# HTTP

Hypertext Transfer Protocol (HTTP) is an Internet protocol that is used to transfer and display hypertext and XML documents on the Web.

You can configure message flows that include the HTTP or SOAP nodes to access the HTTP transport to work with the following resources:

* SOAP-based Web services
* Other Web services standards, such as REST
* General HTTP messaging, where the payload might be XML

HTTP nodes can process non-secure (HTTP) messages and secure (HTTPS or HTTP over SSL) messages.

You need to configure the ssl to the broker to work with HTTPS urls.

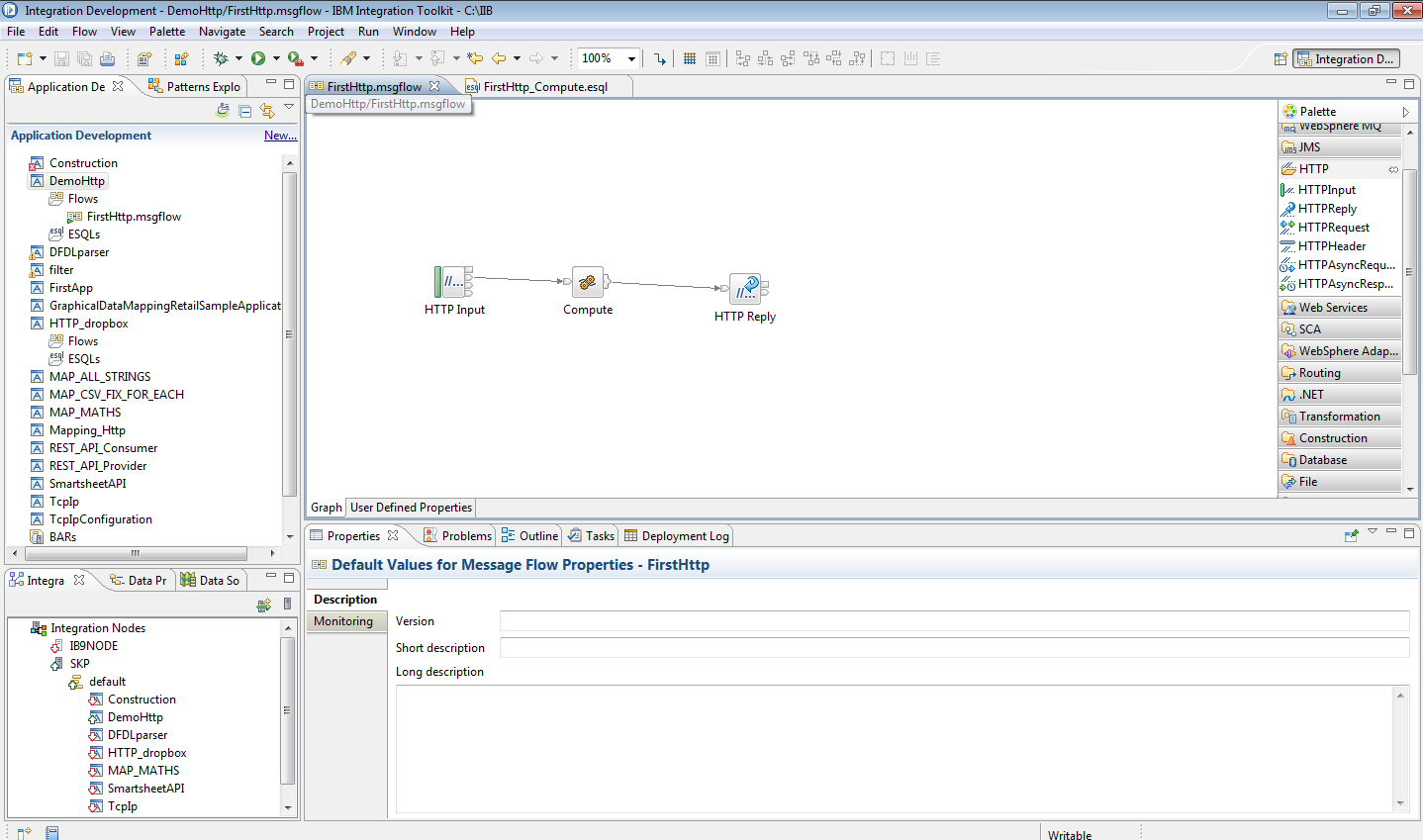
Here assuming you know how to create a application in the broker and generating the flows with in it.

In this doc we will concentrate on http nodes.

Step -1

Drag HttpInput, compute and HttpReply components from the palette and arrange them as shown in the picture below.

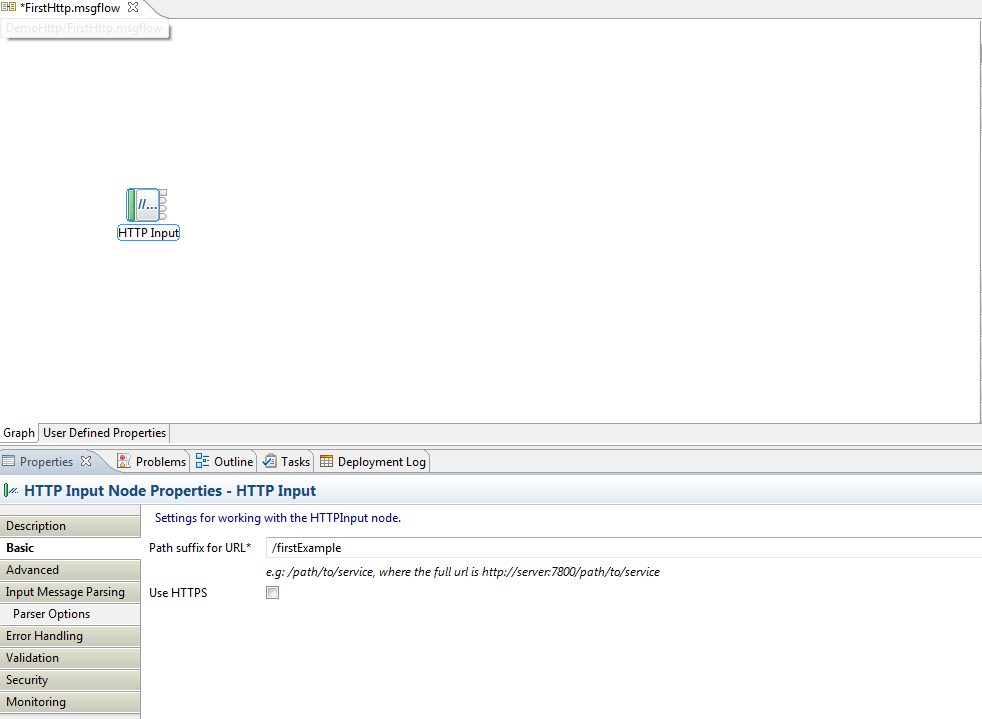
Connect HttpInput **OUT** terminal to **IN** terminal of compute node and one **OUT** terminal of compute node to **IN** terminal of HttpReply node.



**Step-2**

Now click on the HttpInput node open the properties (ALT+Enter) for that node.

you will see some properties window like below.



In the Basic section of the properties fill the path suffix for URL as of your wish.Here given /firstExample.

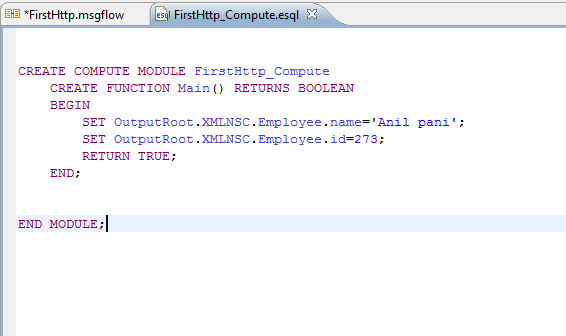
The default port number is 7080 where our application will listen on localhost,if you want to change the port number then you can be able to this by modifying the properties of broker.

This would append the application Url [http://localhost:7080](http://localhost:7080/firstExample).

So here the URL will look like this <http://localhost:7080/firstExample>.

**Step-3**

Now double click on the compute node and replace the default code with this piece of code.



**Step-4**

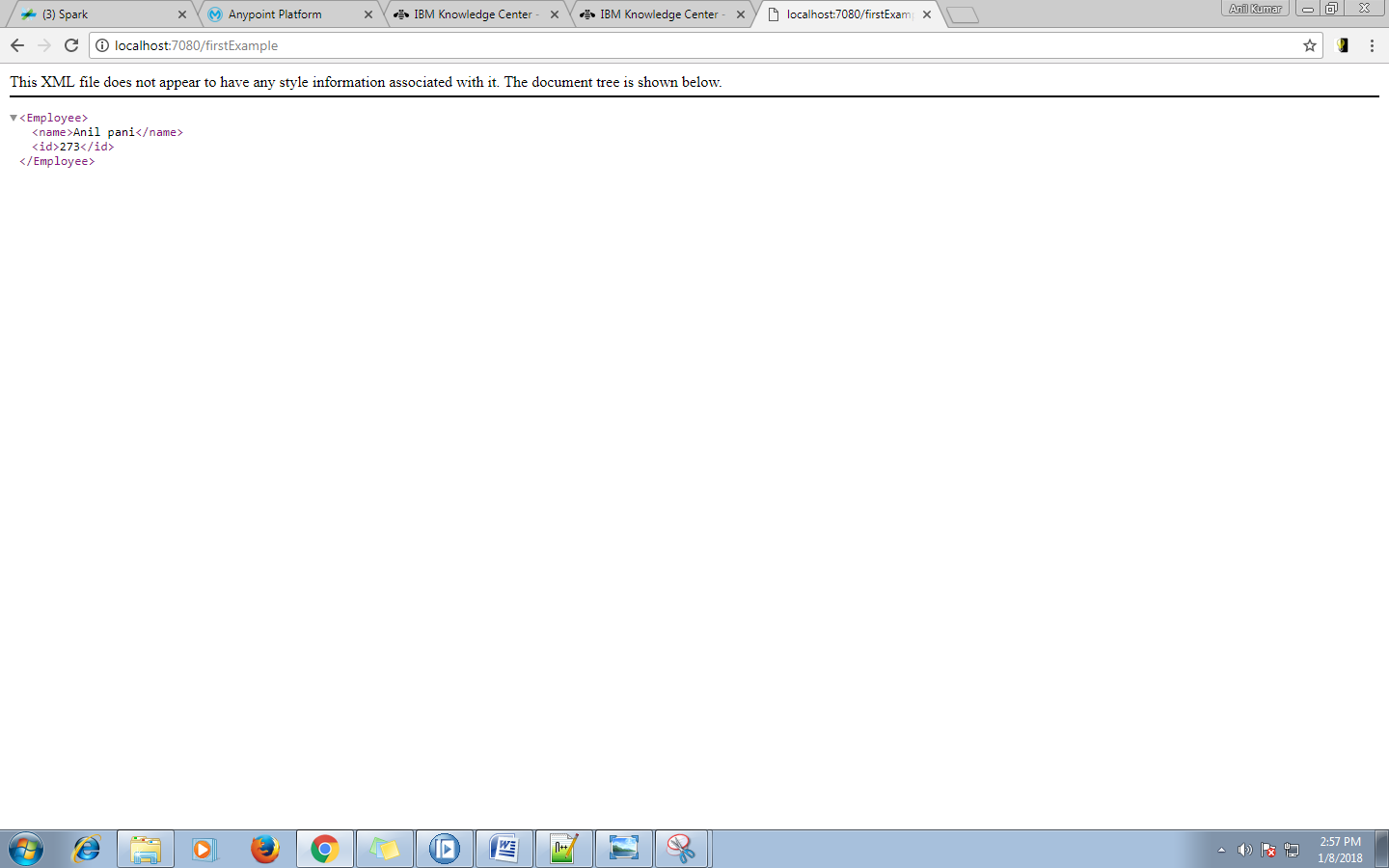
Donot modify any thing to the HttpReply node go with the default settings.

Now save the application and deploy the application to the node by building it.

**Step-5**

Open any browser and enter the URL : <http://localhost:7080/firstExample> in the address bar and hit **Enter.**

If all the configuration was successfull then you will able to see similar kind of response as below in your browser.



**Task-2**

**Parsing Querystring parameters.**

To read the query parameter from the incoming request follow these steps.

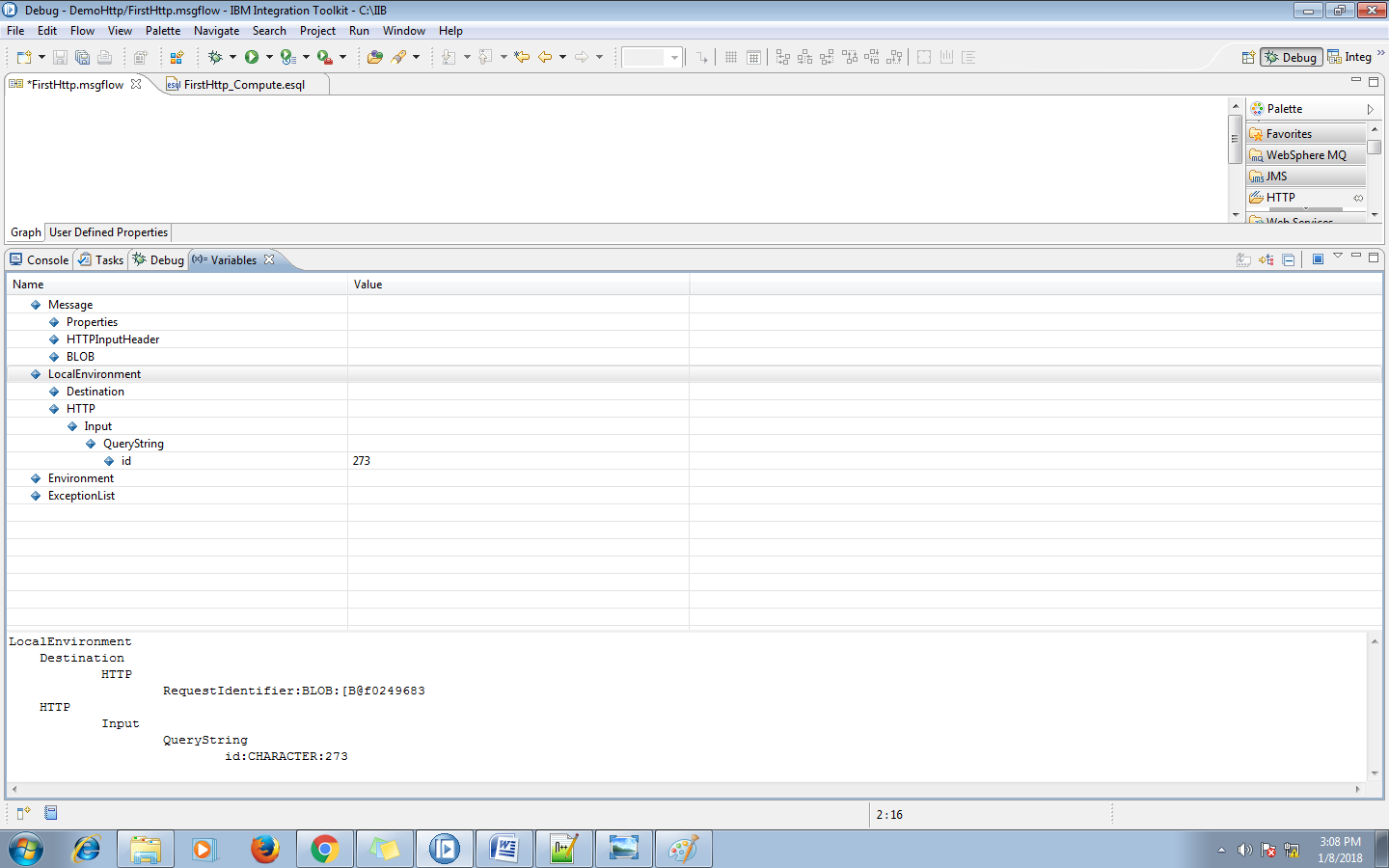
**step-1**

Create the application similar to **Task-1.**

In HttpInput node properties click on the advance tab and check(select) the parse query string option.It will parse and make avalibilty all the query parameter to the **InputLocalEnvironment**.

Now you can access the query paramerter value from this path

InputLocalEnvironment.HTTP.Input.QueryString.<query param name>.



**Step-2**

Now open the properties of compute node by selecting the compute node and pressing the ALT+Enter key .

In the basic tab go to **compute mode**  property and select LocalEnviornment and message option as shown in the below picture.

Update the compute node code as below

CREATE COMPUTE MODULE FirstHttp\_Compute

CREATE FUNCTION Main() RETURNS BOOLEAN

BEGIN

DECLARE id CHARACTER;

SET id= InputLocalEnvironment.HTTP.Input.QueryString.id;

SET OutputRoot.XMLNSC.Employee.name='Anil pani';

IF id=273 THEN

SET OutputRoot.XMLNSC.Employee.id=273;

ELSEIF id =274 THEN

SET OutputRoot.XMLNSC.Employee.id=274;

ELSE

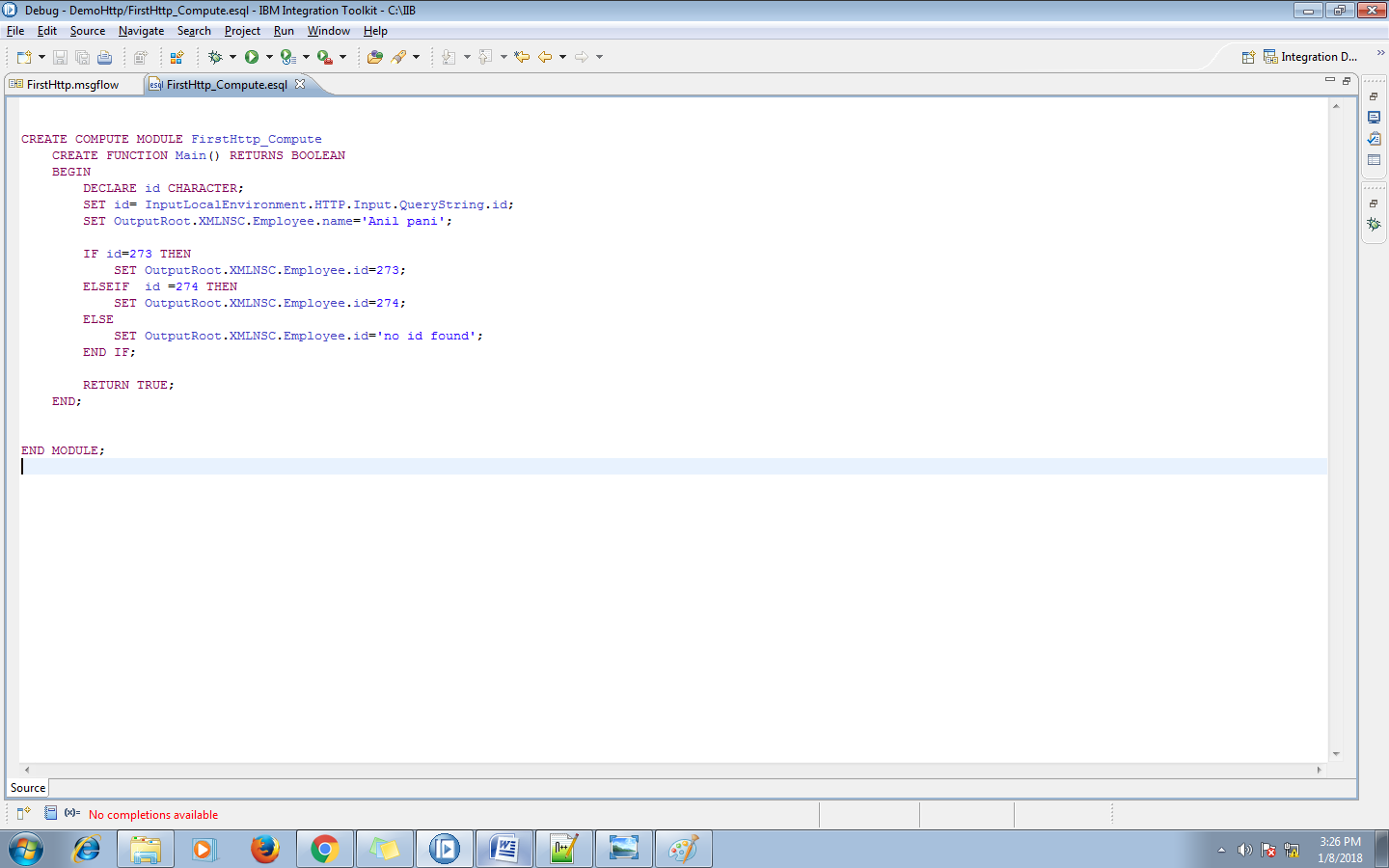
SET OutputRoot.XMLNSC.Employee.id='no id found';

END IF;

RETURN TRUE;

END;

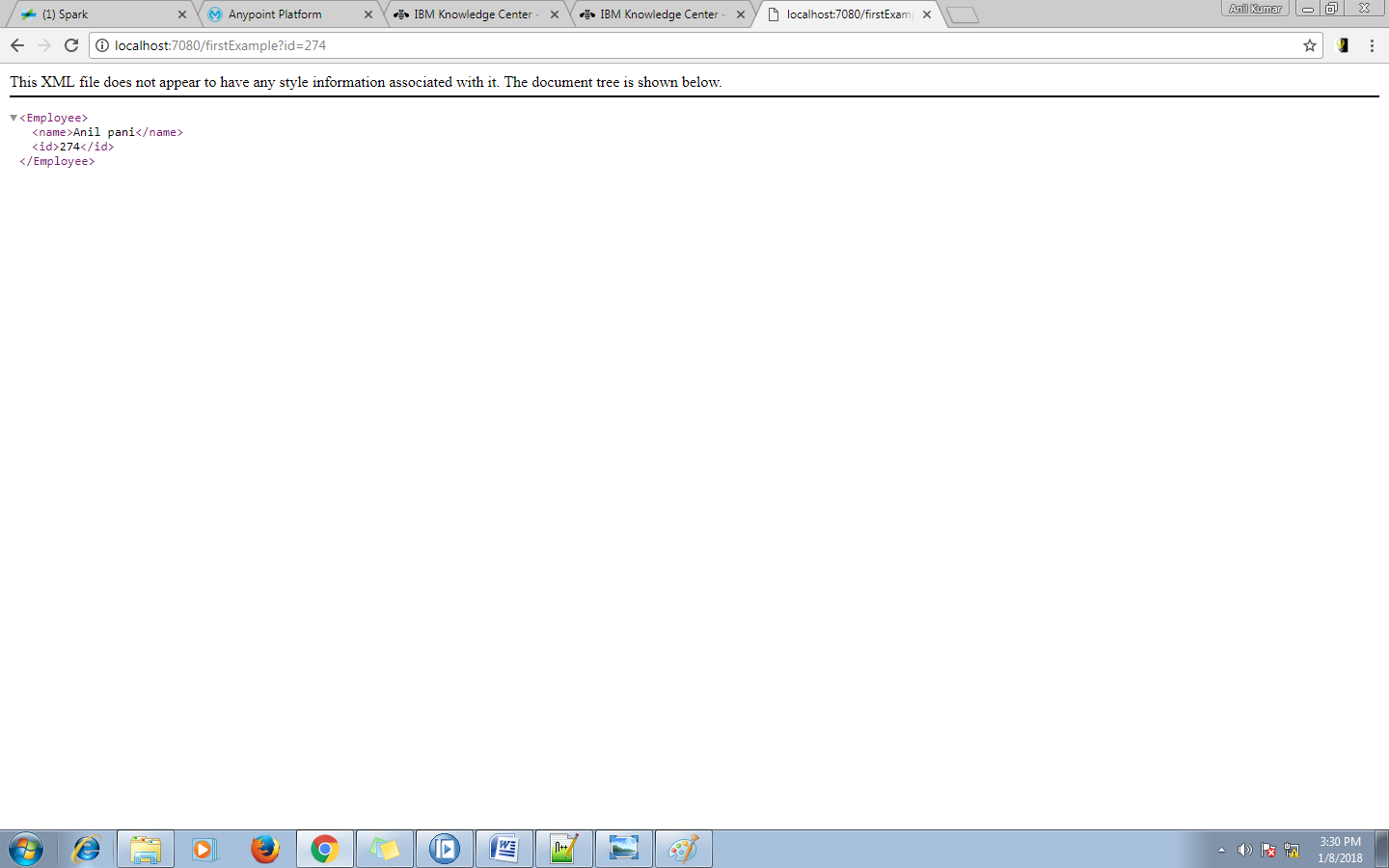
END MODULE;



**Step-3**

Now open the browser and enter the address as <http://localhost:7080/firstExample?id=273>

You will able to see the response similar this one



**Task-3**

**Using the HttpHeader node.**

In this task we will set the response header to user defined one inplace of the default 200 response.

The default status code of any HttpReply is 200 we can modify and set the status code to any valid status code response.

For this we need to follow some steps like follows.

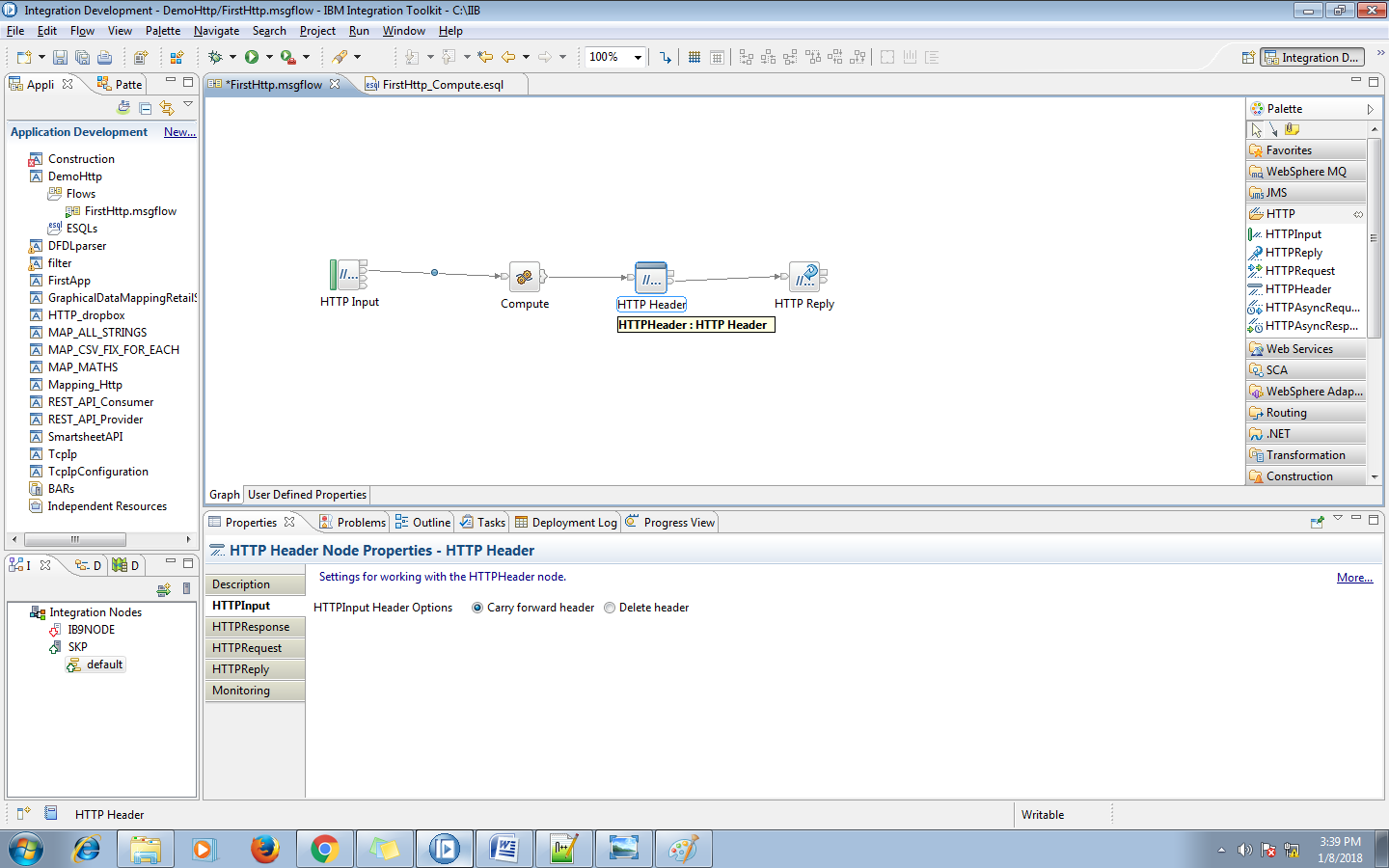
**Step-1**

Create simple application like earlier.

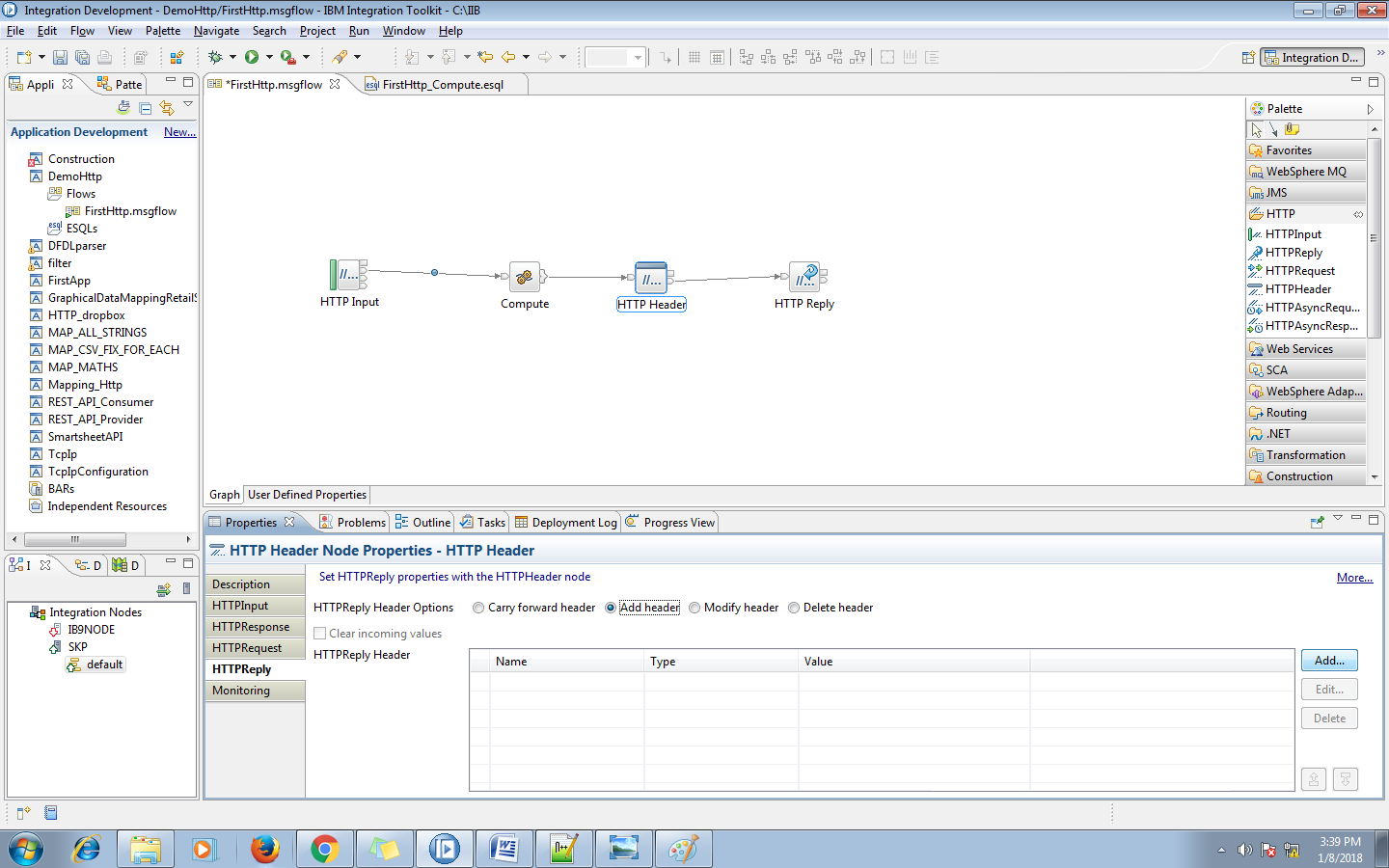
**Step-2**

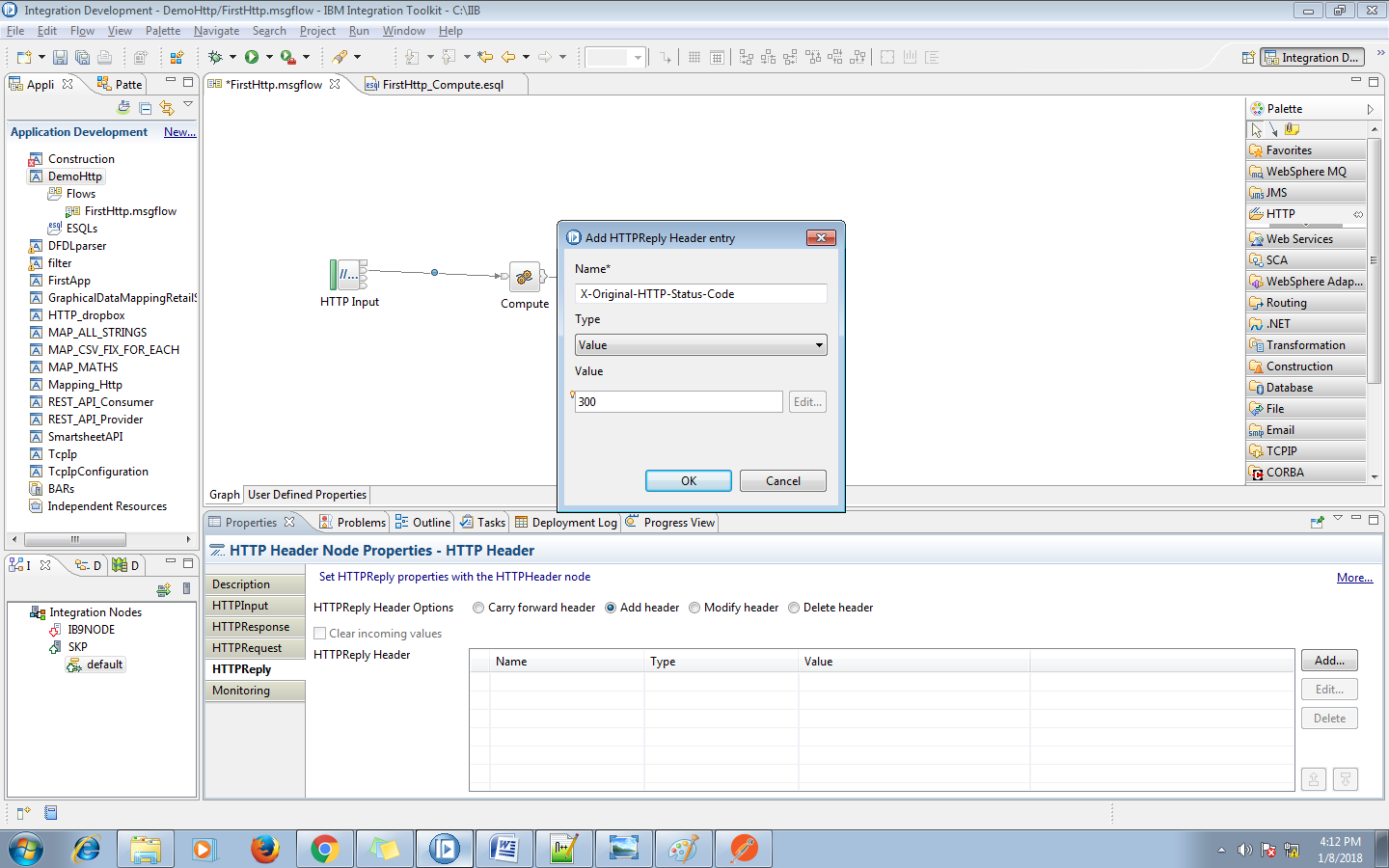
Now add a HttpHeader node after the compute node and connect the **OUT** terminal of compute node to **IN** terminal of HttpHeader node.

Now open the HttpHeader properties you will able to see the below tabs like in the figure below

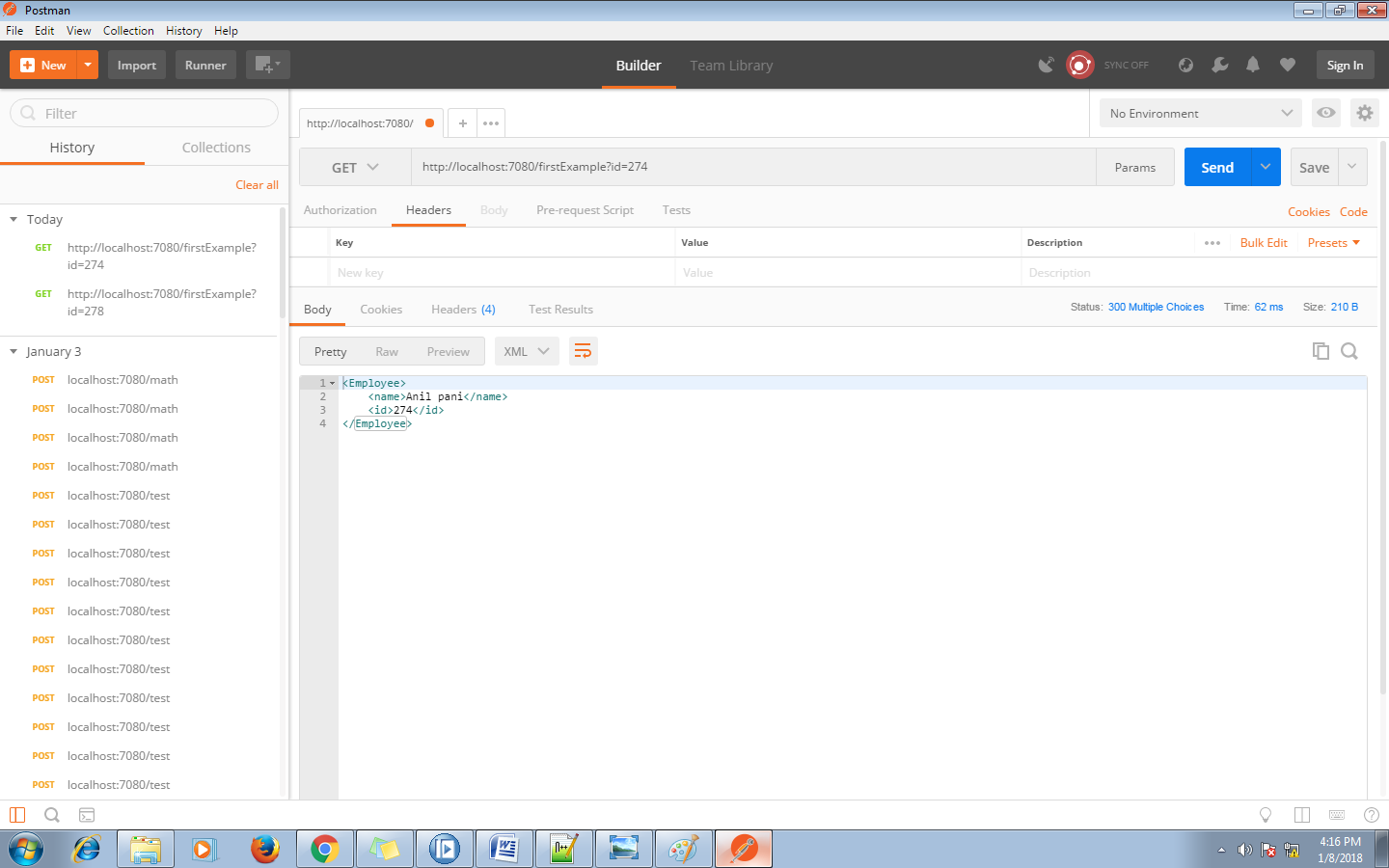


Now go to HTTPReply tab and select Add header option and click on the add button in the right side like below figure



After clicking on the add button you will find a pop up window in that give **"X-Original-HTTP-Status-Code"**  in the name field and **"300"** in the value field,while setting the code we need to follow the Http status code standards.

If you will check them you will find now the response code is changed to 300 in the response.

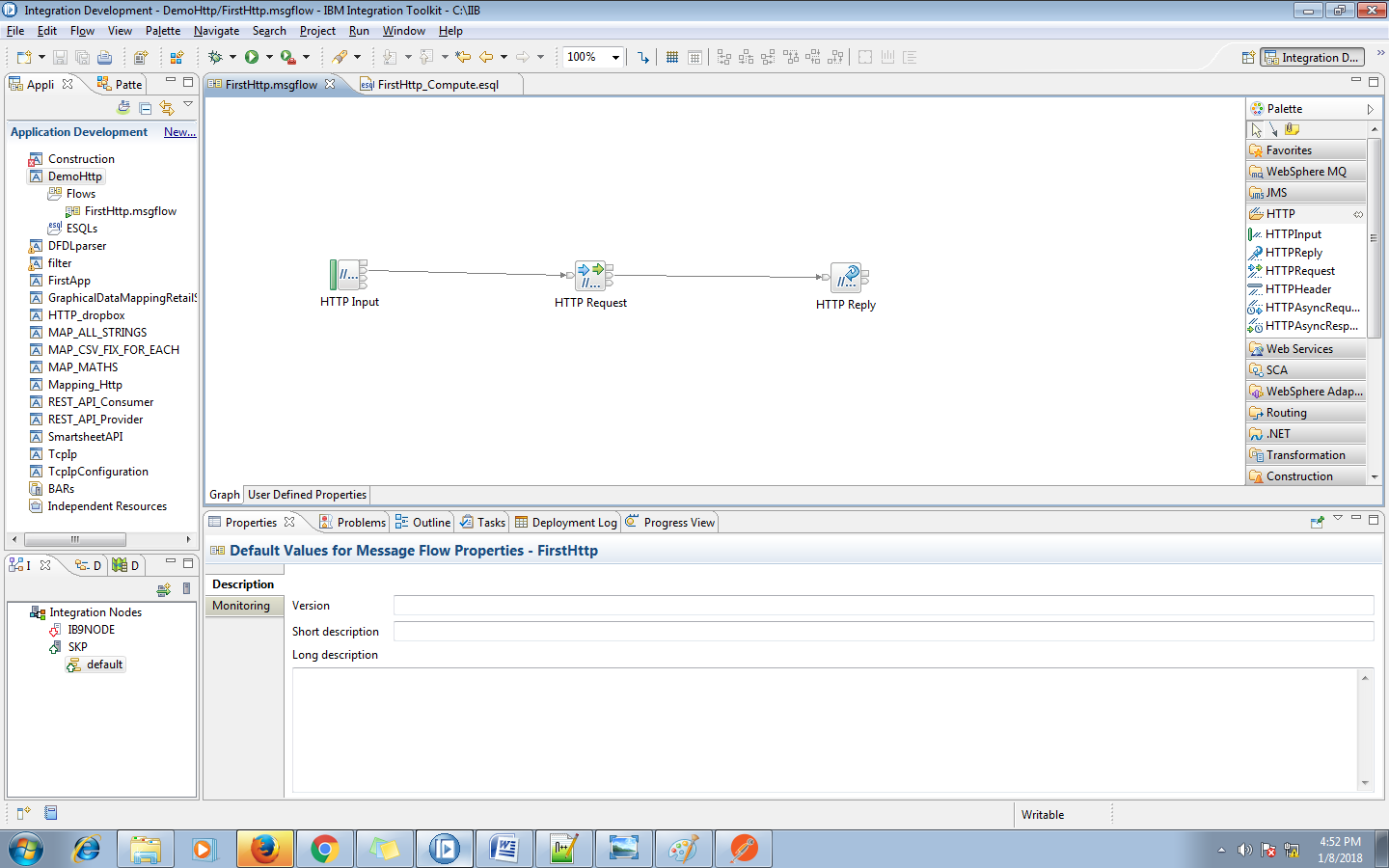


**Task-4**

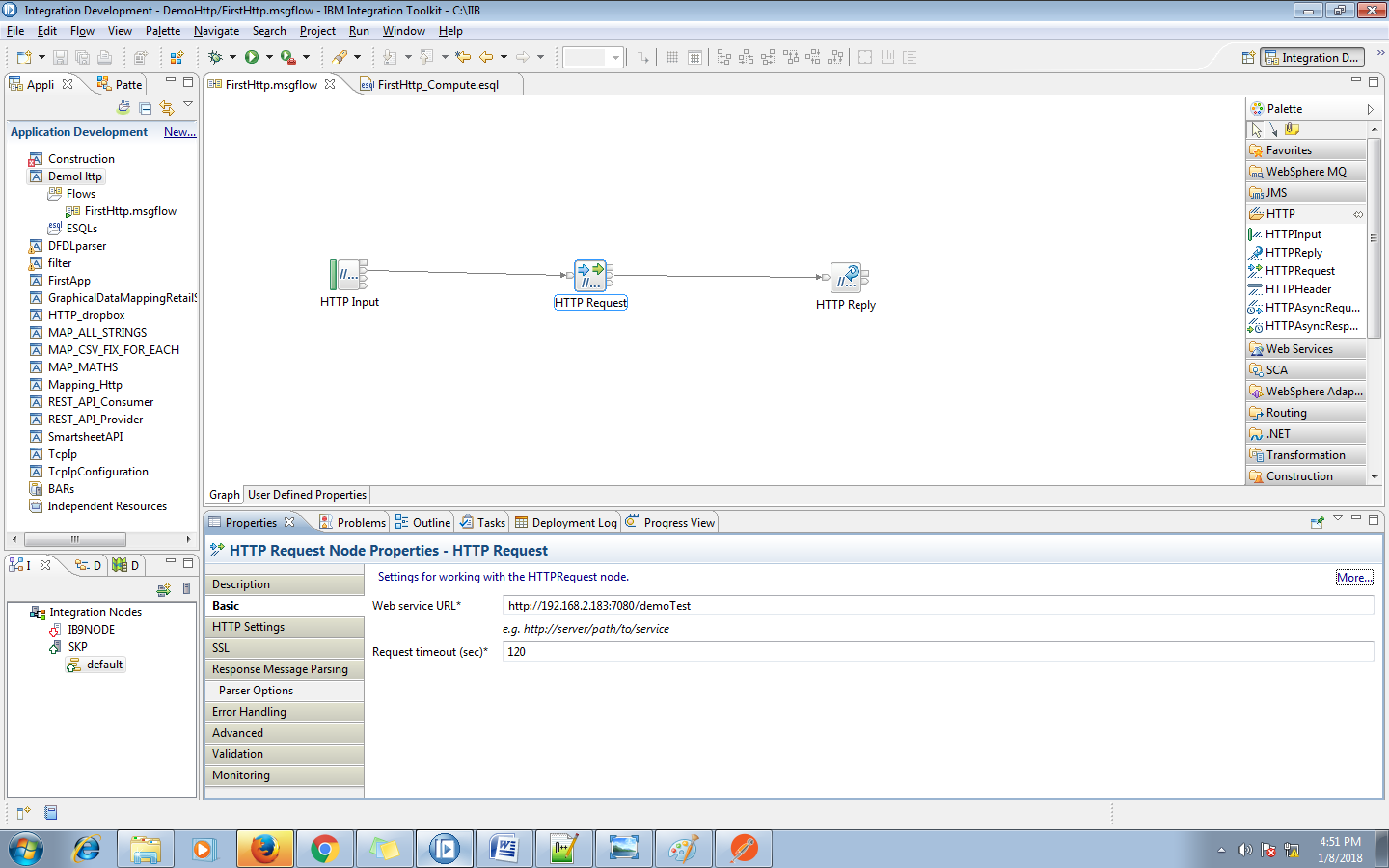
**Using the HttpRequest node.**

This node can be used to consuming rest apis with in the flow processing.

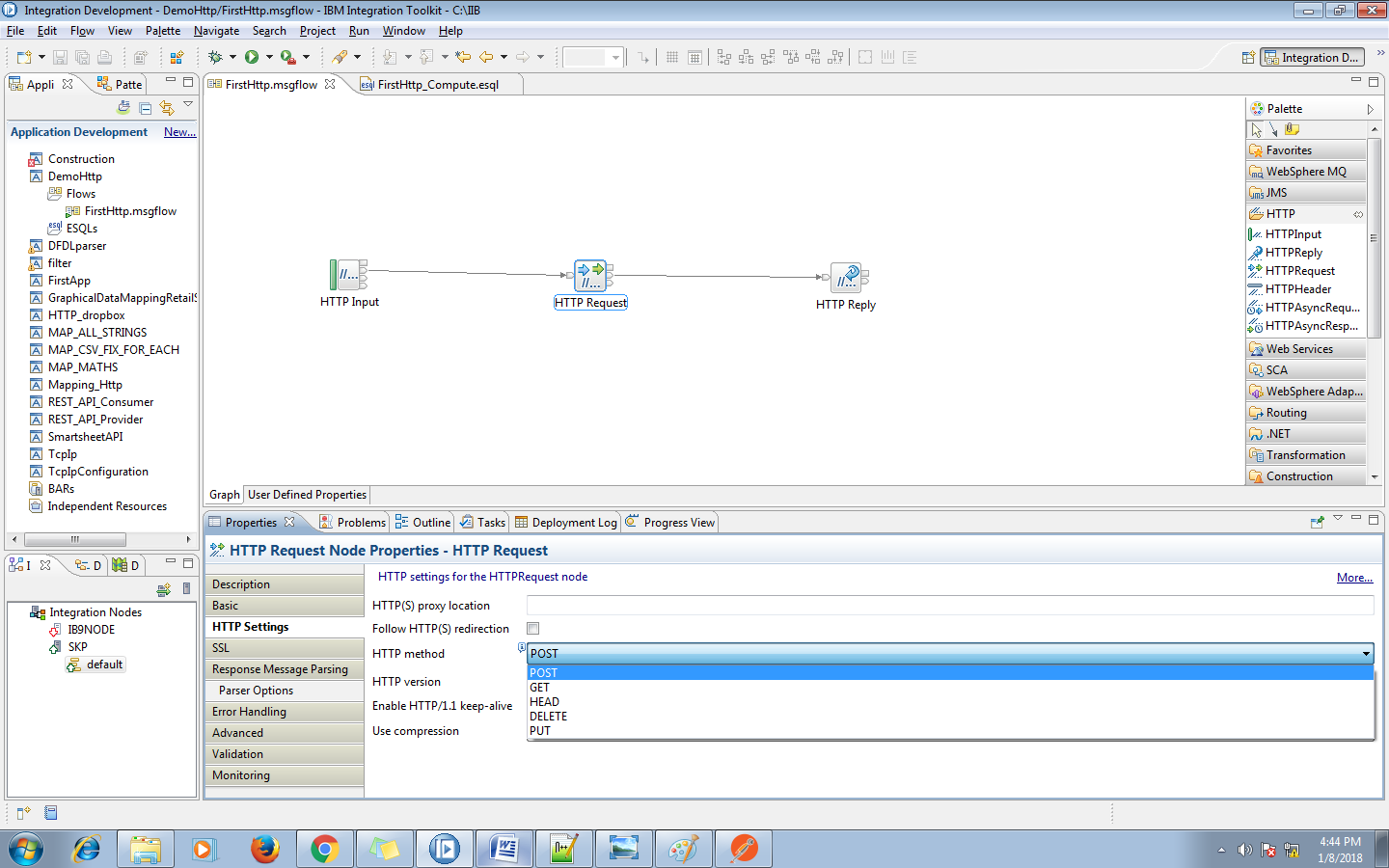
**Step-1**

Create a simple application with one httplistener,httpRequest and http reply node like below and connect them accordingly.

**Step-2**

Now in HttpRequest node go to properties and in the basic section provide the web service URL to which your going to connect.Here i have connected to a local network system using the URL [http://192.168.2.183:7080/demoTest](http://192.168.2.183:7080/demoTest%20) 

In the HTTP setting tab select the HTTP method from the list.

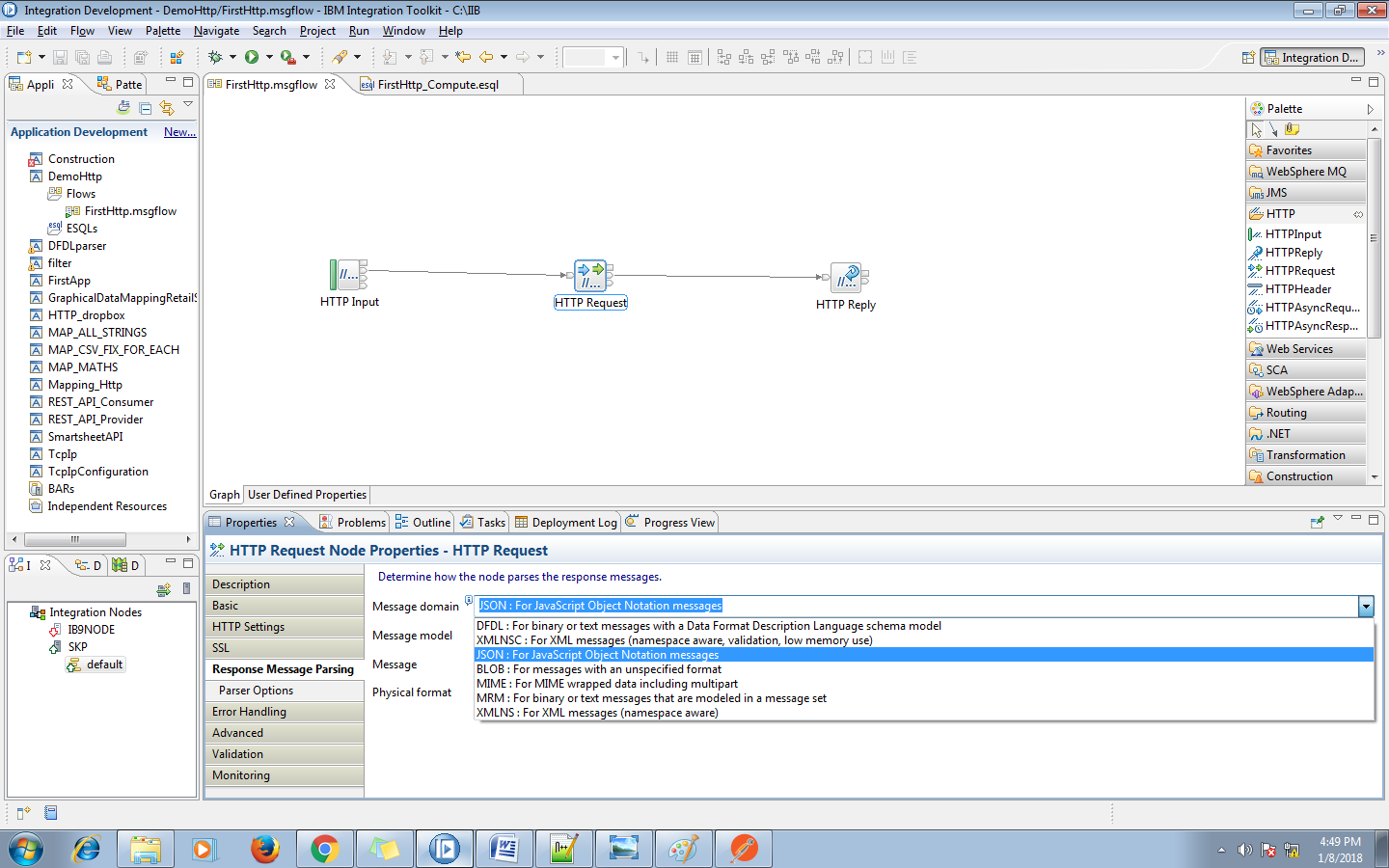


Here the request is a GET method so i will set the HTTP method to GET.

If the requested URL is a HTTPS url then configure the certificate to the truststore of the broker.

if you know the response type and then mention it on the Message domain option of the Response message parsing tab like below figure.

Here i have selected the JSON option as the response type is application/json



Now save the application and deploy it to the node.

Now open the browser and hit your listener path i.e

<http://localhost:7080/firstExample>

You will see the response from the webservice in the browser like below figure

