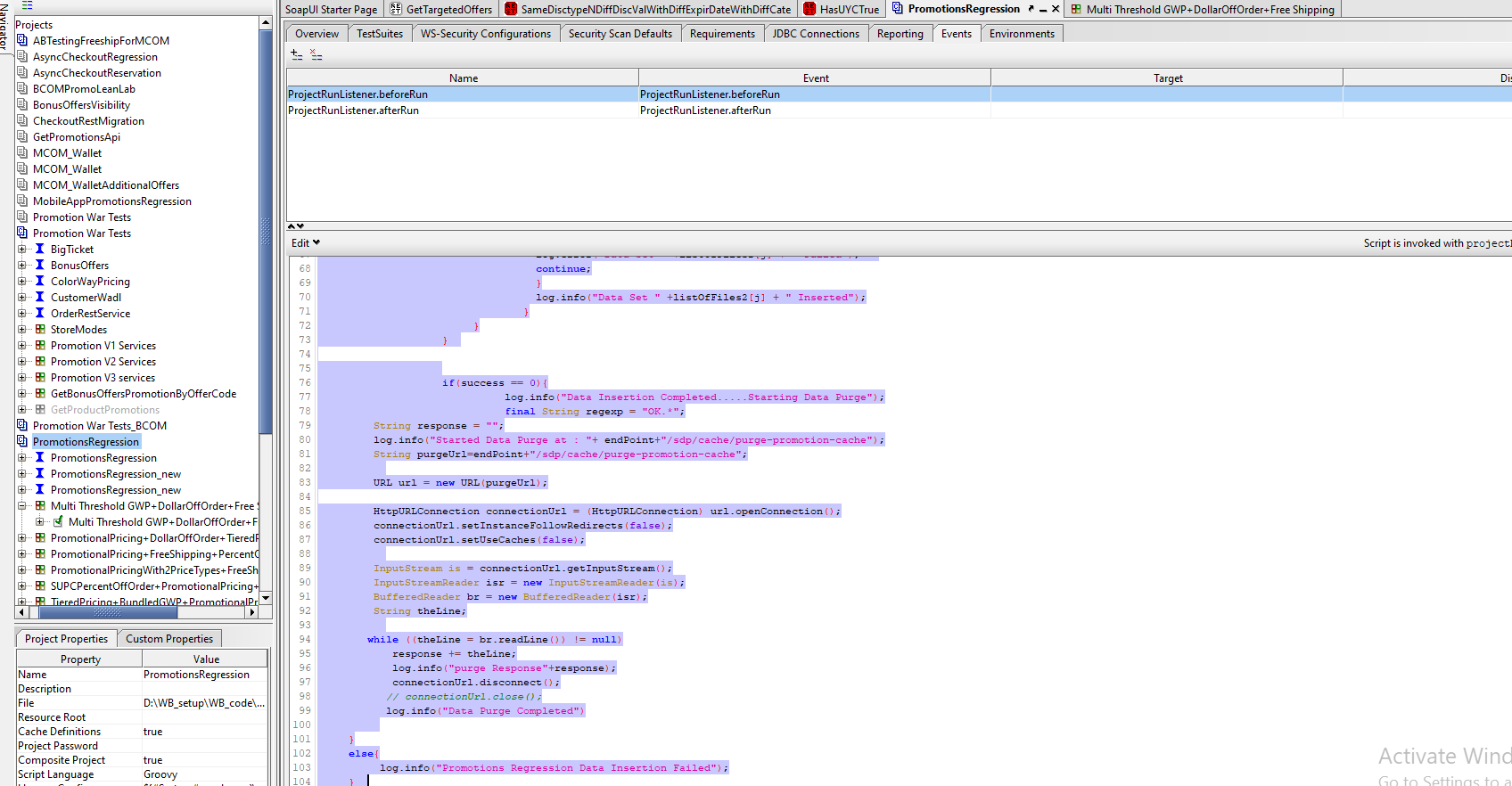
===setup scripts in events=====



import java.io.FileInputStream;

import java.sql.Connection;

import java.sql.DriverManager;

import com.ibatis.common.jdbc.ScriptRunner;

def projectDir = context.expand('${projectDir}');

String configFile=projectRunner.project.getPropertyValue("configFile")+".properties";

String configFilePath =projectDir+"/PromotionRegression\_BCOM/"+configFile;

log.info("configFilePath====================="+configFilePath);

Properties props = new Properties()

File propsFile = new File(configFilePath);

log.info("props file="+propsFile)

props.load(propsFile.newDataInputStream())

String endPoint =props.getProperty("endPoint");

context.endPoint=props.getProperty("endPoint");

String dbURL=props.getProperty("dbURL");

String driver1=props.getProperty("driver");

String customerEndPoint = props.getProperty("customerEndPoint");

log.info("customerEndPoint="+customerEndPoint);

log.info("DBURL="+dbURL+"\nEndPoint="+endPoint);

endPoint=endPoint.substring(0,endPoint.size()-1);

if(customerEndPoint!=null){

customerEndPoint=customerEndPoint.substring(0,customerEndPoint.size()-1);

}

projectRunner.project.setPropertyValue("endPoint",endPoint);

projectRunner.project.setPropertyValue("dbURL",dbURL);

projectRunner.project.setPropertyValue("driver",driver1);

projectRunner.project.setPropertyValue("customerEndPoint",customerEndPoint);

int success=0;

Class.forName(driver1)

try{

context.jdbcConnection = DriverManager.getConnection(dbURL);

}

catch(Exception e){

log.info("Error while getting connection:::"+e);

}

log.info("Connection=="+context.jdbcConnection)

String []promotionScriptFolders = ["/PromotionsData"];

String promotionScriptsPath = projectDir+"/PromotionRegression\_BCOM";

log.info("promotionScriptsPath"+promotionScriptsPath)

log.info("Promotions Regression Scripts Path="+promotionScriptsPath)

for (int i = 0; i < promotionScriptFolders.length; i++) {

File folder2= new File(promotionScriptsPath+promotionScriptFolders[i]);

log.info(folder2);

File[] listOfFiles2 = folder2.listFiles();

Arrays.sort(listOfFiles2);

for (int j = 0; j < listOfFiles2.length; j++){

log.info("File is :"+listOfFiles2[j]);

if(!listOfFiles2[j].toString().contains(".svn")){

try{

ScriptRunner sr2 = new ScriptRunner(context.jdbcConnection, false, true);

Reader reader2 = new BufferedReader(new FileReader((listOfFiles2[j])));

sr2.runScript(reader2);

}

catch(Exception e){

log.error("ERROR2"+e)

success=1;

log.error("Data Set " +listOfFiles2[j] + " Failed");

continue;

}

log.info("Data Set " +listOfFiles2[j] + " Inserted");

}

}

}

if(success == 0){

log.info("Data Insertion Completed.....Starting Data Purge");

final String regexp = "OK.\*";

String response = "";

log.info("Started Data Purge at : "+ endPoint+"/sdp/cache/purge-promotion-cache");

String purgeUrl=endPoint+"/sdp/cache/purge-promotion-cache";

URL url = new URL(purgeUrl);

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

connectionUrl.setInstanceFollowRedirects(false);

connectionUrl.setUseCaches(false);

InputStream is = connectionUrl.getInputStream();

InputStreamReader isr = new InputStreamReader(is);

BufferedReader br = new BufferedReader(isr);

String theLine;

while ((theLine = br.readLine()) != null)

response += theLine;

log.info("purge Response"+response);

connectionUrl.disconnect();

// connectionUrl.close();

log.info("Data Purge Completed")

}

else{

log.info("Promotions Regression Data Insertion Failed");

}