-TYPE-DOMAIN/service-instances/service-instance/MY_DOMAIN

request key string

```
key="customer.global-customer-id = 'my-customer' AND service-subscription.service-type = 'SERVICE-TYPE-DOMAIN' AND service-instance.service-instance-id = 'MY_DOMAIN' "
```

The whole DG node will be defined as follows:

sample <save> request

```
<save
  plugin="org.openecomp.sdnc.sli.aai.AAIService"
  resource="service-instance"
  key="customer.global-customer-id = 'my-customer' AND service-subscription.service-type =
'SERVICE-TYPE-DOMAIN' AND service-instance.service-instance-id = 'MY_DOMAIN'"
  force="true"
  local-only="false"
  pfx="tmp.AnAI-data.vnf" >
<parameter name="service-instance-id" value="MY_DOMAIN" />
<parameter name="service-instance-name" value="MY_DOMAIN" />
</save>
```

2. The next step is to save sub-nodes of type 'metadatum';

a. the request key will contain additional name-value pair identifying metadatum node, for example for the first node, it will look as follows:

request key string

```
key="customer.global-customer-id = 'my-customer' AND service-subscription.service-type = 'SERVICE-
TYPE-DOMAIN' AND service-instance.service-instance-id = 'MY_DOMAIN' AND metadatum.metaname =
'domain-id' "
```

The whole DG node will be defined as follows

<save> matadatum request node

```
<save
  plugin="org.openecomp.sdnc.sli.aai.AAIService"
  resource="metadatum"
  key="customer.global-customer-id = 'my-customer' AND service-subscription.service-type =
'SERVICE-TYPE-DOMAIN' AND service-instance.service-instance-id = 'MY_DOMAIN' AND metadatum.
metaname = 'domain-id' "
  force="true"
  local-only="false"
  pfx="tmp.AnAI-data.vnf" >
<parameter name="metaname" value="domain-id" />
<parameter name="metaval" value="9eec1c7c-4e71" />
</save>
```

b. the request key for the second 'metadatum' node will look as follows:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
key="customer.global-customer-id = 'my-customer' AND service-subscription.service-type = 'SERVICE-
TYPE-DOMAIN' AND service-instance.service-instance-id = 'MY_DOMAIN' AND metadatum.metaname =
'EmailID:user@gmail.com' "
```

The whole DG node will be defined as follows:

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
<save
  plugin="org.openecomp.sdnc.sli.aai.AAIService"
  resource="metadatum"
  key="customer.global-customer-id = 'my-customer' AND service-subscription.service-type =
'SERVICE-TYPE-DOMAIN' AND service-instance.service-instance-id = 'MY_DOMAIN' AND metadatum.metaname
= 'EmailID:user@gmail.com' "
  force="true"
  local-only="false"
  pfx="tmp.AnAI-data.vnf" >
<parameter name="metaname" value="EmailID:user@gmail.com" />
<parameter name="metaval" value="3ef8cc1c-6ad6" />
</save>
```

3. If it is needed to add sub-nodes for a 'sub-node' itself, let's say we want to add a sub-node to metadatum, you need to repeat step 2 and expand the request key by adding the name-value pair identifying next sub-node.



Related articles

- [One-to-N odl message router](#)
- [model-driven DMaaP Agent](#)
- [\[ccsdk\] Team](#)
- [ONAP Beijing: Understanding the vFWCL use-case mechanism](#)
- [Storing nodes and sub-nodes using SLI and AAIService \(under construction\)](#)