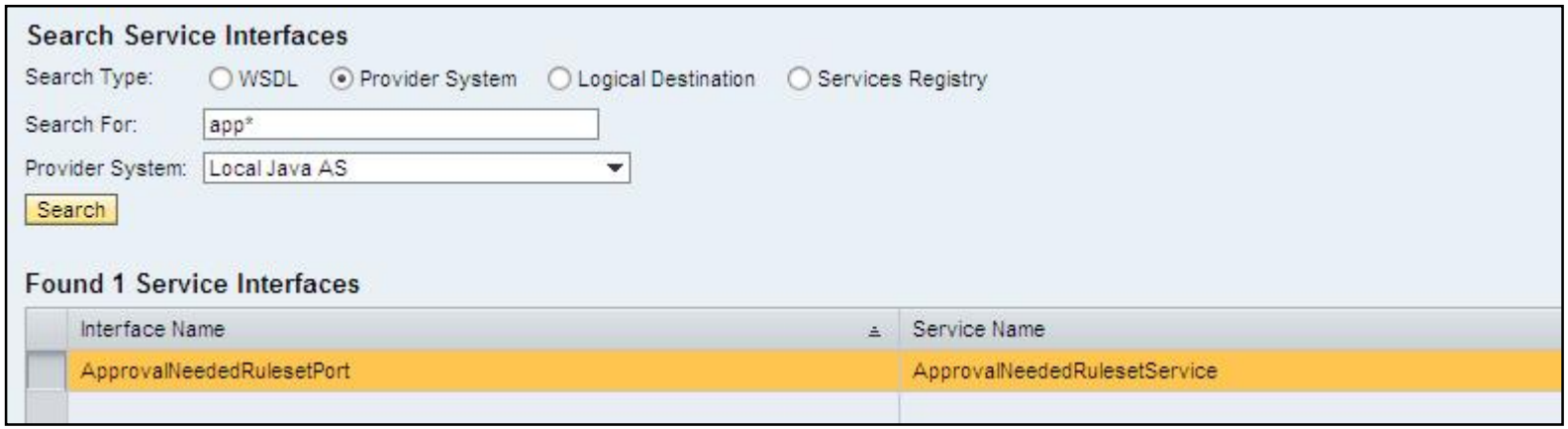
# Invoking Rules Web Service from WebDynpro CE 7.1

Arti Gopalan
Solution Management Rollout, SOA & Platform / BPM, BRM

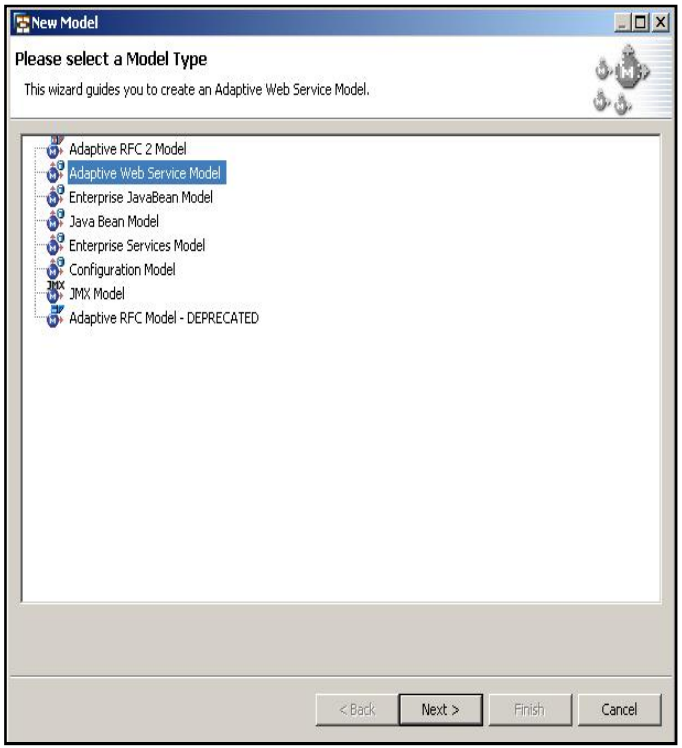
# Step 1:

1. The BRM project has been deployed into the Application Server.
2. Rules Web Service has been created by using the WSGeneratorTool in SDN and deployed in the Application Server.
3. The Web Service generated can be tested in the WSNavigator portal.

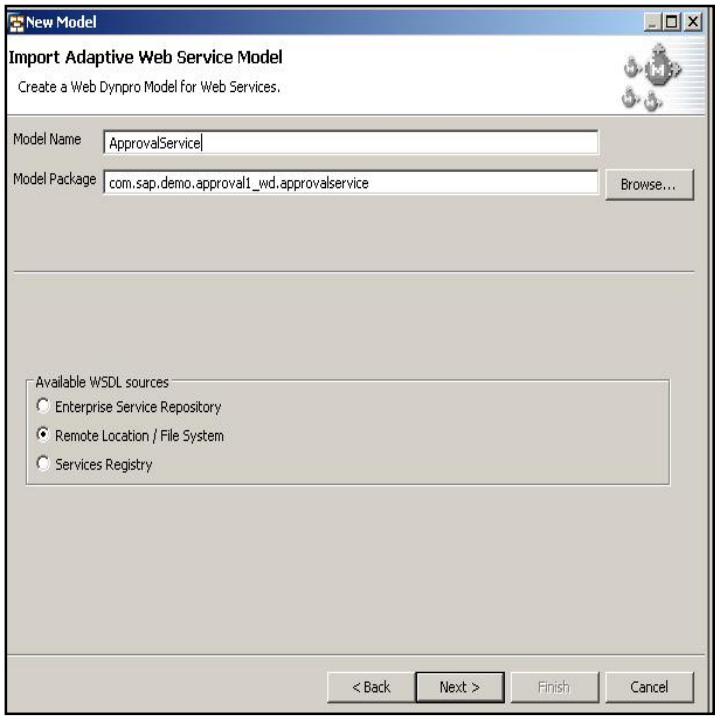


# Step 2: Consuming Web Service in WDPro Application

1. Create a new WebDynpro DC.
2. Using the wsdl of the rules web service create a new Adaptive Web Service Model in the WDPro DC.

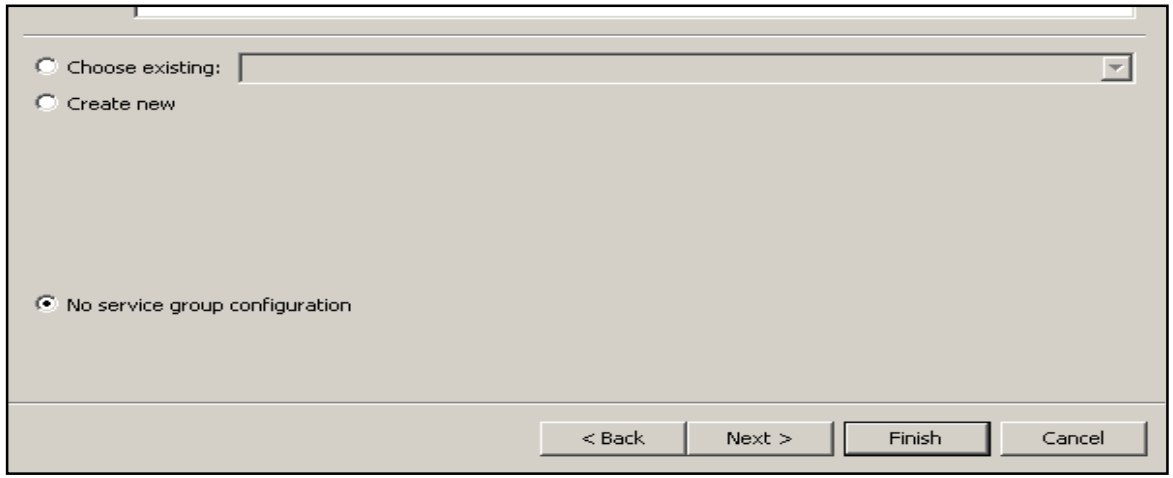



1. Select Model Type

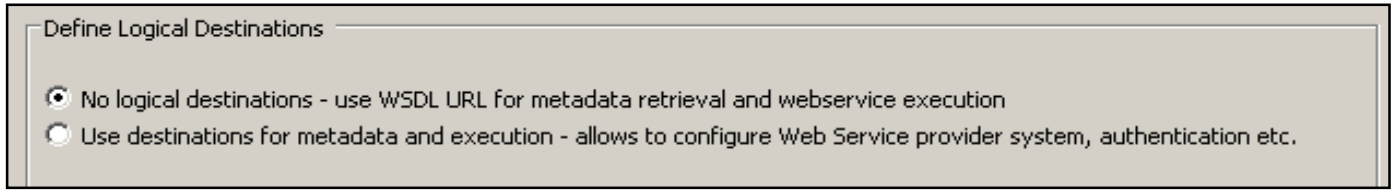



2. Set the model name and select the WSDL source.

# Step 2.2:




3. There is no service group configuration for this service being used.




4. Logical destination for the service needn't be defined for this service being used, here in the example.

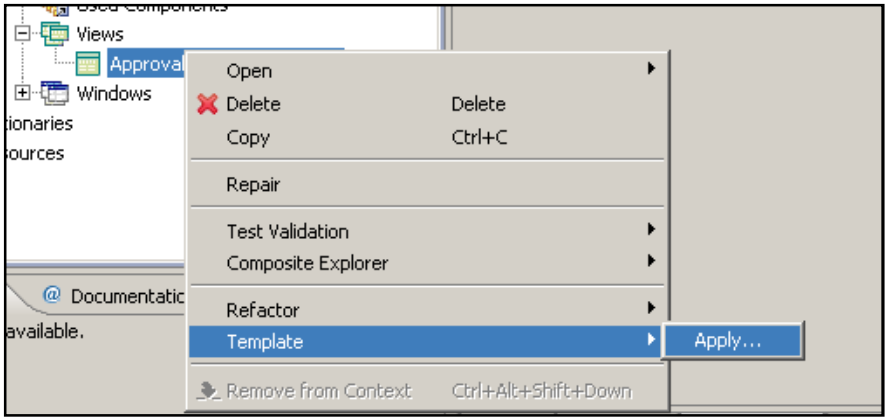
# Step 3: Creation of WDPro Component

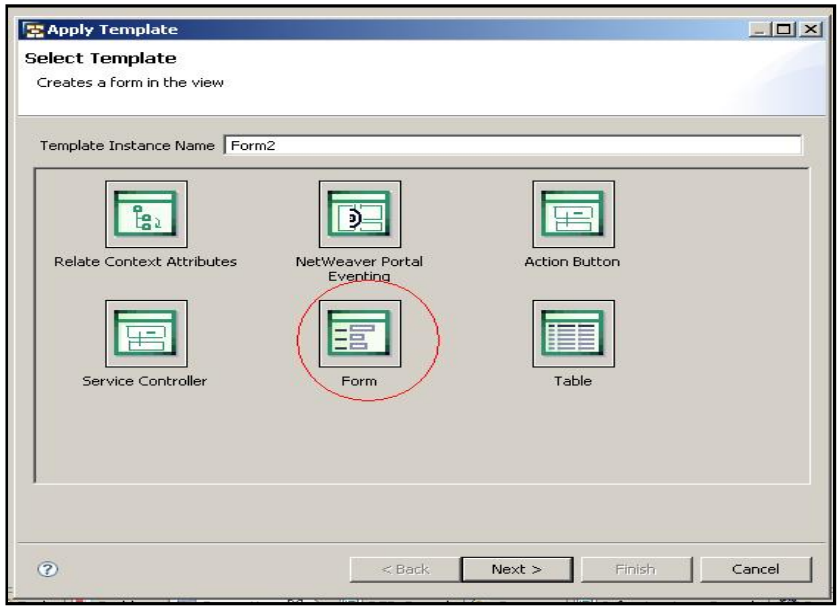
1. Create a new Application and associate a new Component to it.
2. Add the Adaptive Web Service Model created in the previous step to the components used models node.
3. To the component controller Apply Service Controller Template, this will internally create the required lines of code to make the web service model available for use in the component.
4. In the Component, associate the View created to the component controller.
5. Create the required view.

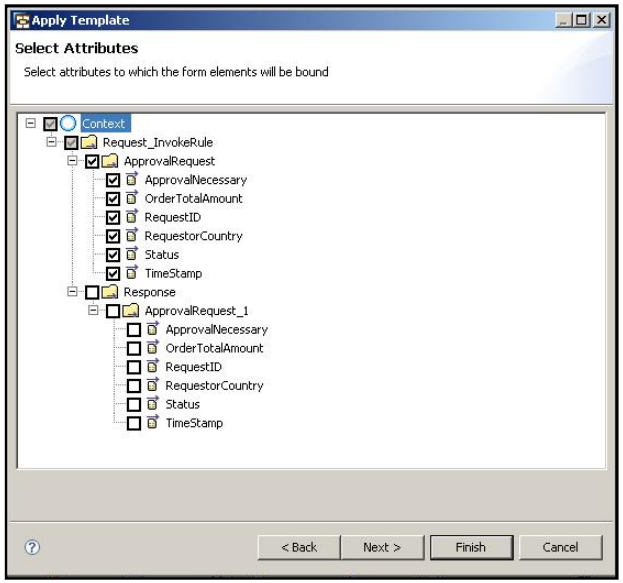
# Step 5: Creating View for Approval Web Service

I. Creating form for input values.
Steps:

- Apply Form Template to the View.
- Select Input Attributes (Request attributes) using which the form is to be created.



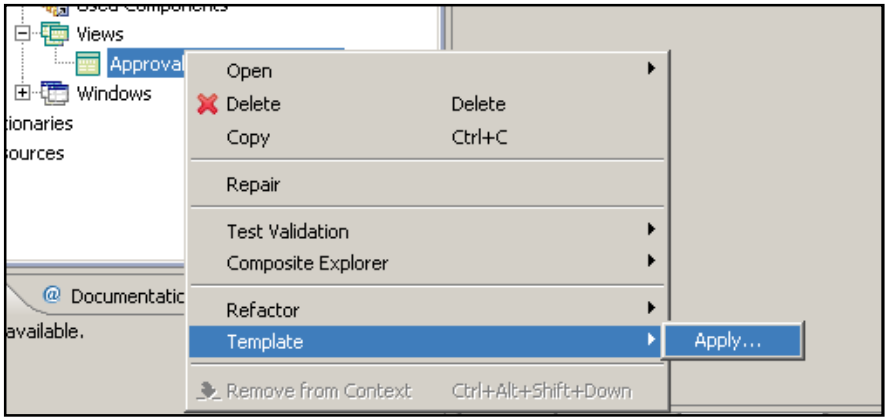



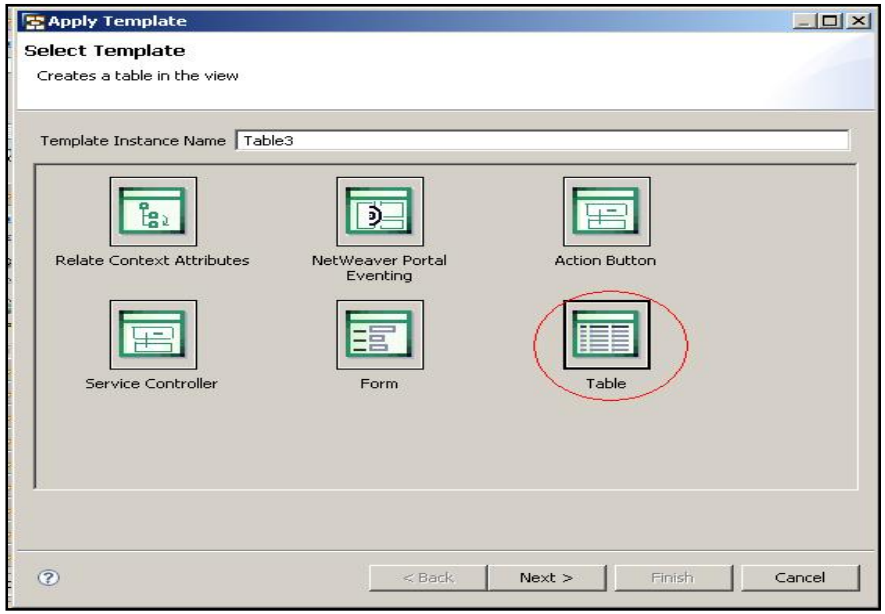



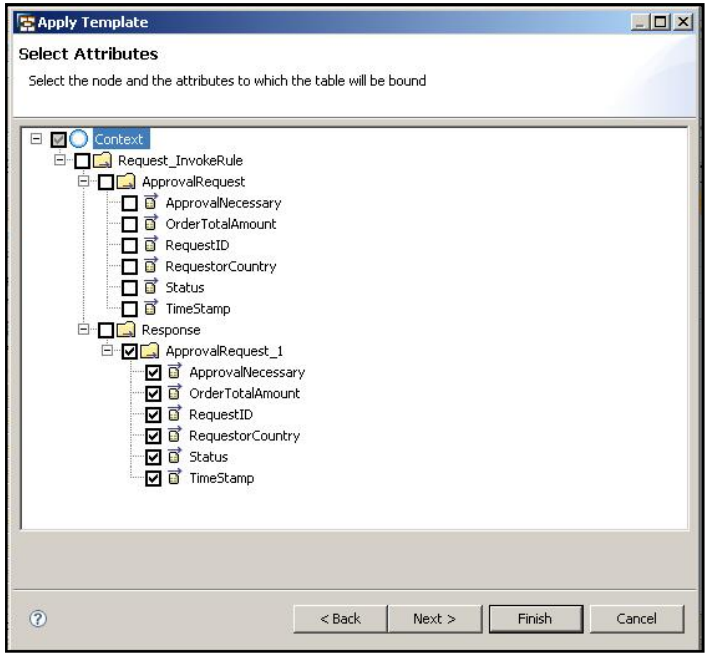
# Step 5.2: Creating View for Approval Web Service

II. Creating table for output values.
Steps:

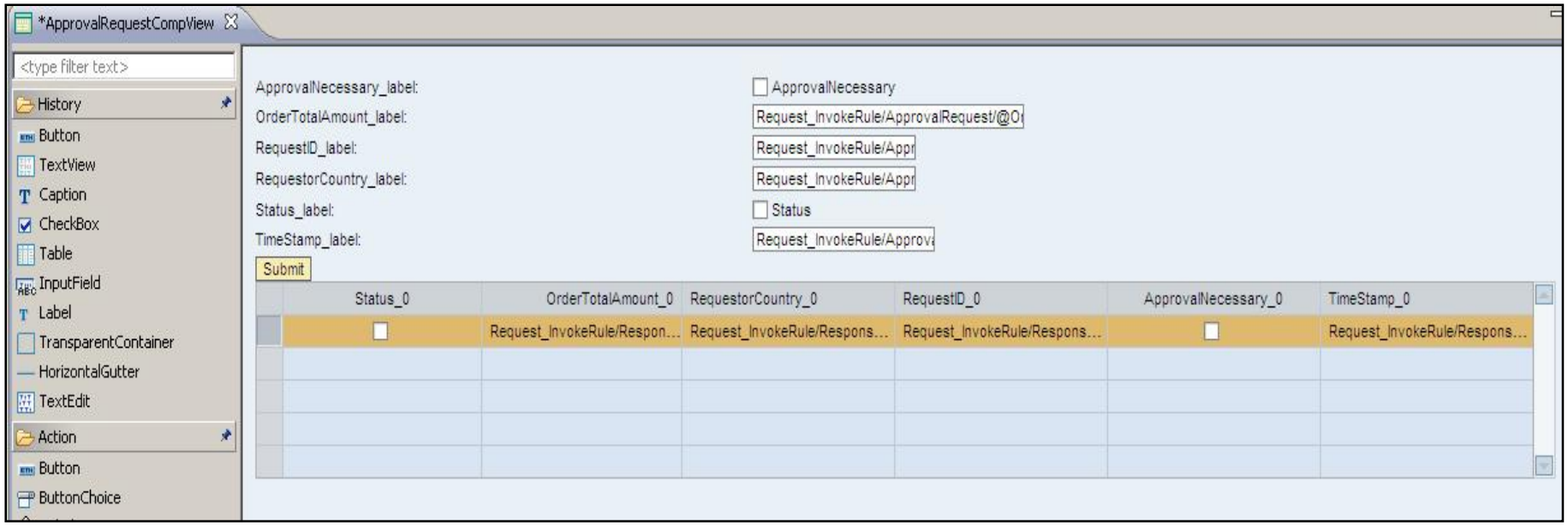
- Apply Table Template to the View.
- Select Output Attributes (Response attributes) for which the table is to be created.



# Approval Request Component View created..




The input attributes of the web service being used in this example are –

1. Approval Necessary
2. Order Total Amount
3. Request ID
4. Requestor Country
5. Status
6. Time Stamp

The required Output is whether or not approval of higher authority is required.

# Step 6: Making Code Changes

Applying the Service Controller Template on the component controller, generates the following lines of code in wdDoInit() and executeInvokeRule() methods.

```
//@@begin javadoc:wdDoInit()
/** Hook method called to initialize controller. */
//@@end
public void wdDoInit()
{
  //@@begin wdDoInit()
  //$$begin Service Controller(-2052985301)
  ApprovalServiceModel approvalServiceModelModel = new ApprovalServiceModel();
  Request_InvokeRule request_InvokeRule = new Request_InvokeRule(approvalServiceModelModel);
  ApprovalRequestType approvalRequest = new ApprovalRequestType(approvalServiceModelModel);

  //Initialize values
  approvalRequest.setApprovalNecessary(false);
  approvalRequest.setOrderTotalAmount(0);
  approvalRequest.setRequestID("");
  approvalRequest.setRequestorCountry("");
  approvalRequest.setStatus(false);

  Date defaultDate = new Date();
  Timestamp timeStamp = new Timestamp(defaultDate.getDate());
  approvalRequest.setTimeStamp(timeStamp);

  request_InvokeRule.setApprovalRequest(approvalRequest);
  Response_InvokeRule response = new Response_InvokeRule(approvalServiceModelModel);
  request_InvokeRule.setResponse(response);
  ApprovalRequestType approvalRequest_1 = new ApprovalRequestType(approvalServiceModelModel);
  response.setApprovalRequest(approvalRequest_1);
  wdContext.nodeRequest_InvokeRule().bind(request_InvokeRule);
  //$$end
  //@@end
}

//@@begin javadoc:wdDoExit()
/** Hook method called to clean up controller. */
//@@end
```

# Step 6.2: Making Code Changes

```
//@@begin javadoc:executeInvokeRule()
/**
 * Method declared by application.
 */
//@@end
public void executeInvokeRule( )   {
  //@@begin executeInvokeRule()
  //$$begin Service Controller(-1620651913)
  try
  {
    wdContext.currentRequest_InvokeRuleElement().modelObject().execute();
    wdContext.nodeResponse().invalidate();
  }
  catch(Exception e)
  {
    wdComponentAPI.getMessageManager().reportException(e.getMessage());
  }
  //$$end
  //@@end
}
```

To initialize the input attribute values we add the following lines of code in the wdDoInit() method -

```
//Initialize values
approvalRequest.setApprovalNecessary(false);
approvalRequest.setOrderTotalAmount(0);
approvalRequest.setRequestID("");
approvalRequest.setRequestorCountry("");
approvalRequest.setStatus(false);

Date defaultDate = new Date();
Timestamp timeStamp = new Timestamp(defaultDate.getDate());

approvalRequest.setTimeStamp(timeStamp);
```

# Step 6.3: Making Code Changes

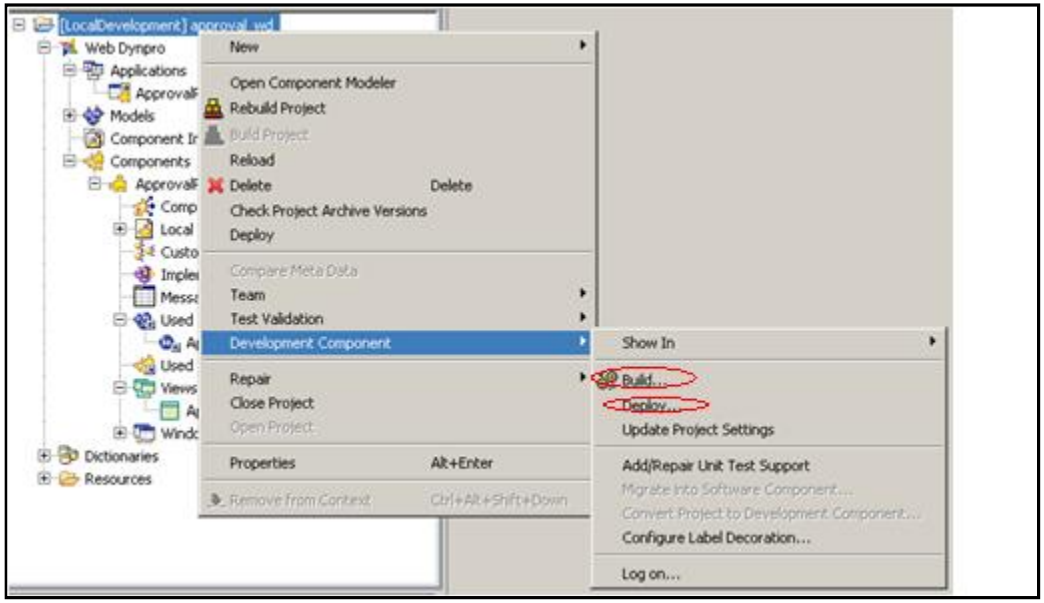
Next step is to add lines of code to call the executeInvokeRule() method of the service controller on button click.

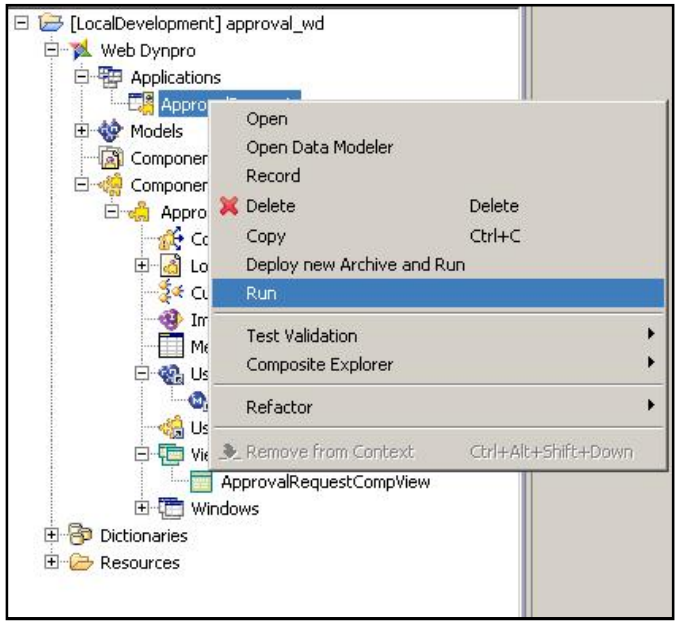
```
//@@begin javadoc:onActionSubmit(ServerEvent)
/**
 * Declared validating event handler.
 *
 * @param wdEvent generic event object provided by framework
 */
//@@end
public void onActionSubmit(com.sap.tc.webdynpro.progmodel.api.IWDCustomEvent wdEvent )
{
  //@@begin onActionSubmit(ServerEvent)
    wdThis.wdGetApprovalRequestCompController().executeInvokeRule();
  //@@end
}
```

# Step 7: Run the Application

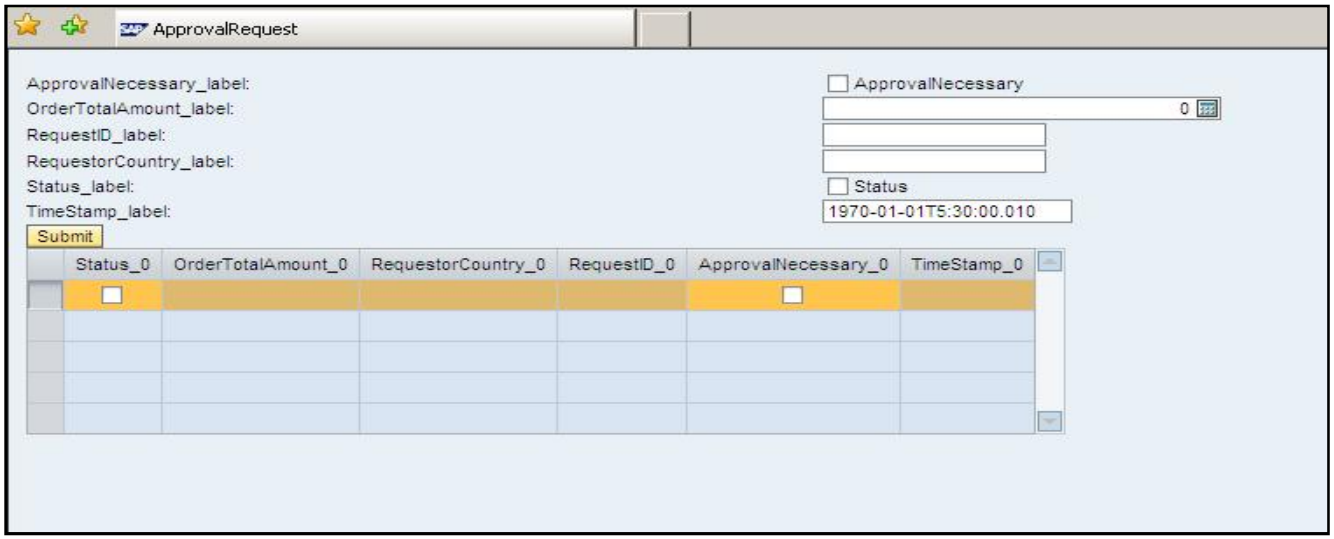
1. Once the model, the component, the view have been created and the required code modifications done, the next step is to build and deploy the DC.

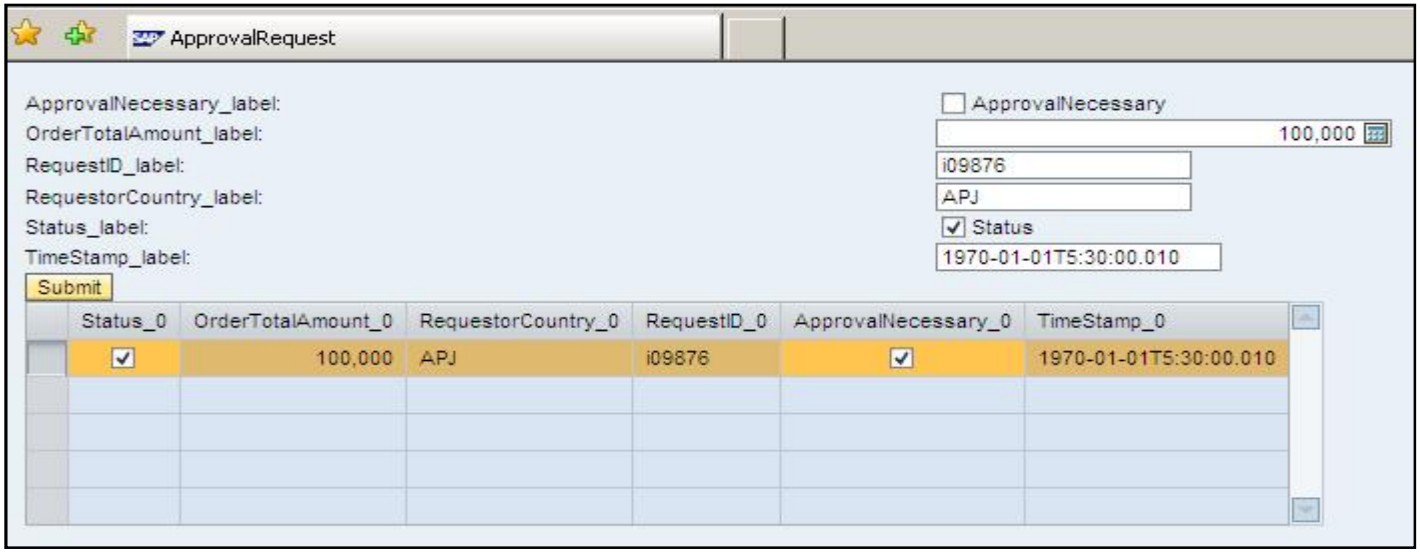
2. Now select the context menu of the Application created in the DC and select the option to Run.



# Step 7.2: Run the Application




After Invocation of Rules

# References


1. https://www.sdn.sap.com/irj/sdn/nw-rules-management?rid=/webcontent/uuid/f066ec08-474b-2b10-4a97-b66d605de037
2. https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/10c6b0c3-0c50-2b10-97ac-e98ad72aaf8f
3. https://www.sdn.sap.com/irj/sdn/nw-wdjava
4. https://www.sdn.sap.com/irj/sdn/nw-wdjava?rid=/webcontent/uuid/403e6bf5-426e-2910-b0a8-a95548724af9#section17

# Thank you!