XACML Authorization for IRIS

IRIS Framework uses the Temenos security framework library to apply Attribute-based Access policies check at IRIS API endpoint (GET/POST/PUT) before the business function gets executed. The XACML Policies are available in the runtime and enforced before APIs getting executed in the T24.

Policy Decision Policy (PDP) Engine provides the infrastructure to evaluate applicable policy and returns an authorization decision to the application to enforce the decision at the right enforcement point, this helps IRIS to take an authorization decision at the API layer than that of the business layer.

To strengthen the security in IRIS, authorization through **XACML has been enabled by default from 202010 build onwards** for the IRIS provider war (**irf-provider-container** and etc.,) as well as for the IRF archetype container. But there is an option to disable the same by following the steps defined at disabling XACML.

Setup:

1. Make sure the IRIS version is on or above 202004.
2. Build the IRIS.war file
3. Generate the policy XML files based on the user needs for the API endpoints, and add them in the <IRIS.WAR>/WEB-INF/classes/xacml directory. (Make sure to create **"resourceType"** as "***API***", and "**resourceManagerId**" as "***IRIS***")
4. Open the war file and navigate to <IRIS.WAR>/WEB-INF/classes/applicationContext.xml
5. From 202010 onwards the XACML feature has been enabled by default
   Note: Uncomment the following XML bean to enable the XACML feature in IRIS for older releases from **202004** to **202009**

```
<!-- Enable this bean for XACML -->
<bean id="irisProcessorAspect" class="com.temenos.irf.core.xacml.authz.IrisProcessorXacmlAspect" />
```

6. **XACML configuration steps:-**

Sample steps for an API.

Root-policy.xml

Define policySet

e.g. <PolicySetIdReference>CreateReservationPolicySet</PolicySetIdReference>

Above PolicySetId will be linked in policy.xml(e.g. iris-admin-createReservation-policy.xml)

pdp-config.xml

Define policy location

e.g. <policyLocation>classpath:xacml/iris-admin-createReservation-policy.xml</policyLocation>

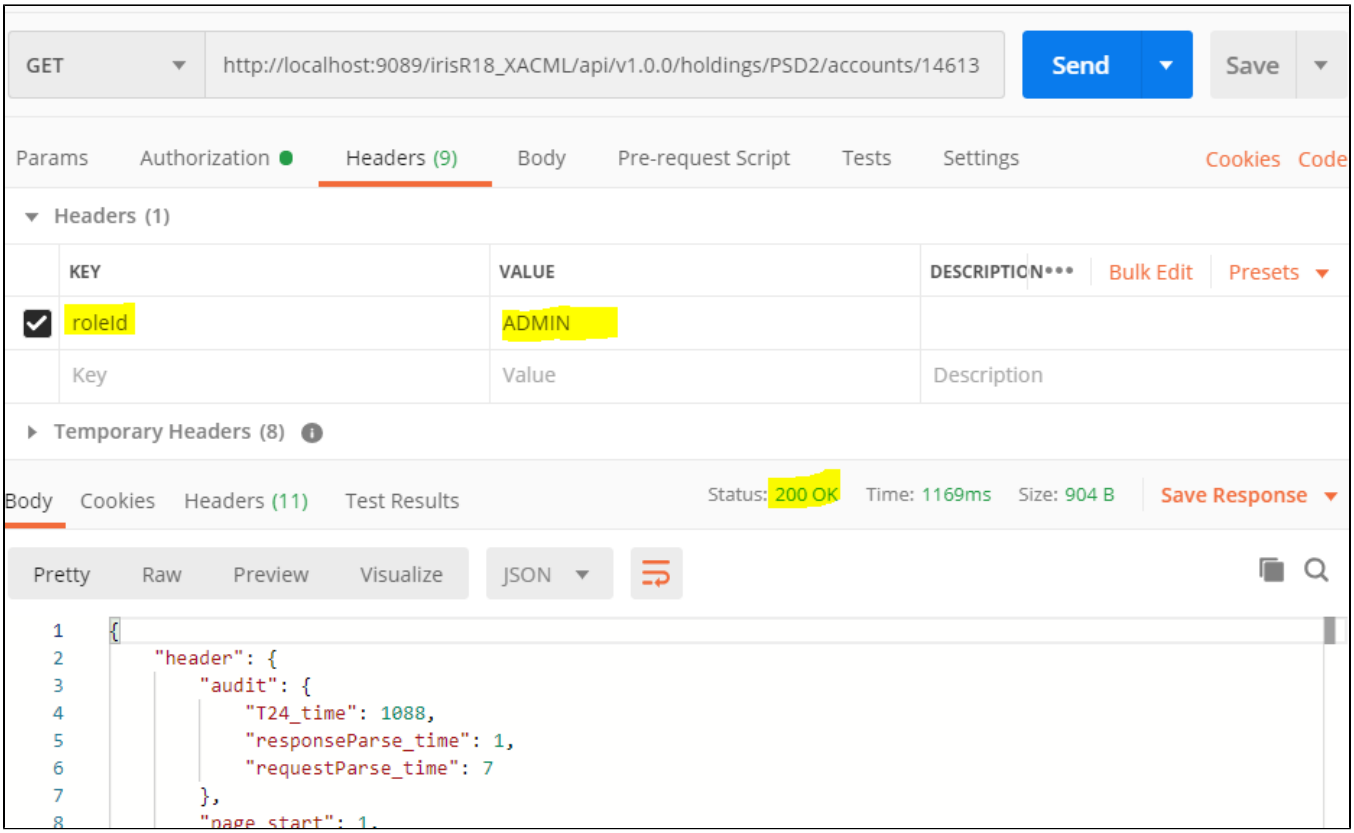
iris-admin-createReservation-policy.xml

If the "resourceId" is defined as "create{companyCode}-arrangements " in the policy xml(iris-admin-createReservation-policy.xml), then make sure this specific resource is mapped with the "Id" tag of your service xml, refer the below screen shot for reference.

7. Save the changes and deploy your war file.

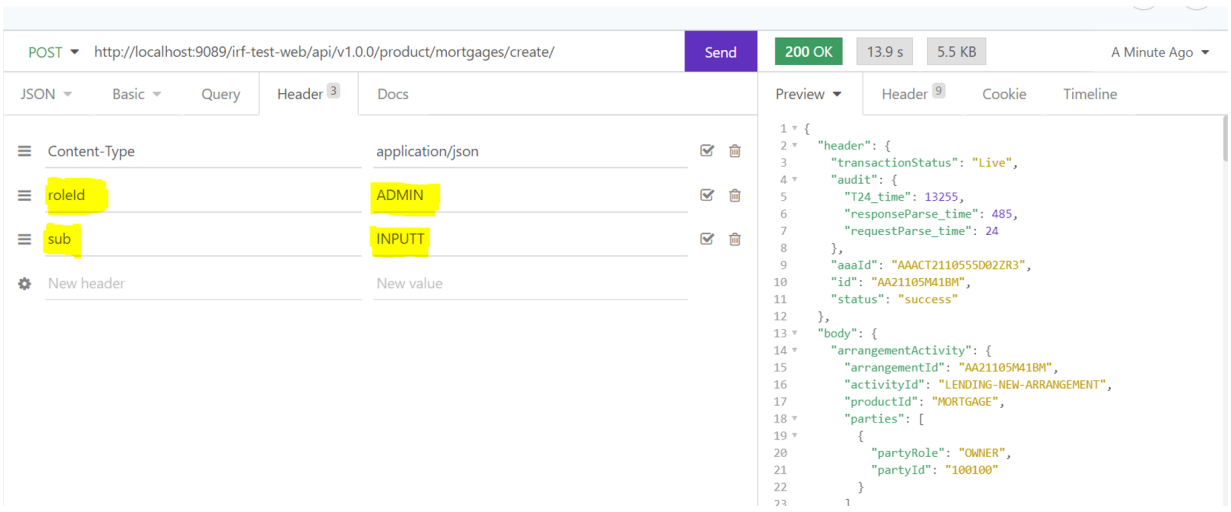
8. Provide the “**roleId**” in the HTTP header and test your API.

Example Screen Shot:



9. Provide "**roleId**" and "**sub**" in HTTP headers in case of validating XACML with Basic auth and not JWT involved.

Example screenshot below,



NOTE:

1. If you are using **UTP-PACK,** the default configuration might not work, the reason is that in **UTP_PACK** is added with an environmental flag **-DPDP_CONFIG=classpath:authzforce/medium-pdp-config.xml ,** so to make this xacml to work on, do any one of the following changes.

1. either Remove this flag from the environmental variables.

2. OR rename the folder and files in the war files from **"xacml" to "authzforce" and "pdp-config.xml" to "pdp-medium-confg.xml".**

2. while generating the api endpoint "resourceId" in the policy xml, make sure the the same id is present in the service xml, if your service xml doesn't have an "Id" element in the rest route, then it wont work, refer below the screen print.

a. If the "resourceId" is defined as "getApis" in the policy xml, then make sure this specific resource is mapped with the "Id" tag of your service xml, refer the below screen shot for reference.

Disabling XACML

To disable the XACML security comment the below XML bean at <IRIS.WAR>/WEB-INF/classes/applicationContext.xml