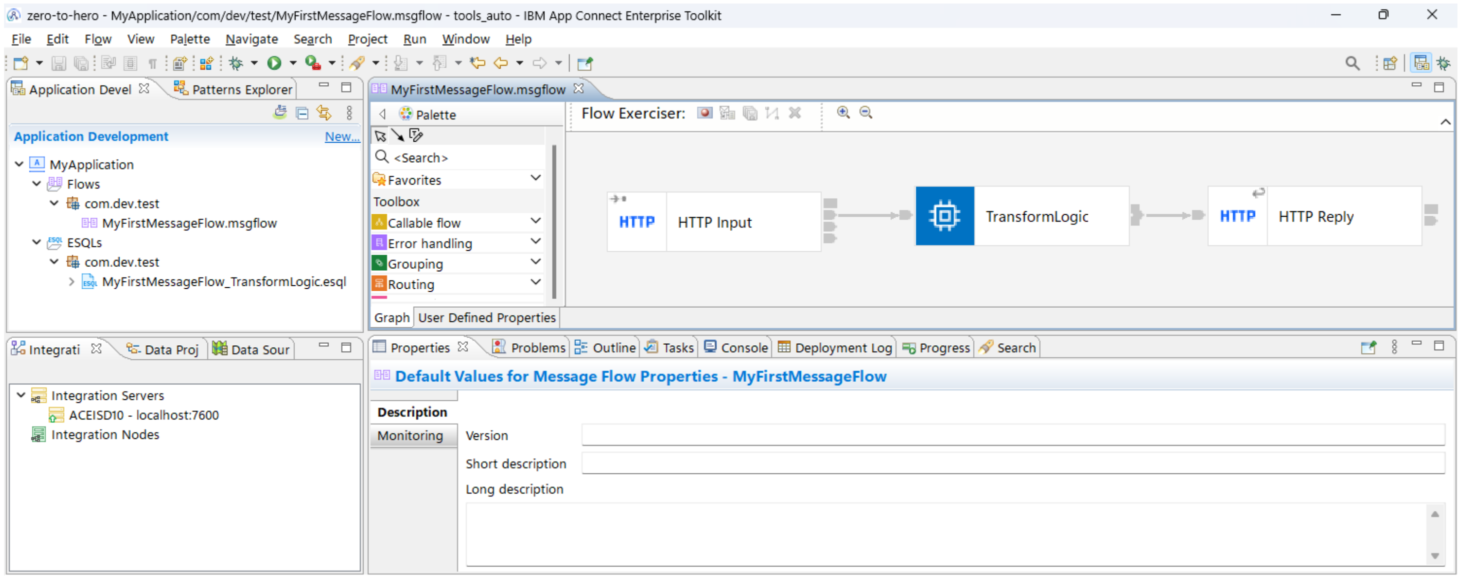
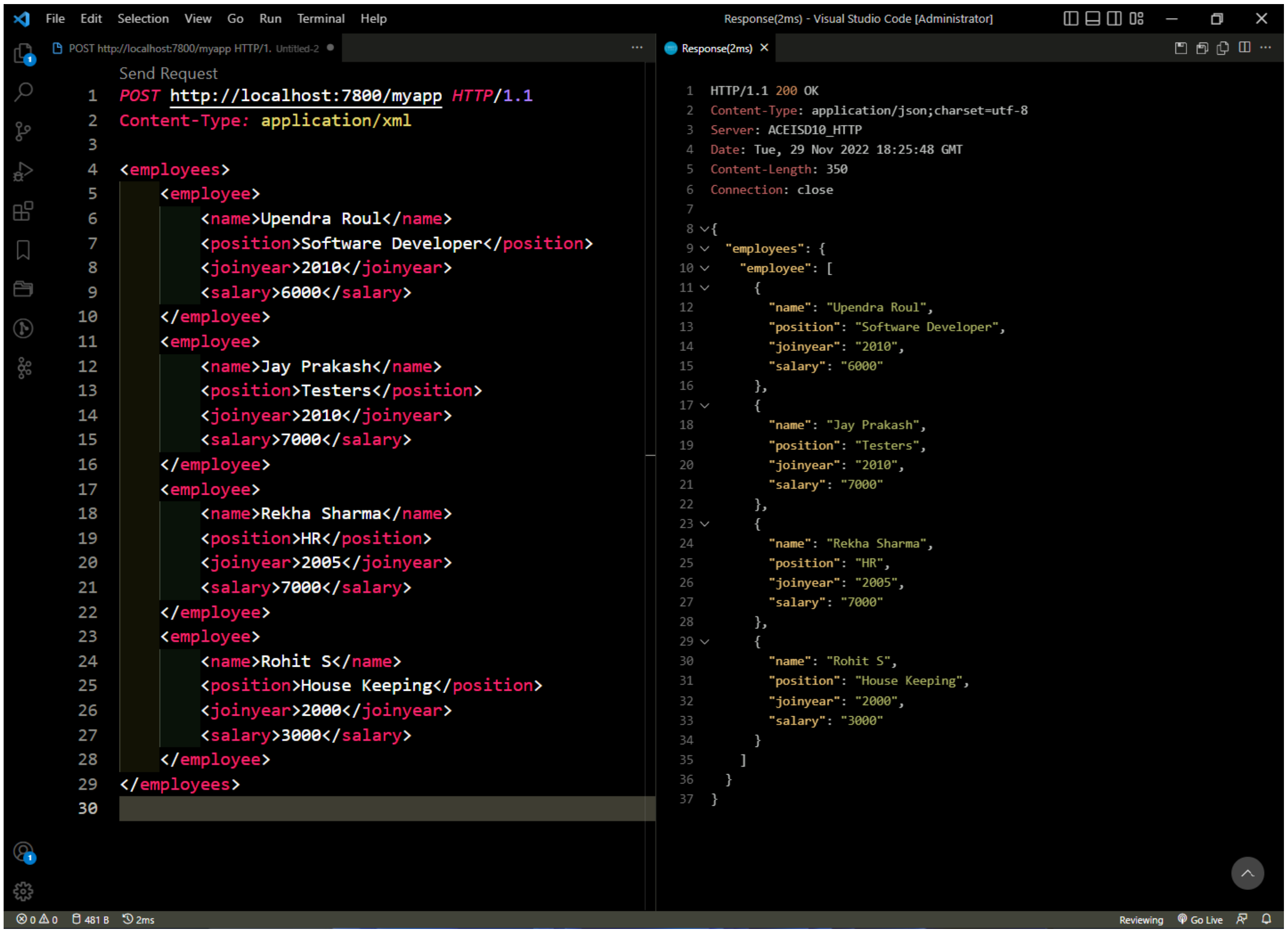
How to secure an HTTP Service in IBM ACE and using Basic Auth

This document demonstrates the capability of securing an HTTP(s) service using Basic Auth. Although Basic Auth is not generally used in production environments (securing the APIs is majorly handled by API Management Tools like IBM API Connect), it is worth learning.

Step 1: Creating a simple HTTP Service



The service transforms an XML to JSON. As shown, the service works without Authorization.


File Edit Selection View Go Run Terminal Help
Response(2ms) - Visual Studio Code [Administrator]
POST http://localhost:7800/myapp HTTP/1. Untitled-2
Response(2ms)
Send Request
0 0
481 B
2ms
Reviewing
Go Live

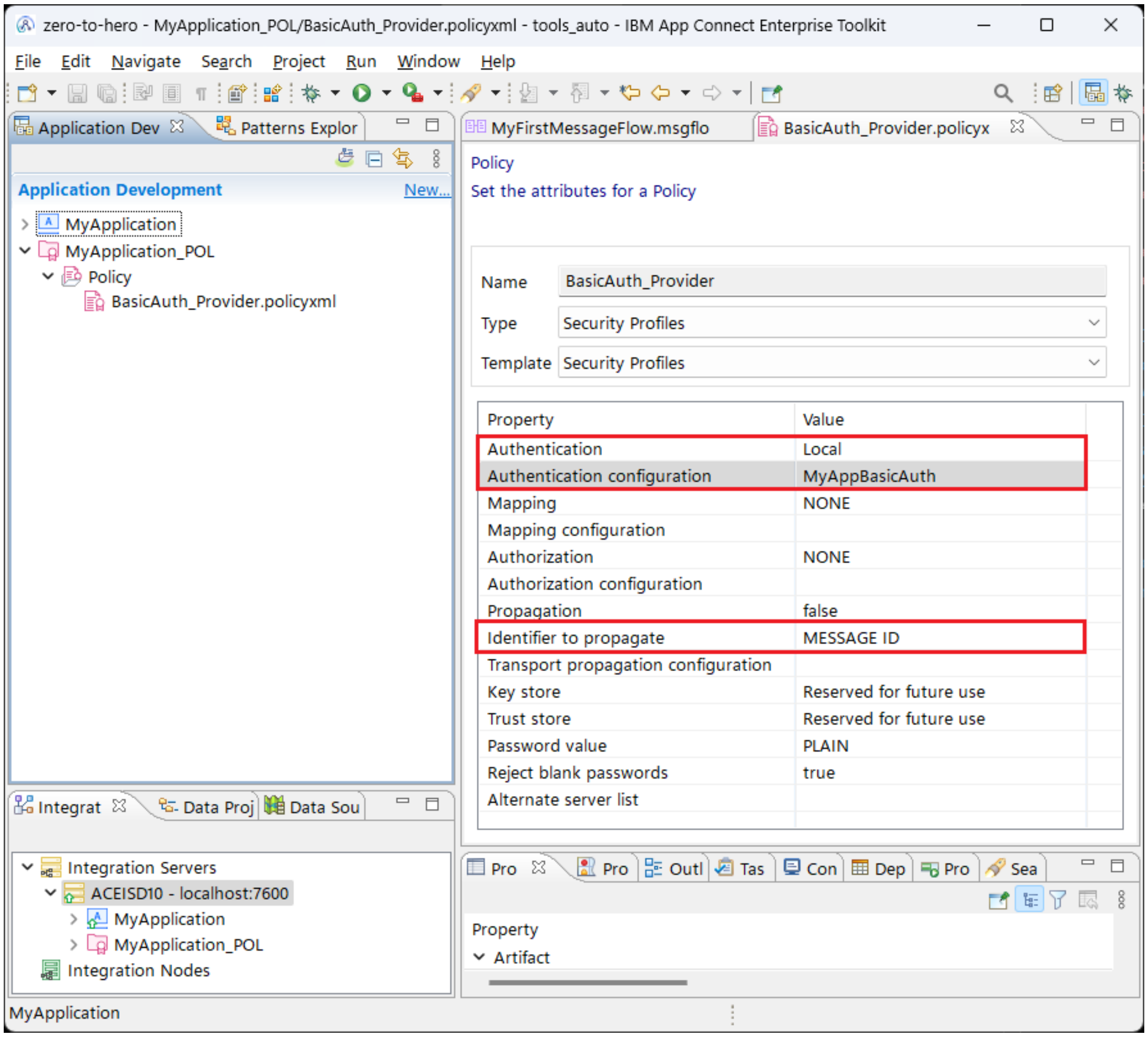
```
POST http://localhost:7800/myapp HTTP/1.1
Content-Type: application/xml

<employees>
    <employee>
        <name>Upendra Roul</name>
        <position>Software Developer</position>
        <joinyear>2010</joinyear>
        <salary>6000</salary>
    </employee>
    <employee>
        <name>Jay Prakash</name>
        <position>Testers</position>
        <joinyear>2010</joinyear>
        <salary>7000</salary>
    </employee>
    <employee>
        <name>Rekha Sharma</name>
        <position>HR</position>
        <joinyear>2005</joinyear>
        <salary>7000</salary>
    </employee>
    <employee>
        <name>Rohit S</name>
        <position>House Keeping</position>
        <joinyear>2000</joinyear>
        <salary>3000</salary>
    </employee>
</employees>
```

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=utf-8
Server: ACEISD10_HTTP
Date: Tue, 29 Nov 2022 18:25:48 GMT
Content-Length: 350
Connection: close

{
  "employees": {
    "employee": [
      {
        "name": "Upendra Roul",
        "position": "Software Developer",
        "joinyear": "2010",
        "salary": "6000"
      },
      {
        "name": "Jay Prakash",
        "position": "Testers",
        "joinyear": "2010",
        "salary": "7000"
      },
      {
        "name": "Rekha Sharma",
        "position": "HR",
        "joinyear": "2005",
        "salary": "7000"
      },
      {
        "name": "Rohit S",
        "position": "House Keeping",
        "joinyear": "2000",
        "salary": "3000"
      }
    ]
  }
}
```

Step 2: Creating a Policy Project which contains the Policy. Notice the Authentication configuration is configured with the value of MyAppBasicAuth. We must set the security credential with the same name. We can use mqsisetdbparms command, IBM ACE's vault or configure credentials in the server.conf.yaml.



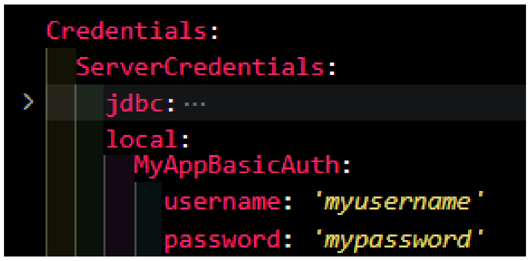
zero-to-hero - MyApplication_POL/BasicAuth_Provider.policyxml - tools_auto - IBM App Connect Enterprise Toolkit

Policy
Set the attributes for a Policy

Name	BasicAuth_Provider
Type	Security Profiles
Template	Security Profiles

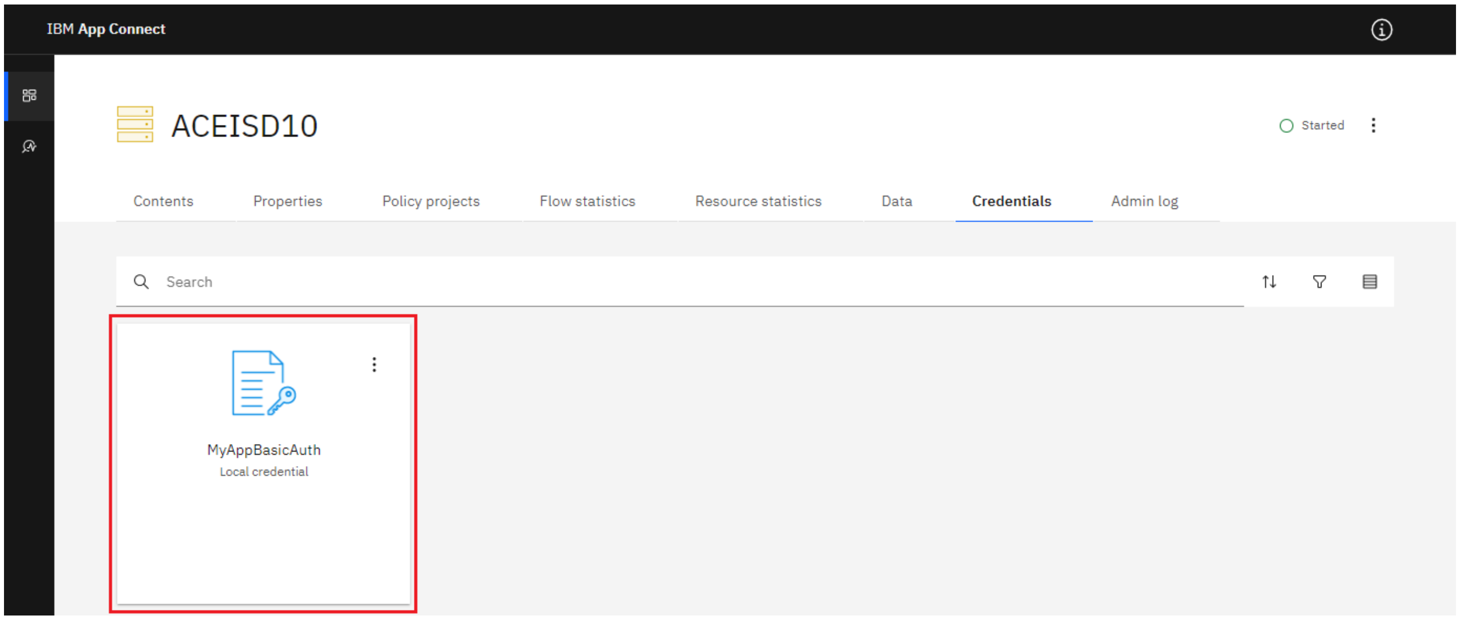
Property	Value
Authentication	Local
Authentication configuration	MyAppBasicAuth
Mapping	NONE
Mapping configuration	
Authorization	NONE
Authorization configuration	
Propagation	false
Identifier to propagate	MESSAGE ID
Transport propagation configuration	
Key store	Reserved for future use
Trust store	Reserved for future use
Password value	PLAIN
Reject blank passwords	true
Alternate server list	

Below snapshot shows the configuration of MyAppBasicAuth in server.conf.yaml. Once configured, restart the Integration Server.

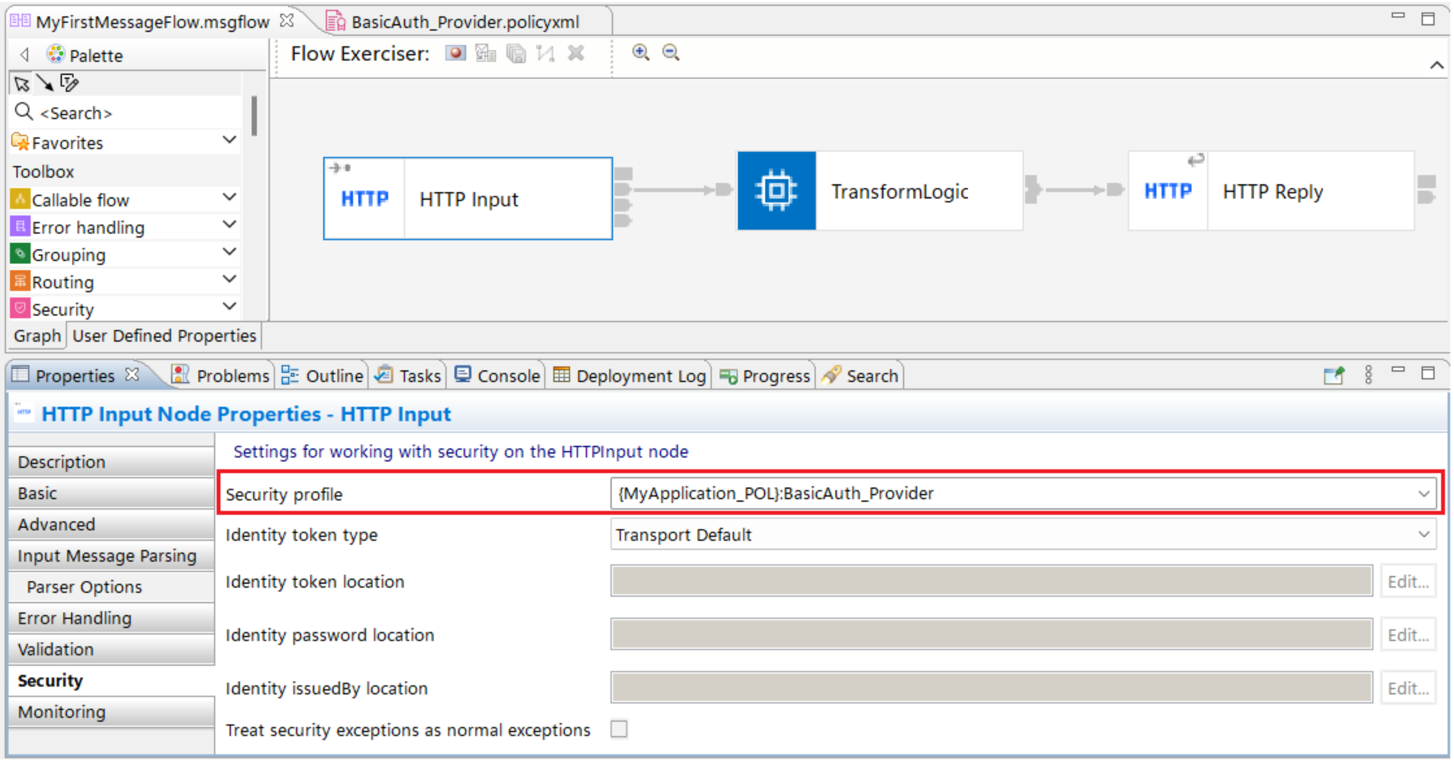


```
Credentials:
  ServerCredentials:
    jdbc: ···
    local:
      MyAppBasicAuth:
        username: 'myusername'
        password: 'mypassword'
```

Step 3: Validate the credential is available or not – I have used WebUI to validate. Once the Credential is available, we can configure the Policy to our Service.



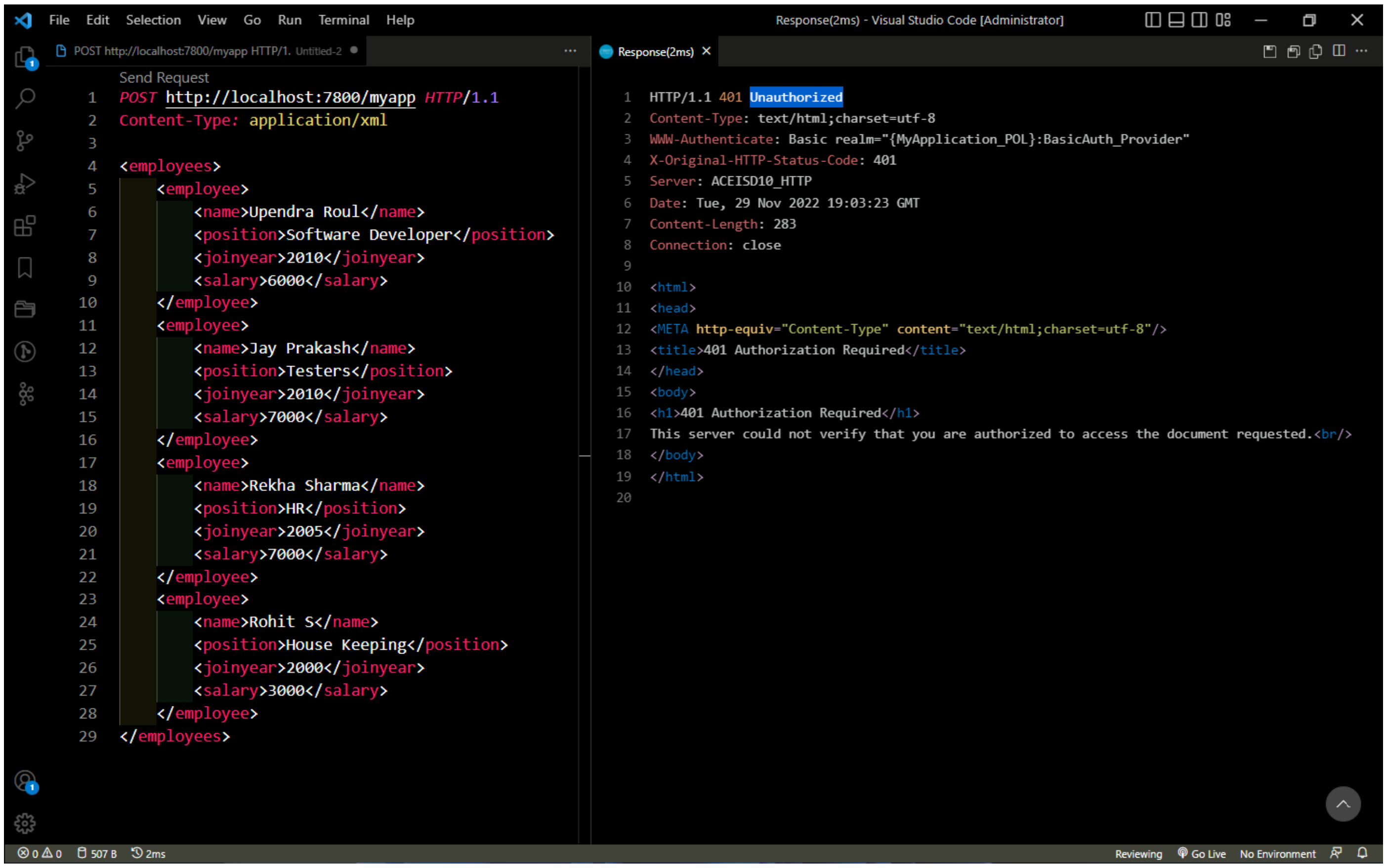
Step 4: In the Application, select the HTTP Input Node, go to Security tab and configure the Security profile as shown. The name must be configured as *{PolicyProject}:PolicyName*. Once the Security profile is configured, re-deploy the application (make sure the Policy is still deployed).



Settings for working with security on the HTTPInput node

Security profile	{MyApplication_POL}:BasicAuth_Provider
Identity token type	Transport Default
Identity token location	
Identity password location	
Identity issuedBy location	
Treat security exceptions as normal exceptions	☐

Step 5: Testing the service – we get HTTP 401 Unauthorized. This means that the service now needs an Authorization.


File Edit Selection View Go Run Terminal Help
Response(2ms) - Visual Studio Code [Administrator]
POST http://localhost:7800/myapp HTTP/1. Untitled-2
Response(2ms)
Send Request
507 B
2ms
Reviewing Go Live No Environment

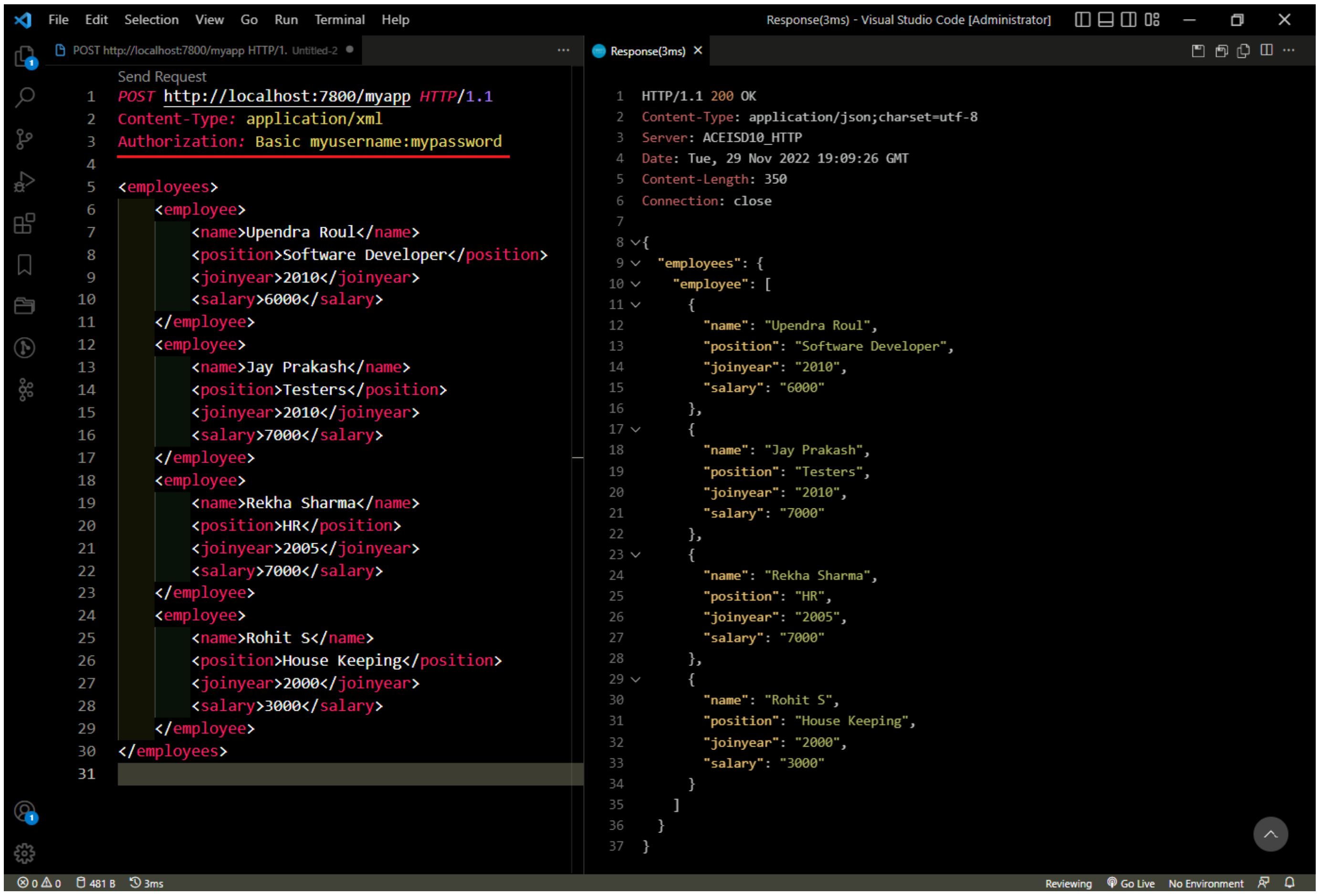
```
POST http://localhost:7800/myapp HTTP/1.1
Content-Type: application/xml

<employees>
    <employee>
        <name>Upendra Roul</name>
        <position>Software Developer</position>
        <joinyear>2010</joinyear>
        <salary>6000</salary>
    </employee>
    <employee>
        <name>Jay Prakash</name>
        <position>Testers</position>
        <joinyear>2010</joinyear>
        <salary>7000</salary>
    </employee>
    <employee>
        <name>Rekha Sharma</name>
        <position>HR</position>
        <joinyear>2005</joinyear>
        <salary>7000</salary>
    </employee>
    <employee>
        <name>Rohit S</name>
        <position>House Keeping</position>
        <joinyear>2000</joinyear>
        <salary>3000</salary>
    </employee>
</employees>
```

```
HTTP/1.1 401 Unauthorized
Content-Type: text/html;charset=utf-8
WWW-Authenticate: Basic realm="{MyApplication_POL}:BasicAuth_Provider"
X-Original-HTTP-Status-Code: 401
Server: ACEISD10_HTTP
Date: Tue, 29 Nov 2022 19:03:23 GMT
Content-Length: 283
Connection: close

<html>
<head>
<META http-equiv="Content-Type" content="text/html;charset=utf-8"/>
<title>401 Authorization Required</title>
</head>
<body>
<h1>401 Authorization Required</h1>
This server could not verify that you are authorized to access the document requested.<br/>
</body>
</html>
```

Step 6: Added the Basic Authorization HTTP Header to invoke the service, the service works as expected.


File Edit Selection View Go Run Terminal Help
Response(3ms) - Visual Studio Code [Administrator]
POST http://localhost:7800/myapp HTTP/1. Untitled-2
Response(3ms)
Send Request
481 B
3ms
Reviewing Go Live No Environment

```
POST http://localhost:7800/myapp HTTP/1.1
Content-Type: application/xml
Authorization: Basic myusername:mypassword

<employees>
    <employee>
        <name>Upendra Roul</name>
        <position>Software Developer</position>
        <joinyear>2010</joinyear>
        <salary>6000</salary>
    </employee>
    <employee>
        <name>Jay Prakash</name>
        <position>Testers</position>
        <joinyear>2010</joinyear>
        <salary>7000</salary>
    </employee>
    <employee>
        <name>Rekha Sharma</name>
        <position>HR</position>
        <joinyear>2005</joinyear>
        <salary>7000</salary>
    </employee>
    <employee>
        <name>Rohit S</name>
        <position>House Keeping</position>
        <joinyear>2000</joinyear>
        <salary>3000</salary>
    </employee>
</employees>
```

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=utf-8
Server: ACEISD10_HTTP
Date: Tue, 29 Nov 2022 19:09:26 GMT
Content-Length: 350
Connection: close

{
  "employees": {
    "employee": [
      {
        "name": "Upendra Roul",
        "position": "Software Developer",
        "joinyear": "2010",
        "salary": "6000"
      },
      {
        "name": "Jay Prakash",
        "position": "Testers",
        "joinyear": "2010",
        "salary": "7000"
      },
      {
        "name": "Rekha Sharma",
        "position": "HR",
        "joinyear": "2005",
        "salary": "7000"
      },
      {
        "name": "Rohit S",
        "position": "House Keeping",
        "joinyear": "2000",
        "salary": "3000"
      }
    ]
  }
}
```