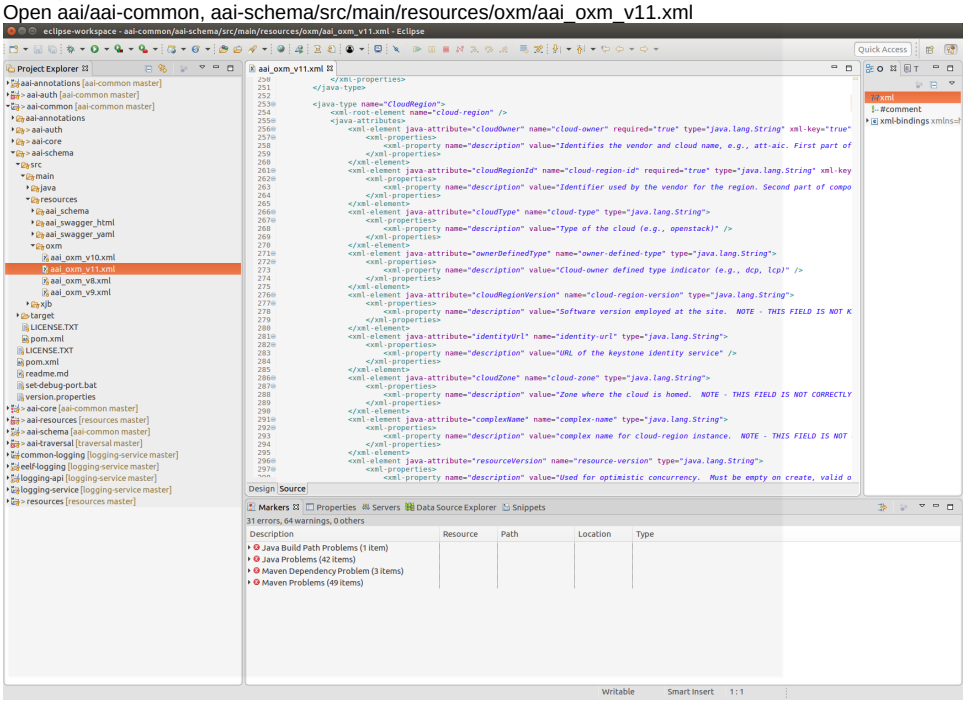
Tutorial: Making and Testing a Schema Change in A&AI in Beijing Release

1. Set up a development environment as described in A&AI Developer Environment Setup - Updated for Beijing!
2. This document will show editing schema files with Eclipse, but Eclipse is not required.
3. In this example, we will add 2 new attributes to the cloud-region node type and a new node type called "new-widget" which will be a child node of "cloud-region"
4. Adding a new attribute to "cloud-region"
5. Open aai/aai-common, aai-schema/src/main/resources/oxm/aai_oxm_v11.xml



6. Add the following attributes between "complexName" and "resourceVersion":

```
<xml-element java-attribute="complexName" name="complex-name" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="complex name for cloud-region instance. NOTE - THIS FIELD IS NOT CORRECTLY POPULATED." />
</xml-properties>
</xml-element>
<xml-element java-attribute="newAttributeForDemo" name="new-attribute-for-demo" required="true" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="Example new attribute for cloud-region instance. " />
</xml-properties>
</xml-element>
<xml-element java-attribute="numberAttributeForDemo" name="number-attribute-for-demo" required="true" type="java.lang.Integer">
<xml-properties>
<xml-property name="description" value="Example number attribute for cloud-region instance. " />
</xml-properties>
</xml-element>
<xml-element java-attribute="resourceVersion" name="resource-version" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="Used for optimistic concurrency. Must be empty on create, valid on update and delete." />
</xml-properties>
</xml-element>
```

7. Save the file, and rebuild the libraries and microservices:

   here's my example file: aai_oxm_v11.xml
   a. Rebuild aai-common first:
   b. $ cd ~/LF/AAI/aai-common
   c. $ mvn versions:set -DnewVersion=0.0.1-TEST-SNAPSHOT
   d. $ mvn clean install
      Should result in BUILD SUCCESS

e. $ cd ~/LF/AAI/resources
f. $ mvn clean install -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
Should result in BUILD SUCCESS
g. $ cd ~/LF/AAI/traversal
mvn clean install -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
h. Should result in BUILD SUCCESS

From https://lists.onap.org/g/onap-discuss/message/11507:

> *The html and yaml gets automatically generated when you run install but the xsd was never part of the autogenerate maven profile.*
> *For Beijing and earlier release, we never auto generated it. In order to run the generation of the xsd, here is the command:*
>
> *cd ~/LF/aai/aai-common/aai-core/*
> *mvn -PgenerateXsd install -DskipTests -Dgendoc.version=v14*
>
> *I have actually noticed that recently within the internal release and realized that it makes sense for xsd to be also auto generated.*
> *I am not sure why it was left out from the autogenerate profile but when developing the model driven feature, I ensured that the auto generate also does xsd generation.*
> *Once that feature gets delivered to Casablanca, there wouldn't be a need to run the profile manually.*

8. Run GenTester, using the target dir under aai-resources:

$ cd ~/LF/AAI/graphadmin;

java -DAJSC_HOME=. -DBUNDLECONFIG_DIR=src/main/resources/ -Dloader.main=org.onap.aai.schema.GenTester -Dloader.path=./src/main/resources -Dschema.ingest.file=src/main/resources/application.properties -jar target/aai-graphadmin-*.jar

You should see the following output:

---- NOTE --- about to open graph (takes a little while)--------;

-- loading schema into JanusGraph

-- loading schema into JanusGraph

-- Loading new schema elements into JanusGraph --

-- graph commit

-- graph shutdown

You can check the logs to see if the graph elements were created successfully and grep for your property
cd logs/createDBSchema/;
grep 'complex-name' metrics.log

2018-12-06T04:43:18.061+0000|2018-12-06T04:43:42.960+0000|449b8583-54c1-42bf-8ada-5b75c52d7a13||main ||AAI|AAI-TOOLS|AAI|main|COMPLETE|0|||INFO||127.0.1.1|24899|localhost||org.onap.aai.dbgen.SchemaGenerator|||||||co=DBGenTester:Creating PropertyKey: [complex-name], [String], [SINGLE]
2018-12-06T04:43:18.061+0000|2018-12-06T04:43:42.960+0000|449b8583-54c1-42bf-8ada-5b75c52d7a13||main ||AAI|AAI-TOOLS|AAI|main|COMPLETE|0|||INFO||127.0.1.1|24899|localhost||org.onap.aai.dbgen.SchemaGenerator|||||||co=DBGenTester:Add index for PropertyKey: [complex-name]

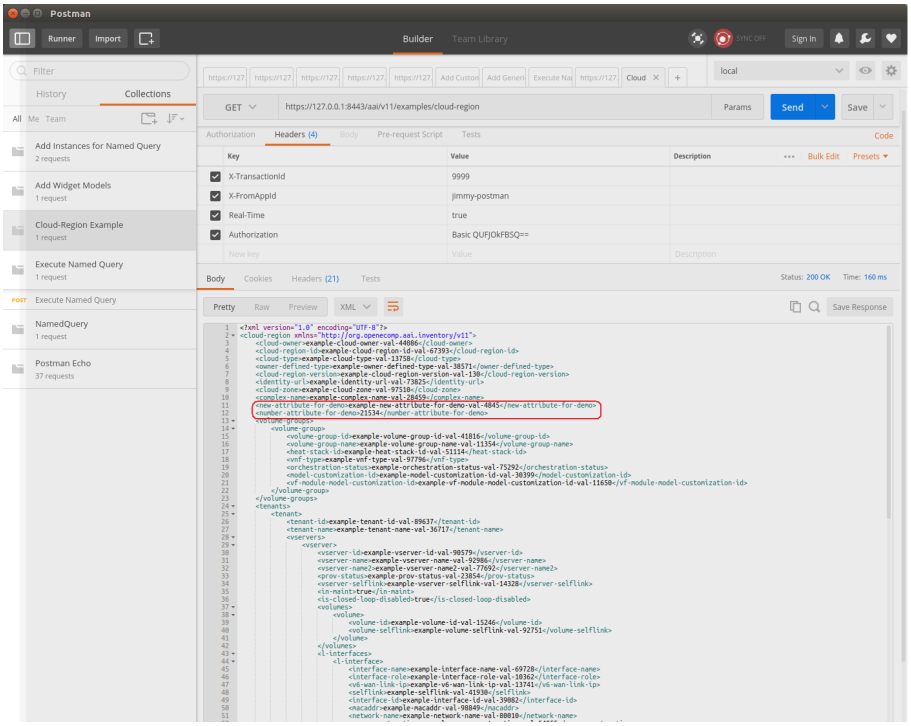
9. Start the "resources" microservice
    a. Resources runs on port 8447. Go to the resources directory
    $ cd ~/LF/AAI/resources
    b. Set the debug port to 9447
    $ export MAVEN_OPTS="-Xms1024m -Xmx5120m -XX:PermSize=2024m -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=9447,server=y,suspend=n"
    c. Start the microservice
    $ mvn -pl aai-resources -P runAjsc -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT

10. Start the "traversal" microservice
    a. Traversal runs on port 8446. Go to the traversal directory
    $ cd ~/LF/AAI/traversal

b. Set the debug port to 9446
$ export MAVEN_OPTS="-Xms1024m -Xmx5120m -XX:PermSize=2024m -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=9446,server=y,suspend=n"

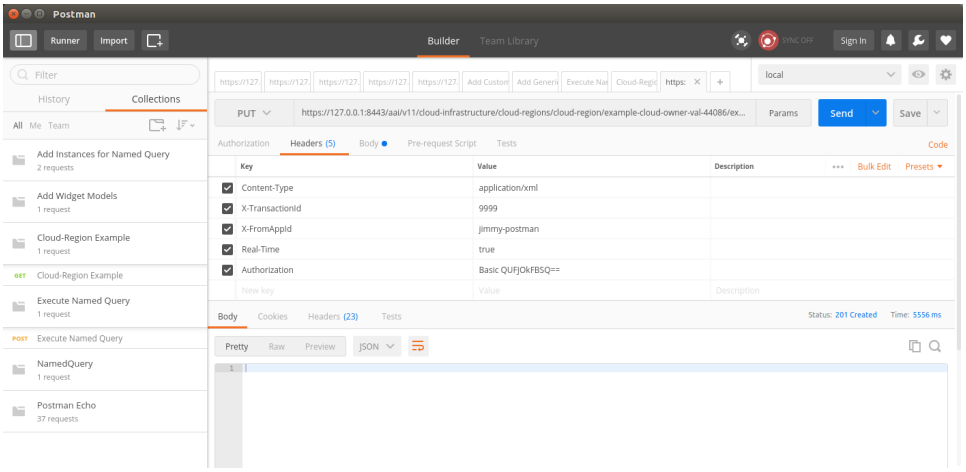
c. Start the microservice
$ mvn -pl aai-traversal -PrunAjsc -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
Should see something like this: Traversal Microservice Started

11. Get an example cloud-region object, postman: Cloud-Region Example.postman_collection.json

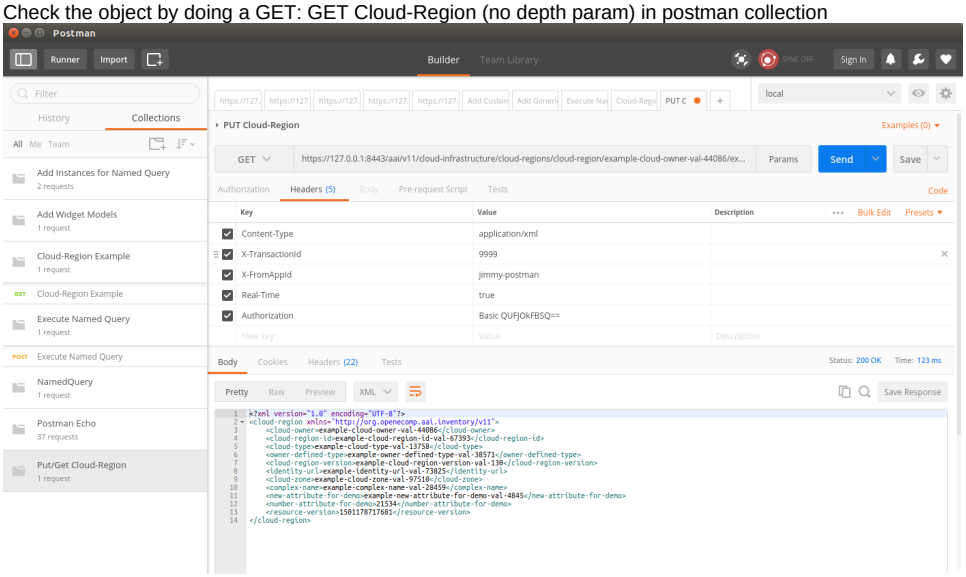


You should see the new attributes on the example object, as highlighted in the red rectangle above

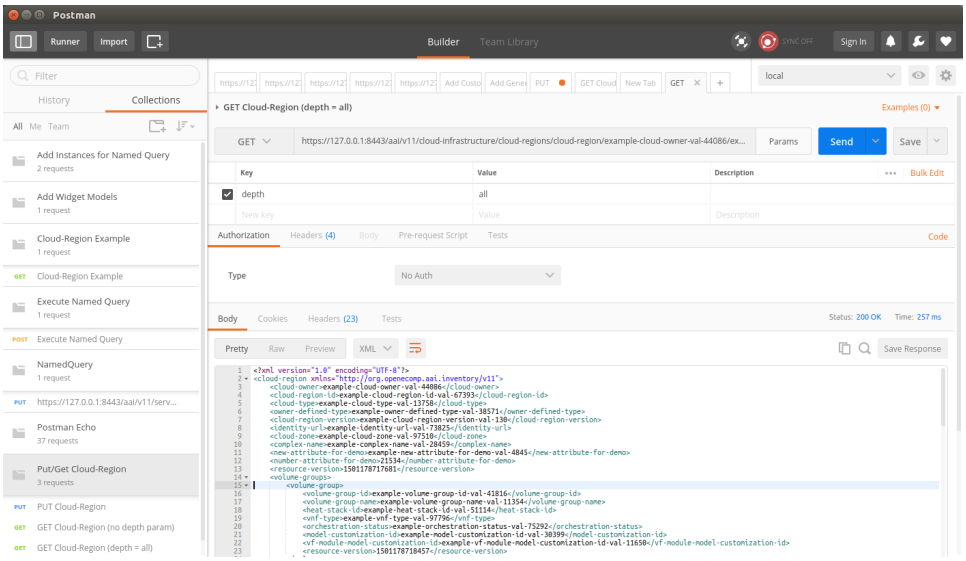
12. Copy and paste the example object and PUT it to persist the new attributes. Use this postman collection thru step 15



13. Check the object by doing a GET: GET Cloud-Region (no depth param) in postman collection



14. Add the depth parameter: GET Cloud-Region (depth = all) in postman collection



    a. This indicates that the schema has been updated with the new attributes and they are making it to the database

15. Clear the cloud region and then delete it by using "Clear Cloud-Region" and "Delete Cloud-Region"
    a. note: you will need to perform a GET each time and update the resource version of the cloud-region in the XML payload for the clear, and do the GET again and update the resource-version QueryParam to perform the delete

16. Run PUT Cloud-Region - missing attr. This will try to PUT the object without the required attribute we defined in the schema, and the response will look like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Fault>
<requestError>
<serviceException>
<messageId>SVC3000</messageId>
<text>Invalid input performing %1 on %2 (msg=%3) (ec=%4)</text>
<variables>
<variable>PUT</variable>
<variable>v11/cloud-infrastructure/cloud-regions/cloud-region/example-cloud-owner-val-44086/example-cloud-region-id-val-67393</variable>
<variable>Invalid input performing %1 on %2:Missing required property: new-attribute-for-demo</variable>
<variable>ERR.5.2.3000</variable>
</variables>
</serviceException>
</requestError>
</Fault>
```

17. Adding a new node type. First, add the container under the top-level inventory category. Adding "newWidgets" to "Network" so the URI will be "aai/v11/network/new-widgets/new-widget/{new-widget-name}"

```
<java-type name="Network">
<xml-properties>
<xml-property name="description" value="Namespace for network inventory resources." />
</xml-properties>
<xml-root-element name="network" />
<java-attributes>
<xml-element java-attribute="logicalLinks" name="logical-links" type="inventory.aai.onap.org.v11.LogicalLinks" />
<xml-element java-attribute="sitePairSets" name="site-pair-sets" type="inventory.aai.onap.org.v11.SitePairSets" />
<xml-element java-attribute="vpnBindings" name="vpn-bindings" type="inventory.aai.onap.org.v11.VpnBindings" />
<xml-element java-attribute="vplsPes" name="vpls-pes" type="inventory.aai.onap.org.v11.VplsPes" />
<xml-element java-attribute="multicastConfigurations" name="multicast-configurations" type="inventory.aai.onap.org.v11.
MulticastConfigurations" />
<xml-element java-attribute="vces" name="vces" type="inventory.aai.onap.org.v11.Vces" />
<xml-element java-attribute="vpes" name="vpes" type="inventory.aai.onap.org.v11.Vpes" />
<xml-element java-attribute="vnfcs" name="vnfcs" type="inventory.aai.onap.org.v11.Vnfcs" />
<xml-element java-attribute="l3Networks" name="l3-networks" type="inventory.aai.onap.org.v11.L3Networks" />
<xml-element java-attribute="networkPolicies" name="network-policies" type="inventory.aai.onap.org.v11.NetworkPolicies" />
<xml-element java-attribute="genericVnfs" name="generic-vnfs" type="inventory.aai.onap.org.v11.GenericVnfs" />
<xml-element java-attribute="lagLinks" name="lag-links" type="inventory.aai.onap.org.v11.LagLinks" />
<xml-element java-attribute="newvces" name="newvces" type="inventory.aai.onap.org.v11.Newvces" />
<xml-element java-attribute="pnfs" name="pnfs" type="inventory.aai.onap.org.v11.Pnfs" />
<xml-element java-attribute="physicalLinks" name="physical-links" type="inventory.aai.onap.org.v11.PhysicalLinks" />
<xml-element java-attribute="newWidgets" name="new-widgets" type="inventory.aai.onap.org.v11.NewWidgets" />
<xml-element java-attribute="ipsecConfigurations" name="ipsec-configurations" type="inventory.aai.onap.org.v11.
IpsecConfigurations" />
<xml-element java-attribute="routeTableReferences" name="route-table-references" type="inventory.aai.onap.org.v11.
RouteTableReferences" />
<xml-element java-attribute="instanceGroups" name="instance-groups" type="inventory.aai.onap.org.v11.InstanceGroups" />
<xml-element java-attribute="zones" name="zones" type="inventory.aai.onap.org.v11.Zones" />
</java-attributes>
</java-type>
```

18. Set up the "NewWidgets" java type:

```
<java-type name="NewWidgets">
<xml-properties>
<xml-property name="description" value="Collection of new Widgets" />
</xml-properties>
<xml-root-element name="new-widgets" />
<java-attributes>
<xml-element container-type="java.util.ArrayList" java-attribute="newWidget" name="new-widget" type="inventory.aai.onap.org.
v11.NewWidget" />
</java-attributes>
</java-type>
```

19. Set up the NewWidget java type:

```
<java-type name="NewWidget">
<xml-root-element name="new-widget" />
<java-attributes>
<xml-element java-attribute="newWidgetName" name="new-widget-name" required="true" type="java.lang.String" xml-key="true">
<xml-properties>
<xml-property name="description" value="e.g.,awesome-new-widget, terrific-new-widget" />
</xml-properties>
</xml-element>
<xml-element java-attribute="newWidgetType" name="new-widget-type" required="true" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="Type of new Widget, e.g., fantastic, amazing" />
</xml-properties>
</xml-element>
<xml-element java-attribute="resourceVersion" name="resource-version" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="Used for optimistic concurrency. Must be empty on create, valid on update and delete."
/>
</xml-properties>
</xml-element>
<xml-element java-attribute="modelInvariantId" name="model-invariant-id" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="the ASDC model id for this resource or service model." />
<xml-property name="visibility" value="deployment" />
<xml-property name="requires" value="model-version-id" />
```

```
<xml-property name="dbAlias" value="model-invariant-id-local" />
</xml-properties>
</xml-element>
<xml-element java-attribute="modelVersionId" name="model-version-id" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="the ASDC model version for this resource or service model." />
<xml-property name="visibility" value="deployment" />
<xml-property name="requires" value="model-invariant-id" />
<xml-property name="dbAlias" value="model-version-id-local" />
</xml-properties>
</xml-element>
<xml-element java-attribute="newWidgetId" name="new-widget-id" type="java.lang.String">
<xml-properties>
<xml-property name="description" value="ID of the newWidget" />
</xml-properties>
</xml-element>
<xml-element java-attribute="relationshipList" name="relationship-list" type="inventory.aai.onap.org.v11.RelationshipList" />
</java-attributes>
<xml-properties>
<xml-property name="description" value="New widgets are the best widgets. Our new widgets are really, really great."/>
<xml-property name="indexedProps" value="new-widget-id,new-widget-name,model-invariant-id,model-version-id" />
<xml-property name="uniqueProps" value="new-widget-id" />
<xml-property name="container" value="new-widgets" />
<xml-property name="namespace" value="network" />
<xml-property name="searchable" value="new-widget-name,new-widget-id" />
</xml-properties>
</java-type>
```

20. Save the file, and rebuild the libraries and microservices:

    here's my example file: aai_oxm_v11.xml
    a. Rebuild aai-common first:
    b. $ cd ~/LF/AAI/aai-common
    c. $ mvn versions:set -DnewVersion=0.0.1-TEST-SNAPSHOT
    d. $ mvn clean install
       Should result in BUILD SUCCESS
    e. $ cd ~/LF/AAI/resources
    f. $ mvn clean install -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
       Should result in BUILD SUCCESS
    g. $ cd ~/LF/AAI/traversal
       mvn clean install -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
    h. Should result in BUILD SUCCESS
21. Run GenTester, using the target dir under aai-resources:

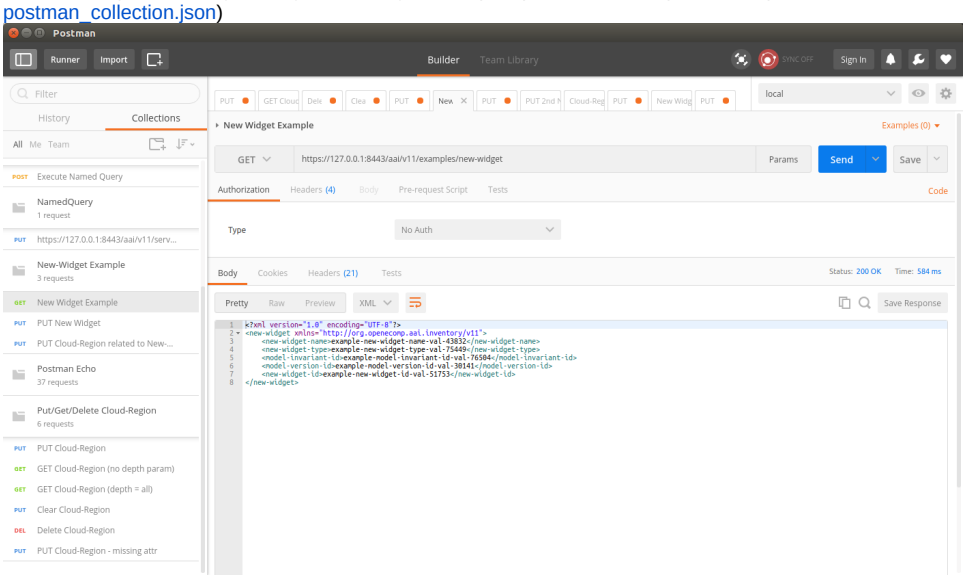
```
$ cd ~/LF/AAI/graphadmin;

java -DAJSC_HOME=. -DBUNDLECONFIG_DIR=src/main/resources/ -Dloader.main=org.onap.aai.schema.GenTester -Dloader.path=.
/src/main/resources -Dschema.ingest.file=src/main/resources/application.properties -jar target/aai-graphadmin-*.jar

You should see the following output:

   ---- NOTE --- about to open graph (takes a little while)--------;

-- loading schema into JanusGraph

-- loading schema into JanusGraph

-- Loading new schema elements into JanusGraph --

-- graph commit

-- graph shutdown

You can check the logs to see if the graph elements were created successfully and grep for your propertycd logs
/createDBSchema/;
grep 'new-widget-name' metrics.log

Creating PropertyKey: [new-widget-name], [String], [SINGLE]
[DEV: 2017-Jul-28 08:44:12,287][INFO ][main ]Creating PropertyKey: [new-widget-name], [String], [SINGLE]
Add index for PropertyKey: [new-widget-name]
[DEV: 2017-Jul-28 08:44:12,289][INFO ][main ]Add index for PropertyKey: [new-widget-name]
Creating PropertyKey: [new-widget-type], [String], [SINGLE]
[DEV: 2017-Jul-28 08:44:12,291][INFO ][main ]Creating PropertyKey: [new-widget-type], [String], [SINGLE]
No index added for PropertyKey: [new-widget-type]
[DEV: 2017-Jul-28 08:44:12,501][INFO ][main ]No index added for PropertyKey: [new-widget-type]
PropertyKey [resource-version] already existed in the DB.
PropertyKey [model-invariant-id] already existed in the DB.
PropertyKey [model-version-id] already existed in the DB.
```

Creating PropertyKey: [new-widget-id], [String], [SINGLE]
[DEV: 2017-Jul-28 08:44:12,501][INFO][main]Creating PropertyKey: [new-widget-id], [String], [SINGLE]

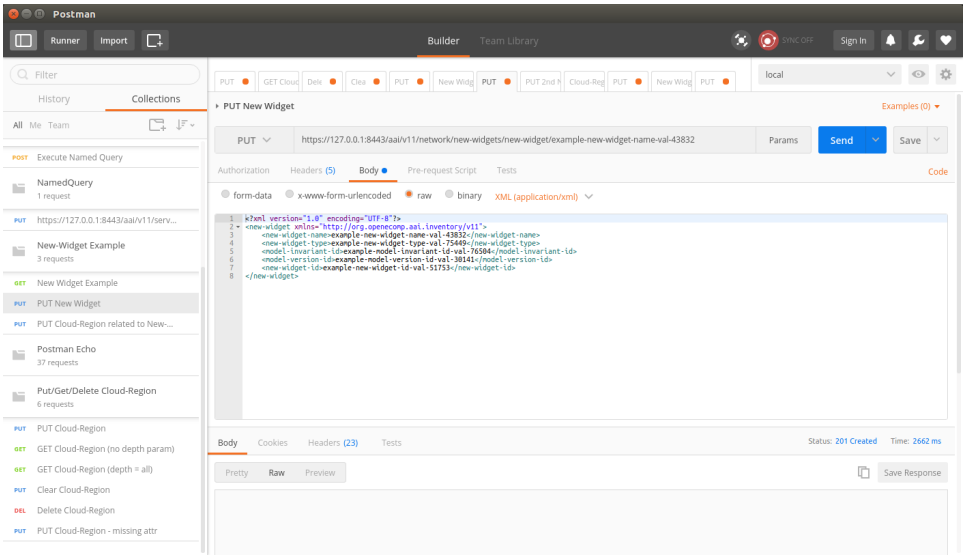
22. Start the "resources" microservice
    a. Resources runs on port 8447. Go to the resources directory
       $ cd ~/LF/AAI/resources
    b. Set the debug port to 9447
       $ export MAVEN_OPTS="-Xms1024m -Xmx5120m -XX:PermSize=2024m -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=9447,server=y,suspend=n"
    c. Start the microservice
       $ mvn -pl aai-resources -P runAjsc -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
       Should see something like this: Resources Microservice Started

23. Start the "traversal" microservice
    a. Traversal runs on port 8446. Go to the traversal directory
       $ cd ~/LF/AAI/traversal
    b. Set the debug port to 9446
       $ export MAVEN_OPTS="-Xms1024m -Xmx5120m -XX:PermSize=2024m -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:transport=dt_socket,address=9447,server=y,suspend=n"
    c. Start the microservice
       $ mvn -pl aai-traversal -P runAjsc -Daai.core.version=0.0.1-TEST-SNAPSHOT -Daai.schema.version=0.0.1-TEST-SNAPSHOT
       Should see something like this: Traversal Microservice Started

24. Get an example newWidget using "New Widget Example" (find these examples in this postman collection: New-Widget Example.postman_collection.json)

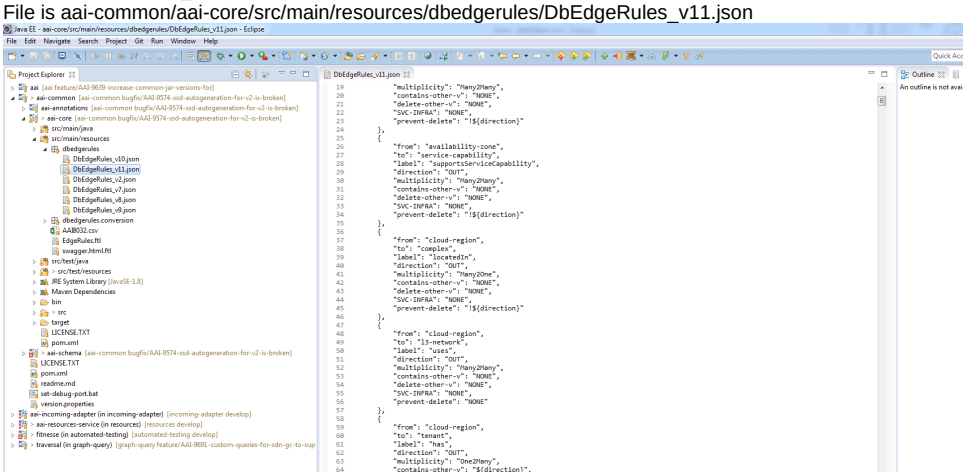


25. Copy the new widget and PUT it: (PUT New Widget)



26. Success! We have updated an existing object type and added a new object type. Next, we will set up an edge between the CloudRegion and the NewWidget

27. Open DbEdgeRules_v11.json
    File is aai-common/aai-core/src/main/resources/dbedgerules/DbEdgeRules_v11.json



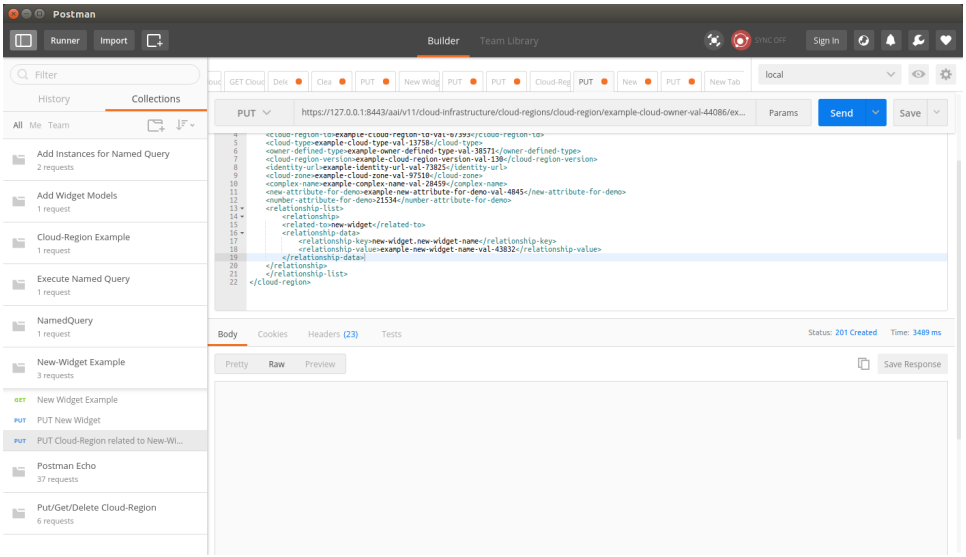
28. Add a new EdgeRule, allowing an edge between the CloudRegion and NewWidget:

```
    {
        "from": "cloud-region",
        "to": "tenant",
        "label": "has",
        "direction": "OUT",
        "multiplicity": "One2Many",
        "contains-other-v": "${direction}",
        "delete-other-v": "NONE",
        "SVC-INFRA": "!${direction}",
        "prevent-delete": "${direction}"
    },
    {
        "from": "cloud-region",
        "to": "new-widget",
        "label": "has",
        "direction": "OUT",
        "multiplicity": "One2Many",
        "contains-other-v": "NONE",
        "delete-other-v": "NONE",
        "SVC-INFRA": "NONE",
        "prevent-delete": "NONE"
    },
```

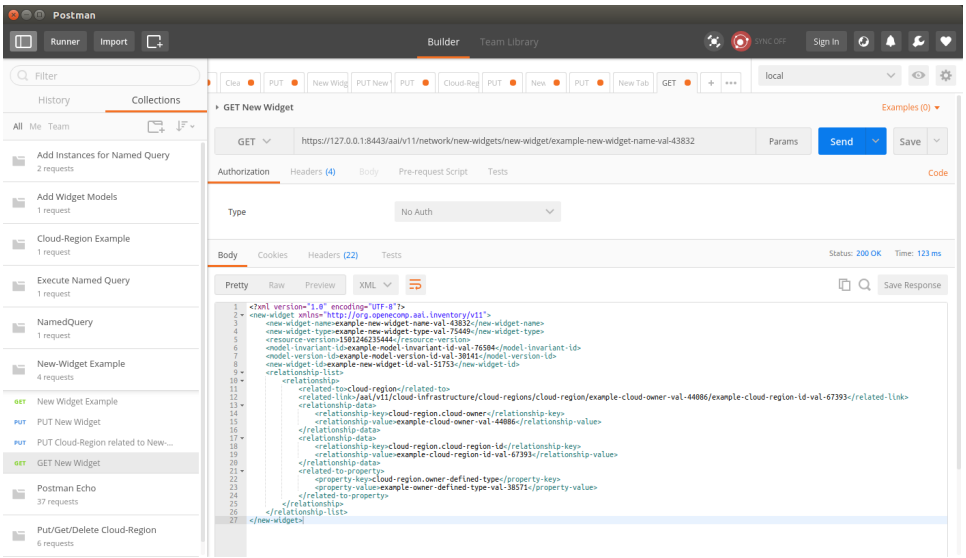
This means cloud-region connects to new-widget, edge label is "has", it's an OUT edge, one cloud region can have multiple edges to new-widgets,
cloud-region does not contain new-widget, new-widget will not be deleted when the cloud-region connected to it is deleted, it is not SVC_INFRA type, and having an edge to a new-widget will not prevent deletion of the cloud-region.

29. Save the file, and rebuild aai-common, resources, and traversal. Start resources and traversal microservices.

30. Create a cloud region with a relationship to the new-widget that was created earlier: (PUT Cloud-Region related to New-Widget)



31. GET the new-widget object: (GET New Widget)



32. Notice that when GETting the new-widget object it shows the relationship to the cloud-region.

    We have successfully modified the schema and edge rules to update an existing type, create a new type, and define an edge relationship which allows for a new-widget type to be connected to a cloud-region.

Attachments

File	Modified
PNG File image2017-7-26_16-58-5.png	Jul 27, 2017 by James Forsyth
File Execute Named Query.postman_collection.json	Jul 27, 2017 by James Forsyth
File Add Instances for Named Query.postman_collection.json	Jul 27, 2017 by James Forsyth
File NamedQuery.postman_collection.json	Jul 27, 2017 by James Forsyth
File Add Widget Models.postman_collection.json	Jul 27, 2017 by James Forsyth
PNG File image2017-7-26_16-23-12.png	Jul 27, 2017 by James Forsyth

PNG File image2017-7-26_16-17-19.png	Jul 27, 2017 by James Forsyth
File models.csv	Jul 27, 2017 by James Forsyth
PNG File image2017-7-26_11-6-11.png	Jul 27, 2017 by James Forsyth
File aai.pem	Jul 27, 2017 by James Forsyth
File haproxy.cfg	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_12-31-54.png	Jul 27, 2017 by James Forsyth
File Cloud-Region Example.postman_collection.json	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_13-57-5.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-6-25.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-6-53.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-7-35.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-7-48.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-11-36.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-15-23.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_14-19-20.png	Jul 27, 2017 by James Forsyth
File Put-Get Cloud-Region.postman_collection.json	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_15-4-5.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_15-5-19.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_15-8-17.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-27_15-14-13.png	Jul 27, 2017 by James Forsyth
PNG File image2017-7-28_8-26-35.png	Jul 28, 2017 by James Forsyth
PNG File image2017-7-28_8-27-41.png	Jul 28, 2017 by James Forsyth
XML File aai_oxm_v11.xml	Jul 28, 2017 by James Forsyth
PNG File image2017-7-28_9-18-56.png	Jul 28, 2017 by James Forsyth
PNG File image2017-7-28_9-20-43.png	Jul 28, 2017 by James Forsyth
PNG File edgeruleScreenshot.png	Aug 21, 2017 by Jane Threefoot
File New-Widget Example.postman_collection.json	Aug 31, 2017 by James Forsyth
PNG File image2017-9-4_11-4-45.png	Sep 04, 2017 by Thamlur Raju
PNG File 1.PNG	Dec 07, 2018 by Alex Hooper
PNG File 2.PNG	Dec 07, 2018 by Alex Hooper
PNG File cr1.PNG	Dec 07, 2018 by Alex Hooper
File l3vpn-service-model.yang	Dec 07, 2018 by Alex Hooper
PNG File l3vpnview.PNG	Dec 07, 2018 by Alex Hooper

Download All