

Starting a BPM Instance from Java Code and Scheduling It using Cron Jobs

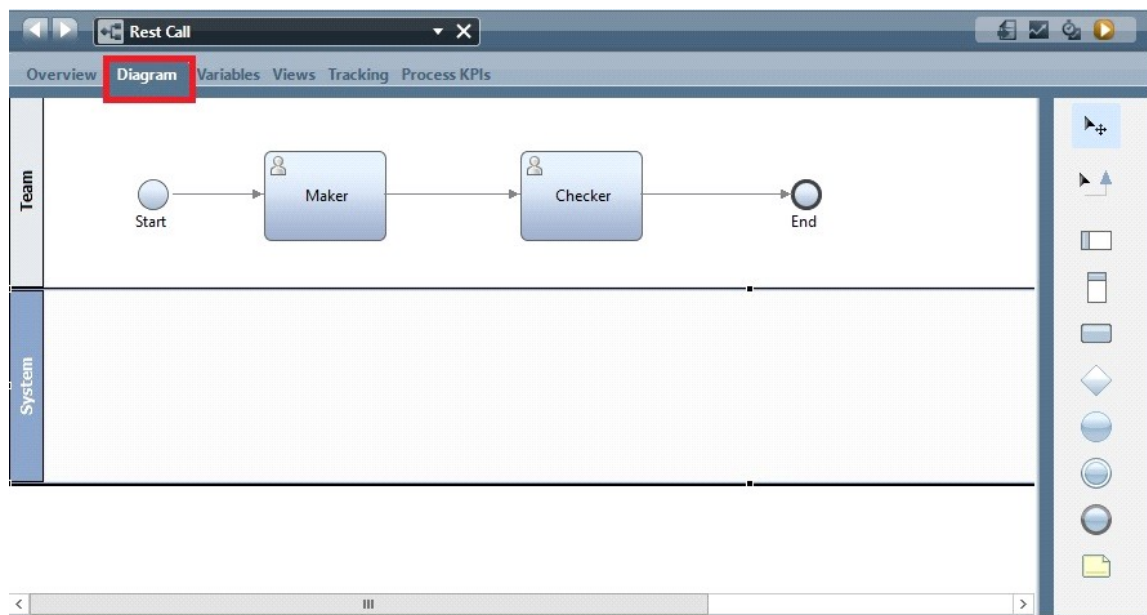
Description:

Using thread timer scheduler(CRON Jobs) in java, BPD instance is created for every 5 seconds and after that sub-sequent task are followed in IBM BPM.

Timer class provide method call that are used by thread to schedule a task, such as running a block of code after some regular instant of time.

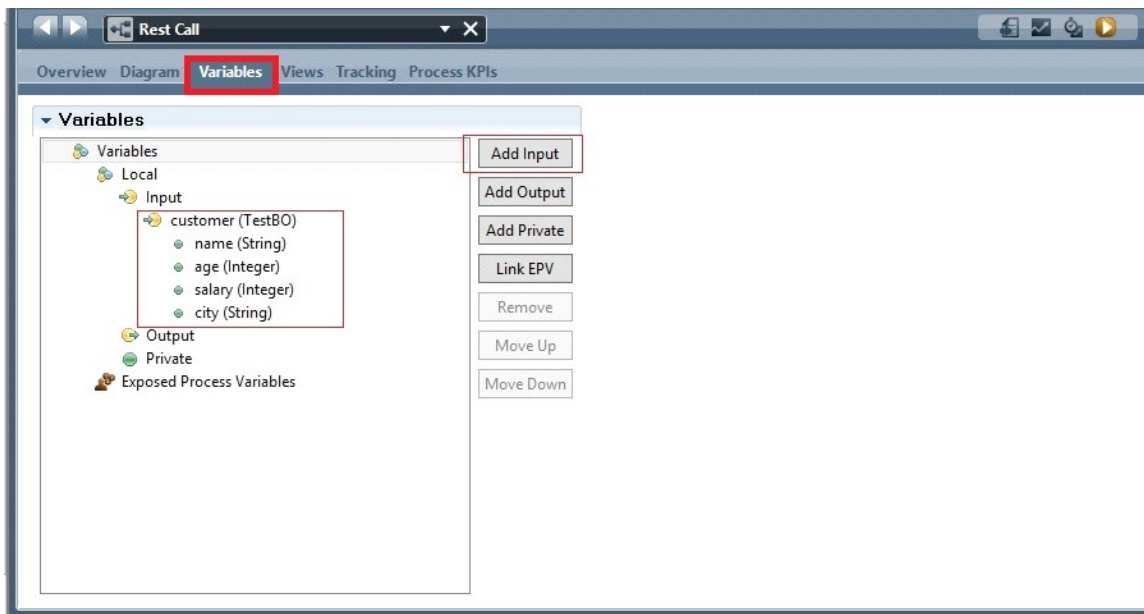
Step 1:

Create BPD, which has two activity namely, maker & Checker.



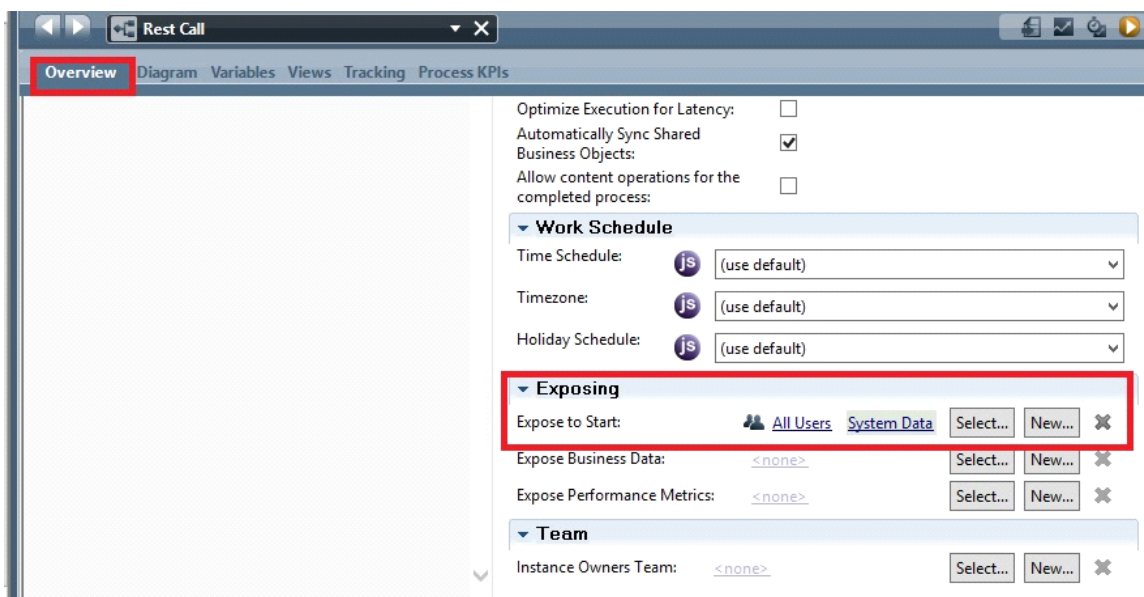
Step 2:

In Variable section, Declare the customer variable type of TestBO. Here Business Object having four parameters.



Step 3:

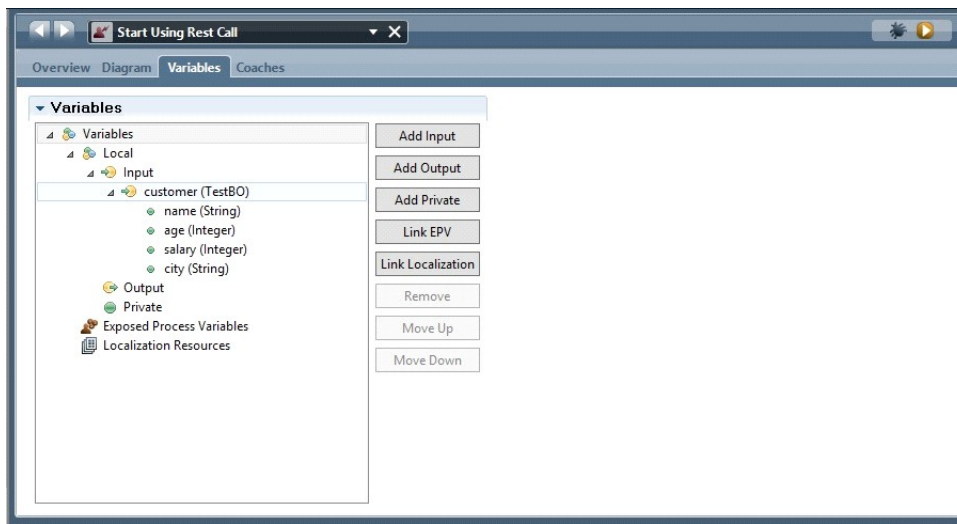
Exposed the BPD to team or Specific set of users.



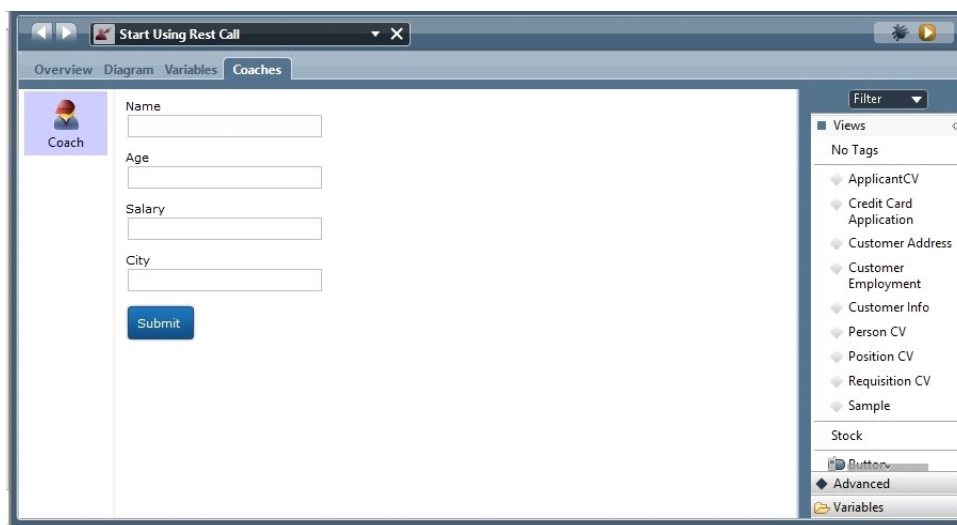
Step 4:

Create Heritage Human Service for maker activity.

In Variable section, Declare the customer variable type of TestBO. Here Business Object having four parameters.



In Coaches section, under variable section in palette component drag & drop the variables fields.



Step 5:

Login into BPMrest-ui console (<https://ibpm85:9444/bpmrest-ui/BPMRestAPITester/index.jsp>) and to form Rest API call, then need to get BPD information like BPD Id, Snapshot Id, Branch Id and Process Application Id from Exposed process(Show all BPD name, which is exposed to team) of Process API.

Find the respective BPD name(Display: Rest Call) & get all related information of BPD.

Result Type:
JSON (Javascript format)

Select API Call:

Business Process Manager REST APIs

Process API

- Retrieve Model
- External Activity Model
- Start Process
- Current State
- Evaluate JavaScript
- Update Due Date
- Suspend Process
- Resume Process
- Retry Process
- Terminate Process
- Delete Process
- Process Bulk Operation
- Instance Error Details
- Run Ad Hoc Event
- Delete Token
- Move Token
- Add Comment
- Add Document
- Update Document
- Delete Document
- Task Summary
- Process Queries [deprecated]
- Process Query Attributes [deprecated]
- Process Query Entity List [deprecated]
- Process Query Entity List Count [deprecated]
- Fire Timer
- Send Message
- Currently Running Instances
- Exposed Processes**

data

BPD Name, Which is exposed to All User, here we can find BPD details

```

{
  "branchID": "2063.1259bd45-4789-4610-97e4-f7a89",
  "branchName": "Main",
  "startURL": "/rest/bpm/wle/v1/process?action=start&42df-985f-a3a4bd6cce3e&branchId=20697e4-f7a89990b33d",
  "isDefault": false,
  "isMobileReady": false,
  "ID": "2015.203"
},
{
  "type": "process",
  "itemID": "25.2ece1c25-137c-4ca9-9c11-eb4eb72",
  "itemReference": "/25.2ece1c25-137c-4ca9-9c11-eb4eb72",
  "processAppID": "2066.6c0a6c0a-fe44-422f-96be-e700e9",
  "processAppName": "Hiring Sample Application",
  "processAppAcronym": "HSAM",
  "snapshotID": "2064.1c2e52e8-2e13-4e85-b0e4-4f969",
  "snapshotCreatedOn": "2019-04-10T05:57:28Z",
  "display": "Rest Call",
  "tip": true,
  "branchID": "2063.58d922bd-8b0b-40a5-87b0-06ec",
  "branchName": "Main",
  "startURL": "/rest/bpm/wle/v1/process?action=start&4ca9-9c11-eb4eb7294d47&branchId=2087b0-06eca31727ec",
  "isDefault": false,
  "isMobileReady": false,
  "ID": "2015.1714"
},
{
  "type": "process",
  "itemID": "25.c904b3b1-afc1-4698-bf5a-a20892c2",
  "itemReference": "/25.c904b3b1-afc1-4698-bf5a-a20892c2",
  "processAppID": "2066.9ab0d0c6-d92c-4355-9ed5-d8a05",
  "processAppName": "Hiring Sample",
  "processAppAcronym": "HSS"
}

```

Step 6:

To create Rest API , we find the required information of BPD Id, Snapshot Id, Branch Id and Process Application Id from Exposed process of Process API.

Note:

In Parameters field, passing parameters in JSON format.

```
{"customer":{"name":"Karan","age":30,"salary":120000,"city":"Chennai"}}
```

BPD ID

```

    type: "process",
    itemID: "25.2ece1c25-137c-4ca9-9c11-eb4eb7294d47",
    itemReference: "25.2ece1c25-137c-4ca9-9c11-eb4eb7294d47",
    processAppID: "2066.6c0a6c0a-fe44-422f-96be-e700e9fefcd1",
    processAppName: "Hiring Sample Application",
    processAppAcronym: "HSAM",
    snapshotID: "2064.1c2e52e8-2e13-4e85-b0e4-4f9695516ff1",
    snapshotCreatedOn: "2019-04-10T05:57:28Z",
    display: "Rest Call",
    tip: true,
    branchID: "2063.58d922bd-8b0b-40a5-87b0-06eca31727ec",
    branchName: "Main",
    startURL: "/rest/bpm/wle/v1/process?action=start&bpdId=25.2ece1c25-137c-4ca9-9c11-eb4eb7294d47&branchId=2063.58d922bd-8b0b-40a5-87b0-06eca31727ec",
    isDefault: false,
    isMobileReady: false,
    ID: "2015.1714"
  },

```

BPD Name, Which is exposed to All User, here we can find BPD details

Business Process Manager

REST API Tester

Call Tester

Service Runner

Result Type:

JSON (Javascript format)

Select API Call:

Process API

Retrieve Model

External Activity Model

Start Process

Current State

Evaluate JavaScript

Update Due Date

Suspend Process

Resume Process

Start Process

Starts a new BPD Instance

Method:

POST

BPD ID:

Business process definition ID. You can use t

Snapshot ID:

Snapshot ID for a particular process. Begins

Branch ID:

The branch ID. Begins with "2063.xxx..."

Process Application ID:

The process app ID begins with "2066.xxx". \

Parameters:

A string that contains a JSON expression tha

We can find the required information of BPD Id, Snapshot Id, Branch Id & Process Application Id from Exposed Process of Process API.

Passing Data in Json format:

{"customer":{"name":"Karan","age":30,"salary":12000,"city":"Chennai"}}

Step 7:

After filling all the fields(ref below picture), click the Execute Call button.

Using displayed URL in screen, place the url in eclipse (java project) for scheduling the CRON job.

Result Type:

JSON (Javascript format)

Select API Call:

REST interface for federated IBM BPM res

REST interface for federated IBM BPM res

REST interface for BPEL-related process i

REST Interface for BPEL-related Task Res

Business Process Manager REST APIs

Process API

Retrieve Model

External Activity Model

BPD ID:

25.2ece1c25-137c-4ca9-9c11-eb4eb7294d47

Snapshot ID:

Snapshot ID for a particular process. Begins

Branch ID:

2063.58d922bd-8b0b-40a5-87b0-06eca31727ec

Process Application ID:

2066.6c0a6c0a-fe44-422f-96be-e700e9fec1

Parameters:

{ "customer": { "name": "Karan", "age": 30, "sal

Mode:

A string indicating whether the process shoul

Request: https://bpm85.9444/rest/bpm/wle/v1/process?action=start&bpdid=25.2ece1c25-137c-4ca9-9c11-eb4eb7294d47&branchid=2063.58d922bd-8b0b-40a5-87b0-06eca31727ec&processAppId=2066.6c0a6c0a-fe44-422f-96be-e700e9fec1¶ms=%7B%22customer%22%3A%7B%22name%22%3A%22Karan%22%2C%22age%22%3A30%2C%22salary%22%3A120000%22%7D%7D

Method: POST

Status: 200 - OK

Header:

date - Wed, 10 Apr 2019 06:02:15 GMT
content-encoding - gzip
x-powered-by - Servlet/3.0
transfer-encoding - chunked
content-language - en-US
bpm_generic_header - SERVED
cache-control - no-cache, no-store, max-age=0
content-type - application/json

Result:

```

{
  "status": "200",
  "data": {
    "creationTime": "2019-04-10T06:02:16Z",
    "description": null,
    "richDescription": null,
    "executionState": "Active",
    "state": "STATE_RUNNING",
    "lastModificationTime": "2019-04-10T06:02:16Z",
    "name": "Rest Call:1856",
    "piid": "1856",
    "caseFolderServerName": null,
    "processTemplateID": "25.2ece1c25-137c-4ca9-9c11-eb4eb7294d47",
    "processTemplateName": "Rest Call"
  }
}

```

Step 8:

Consuming Rest API url & Scheduling Cron Job in Java Application, For that we need to create Java Project in Eclipse and add the below code in your class.

Note: Instead of using 9444 port number need to use 9081 in URL (otherwise it will show SSL certificates Error)

```

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.ProtocolException;

import java.net.URL;

import java.util.Date;

import java.util.Timer;

import java.util.TimerTask;

public class RunScheduler extends TimerTask {

    public void invokeURL() {

        System.out.println(Thread.currentThread().getName() + " Cron
Job has executed successfully " + new Date());
    }
}

```

```

try {

    //Using java.net.HttpURLConnection to fire and handle HTTP
requests

    //url - Here need to use Rest-API URL, which is created from
https://ibpm85:9444/bpmrest-ui/ console.

    URL url = new
URL("http://ibpm85:9081/rest/bpm/wle/v1/process?"

    + "action=start&bpdId=25.2ecelc25-137c-
4ca9-9c11-eb4eb7294d47"

    + "&branchId=2063.58d922bd-8b0b-40a5-87b0-
06eca31727ec"

    + "&processAppId=2066.6c0a6c0a-fe44-422f-
96be-e700e9fefcd1"

    + "&params=%7B%22customer%22%3A%7B%22name
%22%3A%22Karan%22%2C%22age"

    + "%22%3A30%2C%22salary%22%3A120000%2C
%22city%22%3A%22Chennai%22%7D%7D"

    + "&parts=all");

    HttpURLConnection con = (HttpURLConnection)
url.openConnection();

    //Set the method for the URL request
con.setRequestMethod("POST");

    //Sets the general request property,
setRequestProperty(String key,String value)

    //key - the keyword by which the request is known
(e.g., "Authorization" "Accept").

    //value - the value associated with it.

con.setRequestProperty("Authorization", "Basic
a2F0aGlyOnNhcmFzdTEw");

    //Set the DoOutput flag to true if you intend to use the URL

```

connection for output, false if not. The default is false.

```
        con.setDoOutput(true);

        //Gets the status code from an HTTP response message.
        System.out.println("Response Code:" + con.getResponseCode());

        //Gets the HTTP response message from a server.
        System.out.println("Response Message:" +
con.getResponseMessage());
    }

    catch (ProtocolException e) {
        e.printStackTrace();
    }

    catch (MalformedURLException e) {
        e.printStackTrace();
    }

    catch (IOException e) {
        e.printStackTrace();
    }

}

@Override

//Invoking the run() method from main thread
public void run() {

    // Invoke the invokeURL method
    invokeURL();

}

public static void main(String[] args){
```



```

RunScheduler rs=new RunScheduler();

//Timer class provide method call that are used by thread
to schedule a task, such as running a block of code after some regular instant
of time

Timer t=new Timer();

//scheduleAtFixedRate method is used to schedule the specified
task for repeated fixed-rate execution, beginning after the specified delay.

//scheduleAtFixedRate(TimerTask task,long delay,long period)

t.scheduleAtFixedRate(rs, 0,5*1000);

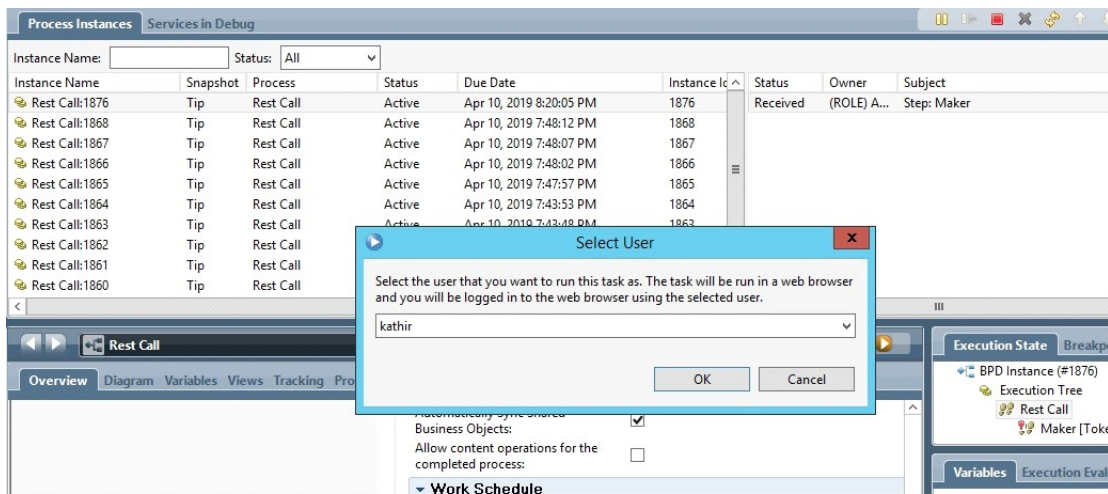
}

}

```

Step 9:

For every 5 seconds Process Instance created using CRON Job Scheduler in java.



Step 10:

After selecting the task, It will display the parameters what are having in the Rest API URL.

Round BPM New Tab http:// java URLC

← → ↻ ⚠ Not secure https://ibpm85.eidikointernal.c

Name

Age

Salary

City

Submit