package longi.bom.BOMCreateProcessService;

import com.alibaba.fastjson.JSONArray;

import com.alibaba.fastjson.JSONObject;

import com.dassault\_systemes.platform.restServices.RestService;

import com.matrixone.apps.domain.util.ContextUtil;

import matrix.db.JPO;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.web.bind.annotation.RequestBody;

import javax.json.Json;

import javax.json.JsonObjectBuilder;

import javax.servlet.http.HttpServletRequest;

import javax.ws.rs.Consumes;

import javax.ws.rs.POST;

import javax.ws.rs.Path;

import javax.ws.rs.Produces;

import javax.ws.rs.core.Context;

import javax.ws.rs.core.MediaType;

import javax.ws.rs.core.Response;

import java.util.HashMap;

import java.util.Map;

/\*\*

\* @author wwy

\* @Description:

\* @date 2023/1/5 13:55

\*/

@Path("/BOM")

public class LGiM\_BOMCreateProcessServiceBase extends RestService {

private final static Logger interfaceLogger = LoggerFactory.getLogger("module\_interface");

@POST // 调用方式

@Path("/SearchCode")

@Consumes({MediaType.APPLICATION\_JSON}) // 输入格式

@Produces({MediaType.APPLICATION\_JSON}) // 输出格式

public Response searchCode(@Context HttpServletRequest request, @RequestBody String requestJson) throws Exception {

System.out.println("searchCode.." + requestJson);

interfaceLogger.info("编码搜索请求入参："+requestJson);

matrix.db.Context context = ContextUtil.getAnonymousContext();

JsonObjectBuilder resultJsonObj = Json.createObjectBuilder();

try {

JSONObject reqJsonObj = JSONObject.parseObject(requestJson);

String codeId = reqJsonObj.getString("codeId") == null ? "" : reqJsonObj.getString("codeId");

Map argsMap = new HashMap();

argsMap.put("codeId", codeId);

resultJsonObj = (JsonObjectBuilder) JPO.invoke(context, "LGiM\_BOMCreateProcess", null, "searchCode", JPO.packArgs(argsMap), JsonObjectBuilder.class);

resultJsonObj.add("state", "0");

System.out.println("searchCode..end");

interfaceLogger.info("编码搜索请求返回："+resultJsonObj.toString());

return Response.status(200).entity(resultJsonObj.build().toString()).build();

} catch (Exception e) {

e.printStackTrace();

resultJsonObj.add("state", "1");

resultJsonObj.add("msg", "PLM\_Code数据获取接口异常：" + e);

System.out.println("searchCode..catch");

interfaceLogger.info("编码搜索请求是失败返回："+resultJsonObj.toString());

return Response.status(500).entity(resultJsonObj.build().toString()).build();

}

}

@POST // 调用方式

@Path("/CreateBOMNewManagementProcess")

@Consumes({MediaType.APPLICATION\_JSON}) // 输入格式

@Produces({MediaType.APPLICATION\_JSON}) // 输出格式

public Response createBOMNewManagementProcess(@Context HttpServletRequest request, @RequestBody String requestJson) throws Exception {

System.out.println("createBOMNewManagementProcess.." + requestJson);

interfaceLogger.info("BON新建管理流程表单创建请求入参："+requestJson);

matrix.db.Context context = ContextUtil.getAnonymousContext();

JsonObjectBuilder respJsonObj = Json.createObjectBuilder();

try {

Map argsMap = new HashMap();

argsMap.put("requestJson", requestJson);

respJsonObj = (JsonObjectBuilder) JPO.invoke(context, "LGiM\_BOMCreateProcess", null, "createBOMNewManagementProcess", JPO.packArgs(argsMap), JsonObjectBuilder.class);

System.out.println("createBOMNewManagementProcess..end");

interfaceLogger.info("BON新建管理流程表单创建请求返回："+respJsonObj.toString());

return Response.status(200).entity(respJsonObj.build().toString()).build();

} catch (Exception e) {

e.printStackTrace();

respJsonObj.add("msg", "PLM\_BOM新建管理流程接口异常：" + e);

System.out.println("createBOMNewManagementProcess..catch");

interfaceLogger.info("BON新建管理流程表单创建请求失败返回："+respJsonObj.toString());

return Response.status(500).entity(respJsonObj.build().toString()).build();

}

}

@POST // 调用方式

@Path("/UpdateManagementProcessMessage")

@Consumes({MediaType.APPLICATION\_JSON}) // 输入格式

@Produces({MediaType.APPLICATION\_JSON}) // 输出格式

public Response updateManagementProcessMessage(@Context HttpServletRequest request, @RequestBody String requestJson) throws Exception {

System.out.println("updateManagementProcessMessage.." + requestJson);

interfaceLogger.info("BPM表单内容更新请求入参："+requestJson);

matrix.db.Context context = ContextUtil.getAnonymousContext();

JsonObjectBuilder respJsonObj = Json.createObjectBuilder();

try {

Map argsMap = new HashMap();

argsMap.put("requestJson", requestJson);

respJsonObj = (JsonObjectBuilder) JPO.invoke(context, "LGiM\_BOMCreateProcess", null, "updateFormData", JPO.packArgs(argsMap), JsonObjectBuilder.class);

System.out.println("updateManagementProcessMessage..end");

interfaceLogger.info("BPM表单内容更新请求返回："+respJsonObj.toString());

return Response.status(200).entity(respJsonObj.build().toString()).build();

} catch (Exception e) {

e.printStackTrace();

respJsonObj.add("msg", "PLM\_BOM新建管理流程更新接口异常：" + e);

System.out.println("updateManagementProcessMessage..catch");

interfaceLogger.info("BPM表单内容更新请求失败返回："+respJsonObj.toString());

return Response.status(500).entity(respJsonObj.build().toString()).build();

}

}

@POST // 调用方式

@Path("/BOMInformationToERP")

@Consumes({MediaType.APPLICATION\_JSON}) // 输入格式

@Produces({MediaType.APPLICATION\_JSON}) // 输出格式

public Response BOMInformationToERP(@Context HttpServletRequest request, @RequestBody String requestJson) throws Exception {

System.out.println("BOMInformationToERP.." + requestJson);

interfaceLogger.info("流程结束编码下发ERP请求入参："+requestJson);

matrix.db.Context context = ContextUtil.getAnonymousContext();

JsonObjectBuilder respJsonObj = Json.createObjectBuilder();

try {

JSONObject reqJsonObj = JSONObject.parseObject(requestJson);

String pid = reqJsonObj.getString("pid");

String state = reqJsonObj.getString("state");

String psId = reqJsonObj.getString("psId");

JSONArray FMSNameList = reqJsonObj.getJSONArray("FMSNameList");

Map argsMap = new HashMap();

argsMap.put("pid", pid);

argsMap.put("psId", psId);

argsMap.put("state", state);

argsMap.put("FMSNameList", FMSNameList);

JPO.invoke(context, "LGiM\_BOMCreateProcess", null, "sendBOMInformationToERP", JPO.packArgs(argsMap), String.class);

respJsonObj.add("state", "0");

respJsonObj.add("msg", "success");

System.out.println("BOMInformationToERP..end");

interfaceLogger.info("流程结束编码下发ERP请求返回："+respJsonObj.toString());

return Response.status(200).entity(respJsonObj.build().toString()).build();

} catch (Exception e) {

e.printStackTrace();

respJsonObj.add("state", "1");

respJsonObj.add("msg", "PLM\_SendBOMToERP接口异常：" + e);

System.out.println("BOMInformationToERP..catch");

interfaceLogger.info("流程结束编码下发ERP请求失败返回："+respJsonObj.toString());

return Response.status(500).entity(respJsonObj.build().toString()).build();

}

}

}

必须要引用的类

import com.dassault\_systemes.platform.restServices.RestService;

这也是3DE的RestService服务接口的类，它可以根据requirst来获取context

日志类：

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;