# Cookie-based valve [Technical Documentation]

# Cookie-based valve:

* The cookie-based valve is a custom valve to authenticate users by integrates with a validation service with a custom cookie to validate a user and respond by the user name and status
* It’s a custom valve for MOMRA

# Cookie-based valve workflow:

**1-** Firstly, the user needs to set the valve at the server.xml as following:

<Valve

LoggingEnabled="true"

className="com.incorta.sso.valves.CookieBasedValve"

cookieName="{Authentication cookie name}"

validationServiceURL= "{the validation service URL}"

validationServiceUserName="{Incorta userName, to access the service}"

validationServicePassword="{Incorta password, Base64 encoded}"

/>

actual example for MOMRA customer with

* **Validation Service URL**: <https://pegadev.momra.gov.sa/prweb/PRRestService/Incorta/v1/Authentication>
* **Validation Service UserName**: IncortaUser
* **Validation Service Password**: 6+g4^!S+V`+qXpQ

The valve will be:

<Valve

LoggingEnabled="true"

className="com.incorta.sso.valves.CookieBasedValve"

cookieName="RequestorID"

validationServiceURL="https://pegadev.momra.gov.sa/prweb/PRRestService/Incorta/v1/Authentication"

validationServiceUserName="IncortaUser"

validationServicePassword="PDYrZzReIVMrVmArcVhwUQ=="

/>

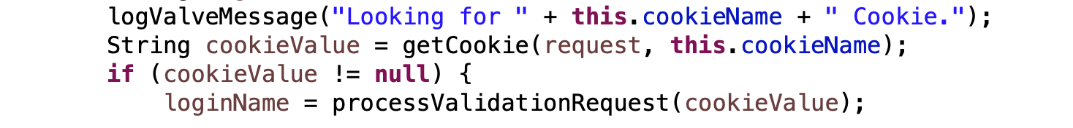
**Note:**  
We decided that the password should be Base64 encoded, because if the password contains any special characters this may cause parsing issue in the XML

**2-** When the user calls Incorta with this URL pattern

https://<incorta-server>/incorta/!<tenant-name>/, with the **cookie** specified at the valve, the Cookie-based valve will handle it.

**3-** The Cookie-based valve will look for a cookie with the name specified at the valve.

if exits the valve will call the validation service



**4-** The valve will call the validation service URL specified at the valve with a basic authorization header  
The request body contrat

{

"RequestData":{

"RequestorID": {userId}

}

}



**5-** We expect the following JSON response from the validation service

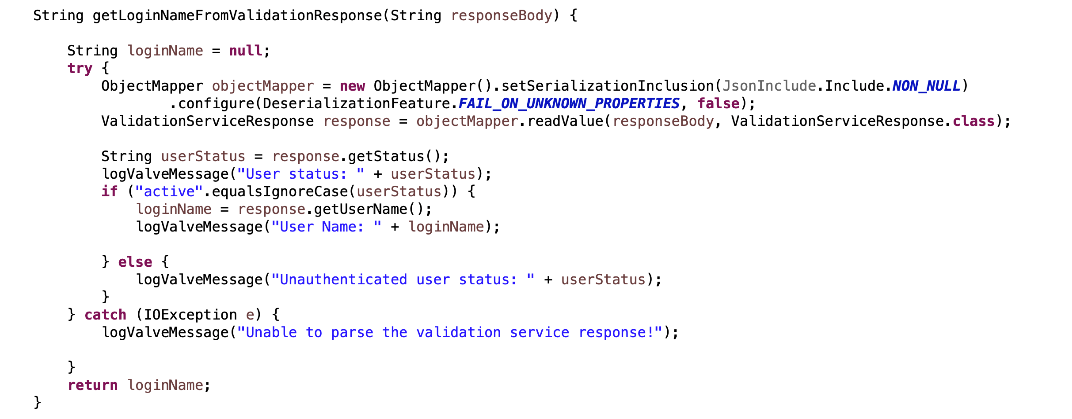
{

"ID":{loginName},

"Status": [Active| Inactive | Invalid]

}

Then we parse this response to get the loginName if the status is Active



**6-** Then we set the loginName in a principle and redirect the request to Incorta



**7-** We check at the beginning of the valve if the request already has a context, we redirect it to the Incorta directly

