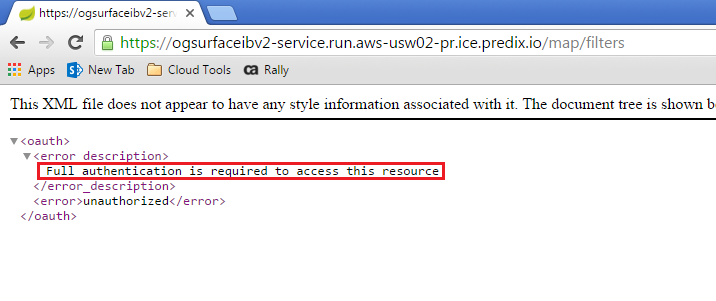
Predix 2.0 UAA validation for REST service

|  |  |
| --- | --- |
| Published Date: | 29-July-2016 |
| Version: | 1.0 |
| Author: | Komal/Sai |

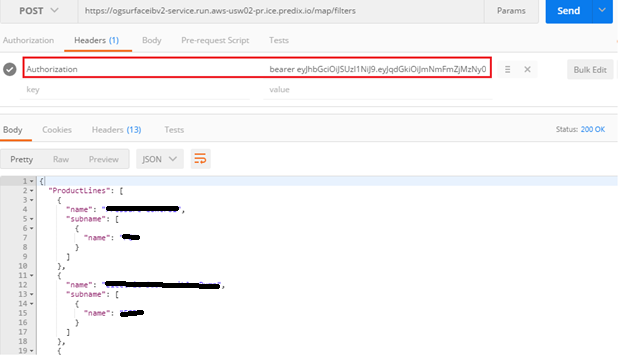
## UAA Validation for REST Services

**Go to your service URL (REST) from browser**

****

5

In the above example the service URL is protected and will not be accessible to any user unless it passes token

Below screen shot shows how to access our secured URL using passing the token in request header which will be time bounded.

**Step for implementation**

For Spring Boot 1.3.0 application below changes needs to be done for UAA validation.

**Changes required in files**

1. Pom.xml for adding dependencies
2. Application.java for enabling resource server
3. application.properties for UAA configuration information
4. **Addition of the following dependency in pom.xml**

<dependency>

<groupId>org.springframework.security.oauth</groupId>

<artifactId>spring-security-oauth2</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-data</artifactId>

</dependency>

1. **Application.java for enabling resource server**

* Add the @EnableResourceServer on Application class
* Add a configure method

**public** **void** configure(ResourceServerSecurityConfigurer resources)

**throws** Exception {

resources.resourceId("openid");

}

**3) Addition of the following parameters in application.properties or (application yml)**

security.oauth2.client.clientId=<client ID>

security.oauth2.client.clientSecret=<clientSecret>

security.oauth2.client.accessTokenUri=<UAA URL>/oauth/access\_token

security.oauth2.client.userAuthorizationUri=<UAA URL>/oauth/authorize

security.oauth2.resource.userInfoUri=<UAA URL>/userinfo

**UAA changes in nginx file**

1. Add the client\_id, uaa\_authorization\_header and user\_token variables inside server tag and set the environment variables.

server

{

set $client\_id <%= ENV["clientId"] %>;

set $uaa\_authorization\_header 'Basic <%= ENV["uaa\_authorization\_header"] %>';

set $user\_token '';

Addition of Authorization header for all URL actions

#Masking the URL For the Map service.

location /connectService{

access\_by\_lua\_file <%= ENV["APP\_ROOT"] %>/set\_access\_token.lua;

proxy\_set\_header Authorization $user\_token;

rewrite /connectService/(.\*) /$1 break;

proxy\_pass https://ogib-service-dev-v2-service.run.aws-usw02-pr.ice.predix.io/;

}

Refer the below nginx.conf file for reference

