**package**com.masiinc.outlet.gsp.templatewarefolds.service;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** cn.com.ebsi.platform.web.service.EbsiBaseService;

**import** com.masiinc.outlet.gsp.templatewarefolds.dao.TempLatewareFoldsDao;

@Service

**Public class** TempLatewareFoldsService **extends** EbsiBaseService

{

@Autowired

**private** TempLatewareFoldsDao dao;

**public** Map<String, Object> selectIndexList(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<String, Object>(0);

// 自己的业务逻辑 --开始

// 查询总数

**long** total = dao.selectIndexListTotal(params);

List<Map<String, Object>> list = **new** ArrayList<>(0);

**if** (0 < total)

{

list = dao.selectIndexList(params);

}

// 自己的业务逻辑 --结束

res.put("total", total);

res.put("rows", list);

// 返回分页数据

**return** res;

}

**public** Map<String, Object> selectdlglist(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<String, Object>(0);

**long** total = dao.selectdlglisttotal(params);

List<Map<String, Object>> list = **new** ArrayList<>(0);

**if** (0 < total)

{

list = dao.selectdlglist(params);

}

res.put("total", total);

res.put("rows", list);

**return** res;

}

**public** Map<String, Object> createRecord(Map<String, Object> params)

{

**long** code = 0;

params.put("mgg001", dao.selectnexval(params));

code = dao.createRecord(params);

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

**public** Map<String, Object> selectEditpageDefaultRecord(

Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<>(0);

res = dao.selectEditpageDefaultRecord(params);

**return** res;

}

**public** Map<String, Object> updateRecord(Map<String, Object> params)

{

**long** code = 0;

code = dao.updateRecord(params);

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

**public** Map<String, Object> deleteRecord(Map<String, Object> params)

{

**long** code = 0;

code = dao.deleteRecord(params);

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

}

**package**com.masiinc.outlet.gsp.storedhousevalid.service;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** cn.com.ebsi.platform.web.service.EbsiBaseService;

**import** com.masiinc.outlet.gsp.storedhousevalid.dao.StoredHouseValidDao;

@Service

**Public class** StoredHouseValidService **extends** EbsiBaseService

{

@Autowired

**private** StoredHouseValidDao dao;

**public** Map<String, Object> selectIndexList(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<String, Object>(0);

// 自己的业务逻辑 --开始

// 查询总数

**long** total = dao.selectIndexListTotal(params);

List<Map<String, Object>> list = **new** ArrayList<>(0);

**if** (0 < total)

{

list = dao.selectIndexList(params);

}

// 自己的业务逻辑 --结束

res.put("total", total);

res.put("rows", list);

// 返回分页数据

**return** res;

}

**public** Map<String, Object> selectEditpageDefaultRecord(

Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<>(0);

// 自己的业务逻辑 --开始

res = dao.selectEditpageDefaultRecord(params);

// 自己的业务逻辑 --结束

**return** res;

}

**public** Map<String, Object> createRecord(Map<String, Object> params)

{

**long** code = 0;

// 自己的业务逻辑 --开始

code = dao.createRecord(params);

// 自己的业务逻辑 --结束

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

**public** Map<String, Object> updateRecord(Map<String, Object> params)

{

**long** code = 0;

// 自己的业务逻辑 --开始

**long** isHave = dao.selectGoodIsInMG14(params);

**if** (isHave > 0)

{

code = dao.updateRecord(params);

} **else**

{

code = dao.createRecord(params);

}

// 自己的业务逻辑 --结束

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

}

**package**com.masiinc.outlet.gsp.storedhousevalid;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.servlet.http.HttpServletRequest;

**import** net.sf.json.JSONObject;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

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

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** com.masiinc.common.MasiBaseController;

**import** com.masiinc.outlet.gsp.storedhousevalid.service.StoredHouseValidService;

@Controller

@RequestMapping("/storedhousevalid")

**publicclass**StoredHouseValidController**extends** MasiBaseController

{

@Autowired

**private** StoredHouseValidService service;

@RequestMapping("index.html")

**public** String index(HttpServletRequest request, Map<String, Object> params)

{

**this**.authentication(request, params);

**return** retMapper("storedhousevalid");

}

@RequestMapping("selectIndexList.html")

@ResponseBody

**public** String selectIndexList(HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

// 获得用户信息

**this**.authentication(request, params);

// 分页

**this**.pagingCalcu(request, params);

params.put("AAA008", request.getParameter("AAA008"));

params.put("AAA014", request.getParameter("AAA014"));

params.put("BAA504", request.getParameter("BAA504"));

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectIndexList(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.fromObject(res).toString();

}

@RequestMapping("selectToEditRecord")

@ResponseBody

**public** String selectEditpageDefaultRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectEditpageDefaultRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.fromObject(res).toString();

}

@RequestMapping("saveRecord.html")

@ResponseBody

**public** String createRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.createRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.fromObject(res).toString();

}

@RequestMapping("editById.html")

@ResponseBody

**public** String updateRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.updateRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.fromObject(res).toString();

}

}

**package**com.masiinc.outlet.gsp.searchgoodscattered;

**import** java.io.ByteArrayOutputStream;

**import** java.io.UnsupportedEncodingException;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**import** javax.servlet.ServletOutputStream;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** net.sf.json.JSONObject;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

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

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** cn.com.ebsi.platform.common.util.Utility;

**import** cn.com.ebsi.platform.exception.EbsiException;

**import** com.masiinc.common.MasiBaseController;

**import** com.masiinc.common.report.JasperReportControl;

**import** com.masiinc.common.report.MasiJRDataSourceFactory;

**import** com.masiinc.outlet.gsp.searchgoodscattered.service.SearchGoodScatteredService;

**import** com.masiinc.utility.MasiUtil;

/\*\*

\* 商品拆零查询

\* <描述>

\* **@author**<a href="mailTo:mengkai@masiinc.com">EYEDATA</a>

\* **@time** 2017年5月8日 下午5:44:54

\* **@version** 1.0

\* **@since**jdk 1.7.0

\*/

@Controller

@RequestMapping("/searchgoodscattered")

**publicclass**SearchGoodScatteredController**extends** MasiBaseController

{

@Autowired

**private** SearchGoodScatteredService service;

**privatefinalstatic** String *JR\_MAIN\_DATASOURCE* = "jrMainDataSource";

/\*\*

\* [具体使用时功能描述] 访问功能首页(页面)

\* **@return**

\* **@see** index(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("index.html")

**public** String index(HttpServletRequest request, Map<String, Object> params)

{

**this**.authentication(request, params);

**return** retMapper("searchgoodscattered");

}

/\*\*

\* [具体使用时功能描述] ajax访问功能首页(分页列表)

\* **@return**

\* **@see** selectIndexList(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("selectIndexList.html")

@ResponseBody

**public** String selectIndexList(HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

// 获得用户信息

**this**.authentication(request, params);

// 分页

**this**.pagingCalcu(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectIndexList(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

@RequestMapping("selecttotal")

@ResponseBody

**public** String selecttotal(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selecttotal(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* 打印报表

\* **@param** model

\* **@param** request

\* **@return**

\*/

@RequestMapping(value = "/selectreport.html", method = RequestMethod.*GET*)

**public** String report(Model model, HttpServletRequest request)

{

**try**

{

// 获取请求中的参数并放到Model中

Map<String, Object> paramMap = MasiUtil.*getParamsByRequest*(request);

Map<String, Object> params = Utility.*toMapByJson*(String

.*valueOf*(paramMap.get("json")));

model.addAllAttributes(params); // 报表参数集

**this**.authentication(request, params);

List<Map<String, Object>> dataList = service.getListdata(params);

Map<String, Object> exportInfo = **new** HashMap<>();

exportInfo.put("PAA621", "拆零商品");

exportInfo.put("PAA623", "1");

exportInfo.put("paa624", "searchgoodscattered.jrxml");

exportInfo.put("paa625", "1");

exportInfo.put("paa626", "04");

exportInfo.put("PAA627", "拆零商品");

exportInfo.put(*JR\_MAIN\_DATASOURCE*,

MasiJRDataSourceFactory.*getDataSource*(dataList));

model.addAllAttributes(exportInfo);

**return**"reportView";

} **catch** (Exception ex)

{

Logger.*getLogger*(JasperReportControl.**class**.getName()).log(

Level.*SEVERE*, **null**, ex);

}

**returnnull**;

}

/\*\*

\* 导出功能

\*/

@RequestMapping("selectexport.html")

**public** Object export(HttpServletRequest request,

HttpServletResponse response) **throws** EbsiException,

UnsupportedEncodingException

{

Map<String, Object> params = **new** HashMap<>();

**this**.authentication(request, params);

params.putAll(Utility.*toMapByJson*(String.*valueOf*(MasiUtil

.*getParamsByRequest*(request).get("json"))));

**if** (**null** == params.get("exportType"))

{

**thrownew** EbsiException("MSO-71010");

}

**try**

{

ServletOutputStream outputStream = response.getOutputStream(); // 向浏览器输入的流

Map<String, Object> servleMap = service.export(params,

params.get("exportType").toString());

String fileName = **new** String(servleMap.get("fileName").toString()

.getBytes("UTF-8"), "ISO-8859-1");// 文件名

ByteArrayOutputStream buffere = (ByteArrayOutputStream) servleMap

.get("buffere"); // 文件保存的反冲区域

**if** (servleMap.get("exportType").equals("1"))

{

response.setContentType("application/vnd.ms-Excel"); // 文件类型

response.setHeader("Content-disposition",

"attachment;filename=" + fileName + ".xls"); // 设置头信息

} **else**

{

response.setContentType("application/vnd.ms-PDF");

response.setHeader("Content-disposition",

"attachment;filename=" + fileName + ".pdf");

}

response.setContentLength(buffere.size());// 设置文件大小

buffere.writeTo(outputStream); // 将缓存区的信息写入 输出流

outputStream.flush();

} **catch** (Exception e)

{

e.printStackTrace();

}

**returnnull**;

}

}

**package**com.masiinc.outlet.gsp.searchgoodscattered.service;

**import** java.io.ByteArrayOutputStream;

**import** java.math.BigDecimal;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.LinkedList;

**import** java.util.List;

**import** java.util.Map;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** cn.com.ebsi.platform.common.util.Utility;

**import** cn.com.ebsi.platform.web.service.EbsiBaseService;

**import** com.masiinc.common.warehous.tools.CodeTool;

**import** com.masiinc.outlet.gsp.searchgoodscattered.dao.SearchGoodScatteredDao;

**import** com.masiinc.utility.DataEnum;

**import** com.masiinc.utility.ExportUtil;

**import** com.masiinc.utility.MasiUtil;

/\*\*

\* 商品拆零查询

\* <描述>

\* **@author**<a href="mailTo:mengkai@masiinc.com">EYEDATA</a>

\* **@time** 2017年5月8日 下午5:44:54

\* **@version** 1.0

\* **@since** jdk 1.7.0

\*/

@Service

@Transactional

**publicclass**SearchGoodScatteredService**extends** EbsiBaseService

{

@Autowired

**private** SearchGoodScatteredDao dao;

**privatestaticfinal** Long rows = Long.valueOf(3000);

/\*\*

\* [具体使用时功能描述] ajax访问功能首页(分页列表)

\* **@param** params

\* **@return**

\* **@see** selectIndexList(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> selectIndexList(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<String, Object>(0);

// 自己的业务逻辑 --开始

// 查询总数

**long** total = dao.selectIndexListTotal(params);

List<Map<String, Object>> list = **new** LinkedList<>();

**if** (0 < total)

{

list = dao.selectIndexList(params);

**for** (Map<String, Object> map : list)

{

map.put("daa422",

**null** == map.get("daa422") ? 0 : MasiUtil.toBigDecimal(

map.get("daa422").toString(), DataEnum.DETAILS));

map.put("daa423",

**null** == map.get("daa423") ? 0 : MasiUtil.toBigDecimal(

map.get("daa423").toString(), DataEnum.DETAILS));

map.put("daa424",

**null** == map.get("daa424") ? 0 : MasiUtil.toBigDecimal(

map.get("daa424").toString(), DataEnum.DETAILS));

}

}

// 自己的业务逻辑 --结束

res.put("total", total);

res.put("rows", list);

// 返回分页数据

**return** res;

}

**public** Map<String, Object> selecttotal(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<>(0);

Map<String, Object> map = dao.selecttotal(params);

BigDecimal num = BigDecimal.ZERO;

BigDecimal daa423 = BigDecimal.ZERO;

BigDecimal daa424 = BigDecimal.ZERO;

**if** (Utility.notEmpty(map))

{

num = num.add(MasiUtil.toBigDecimal(map.get("daa422").toString()));

daa423 = daa423.add(MasiUtil.toBigDecimal(map.get("daa423")

.toString()));

daa424 = daa424.add(MasiUtil.toBigDecimal(map.get("daa424")

.toString()));

}

res.put("num", num); // 数量总计

res.put("daa423", daa423); // 单价

res.put("daa424", daa424); // 金额

**return** res;

}

**public** List<Map<String, Object>> getListdata(Map<String, Object> params)

{

List<Map<String, Object>> datasList = **new** ArrayList<>();

params.put("rows", rows);

**int** count = 0;

**for** (**long** page = 0; page + 1 > page; page++)

{

**long** start = rows \* page; // 分页查询开始行号

params.remove("page");

params.put("page", start);

List<Map<String, Object>> dataList = dao.selectIndexList(params);

**for** (Map<String, Object> map : dataList)

{

map.put("daa425",

CodeTool.getCodeName("USERID", map.get("daa425")));

map.put("daa422",

**null** == map.get("daa422") ? 0 : MasiUtil.toBigDecimal(

map.get("daa422").toString(), DataEnum.DETAILS));

map.put("daa423",

**null** == map.get("daa423") ? 0 : MasiUtil.toBigDecimal(

map.get("daa423").toString(), DataEnum.DETAILS));

map.put("daa424",

**null** == map.get("daa424") ? 0 : MasiUtil.toBigDecimal(

map.get("daa424").toString(), DataEnum.DETAILS));

}

datasList.addAll(dataList);

count += dataList.size();

**if** (count > 50000 || rows > dataList.size())

{

**break**;

}

}

**return** datasList;

}

**public** Map<String, Object> export(Map<String, Object> params,

String exportType)

{

// 创建表格标题

Map<String, Object> titles = **new** LinkedHashMap<>();

titles.put("aaa009", "商拆零编码");

titles.put("aaa008", "商品编码");

titles.put("aaa014", "商品名称");

titles.put("aaa017", "规格");

titles.put("aaa010", "单位");

titles.put("aca002", "批次");

titles.put("daa422", "拆零数量");

titles.put("daa423", "拆零单价");

titles.put("daa424", "拆零金额");

titles.put("aaa214", "产地");

titles.put("aaa031\_n", "剂型");

titles.put("daa425name", "操作人");

titles.put("daa426", "操作时间");

// 创建导出工具实体

ExportUtil export = **new** ExportUtil(

(LinkedHashMap<String, Object>) titles, "拆零商品");

params.put("rows", rows);

**int** count = 0;

**for** (**long** page = 0; page + 1 > page; page++)

{

**long** start = rows \* page; // 分页查询开始行号

params.remove("page");

params.put("page", start);

List<Map<String, Object>> dataList = dao.selectIndexList(params);

**for** (Map<String, Object> map : dataList)

{

map.put("daa425name",

CodeTool.getCodeName("USERID", map.get("daa425")));

map.put("daa422",

**null** == map.get("daa422") ? 0 : MasiUtil.toBigDecimal(

map.get("daa422").toString(), DataEnum.DETAILS));

map.put("daa423",

**null** == map.get("daa423") ? 0 : MasiUtil.toBigDecimal(

map.get("daa423").toString(), DataEnum.DETAILS));

map.put("daa424",

**null** == map.get("daa424") ? 0 : MasiUtil.toBigDecimal(

map.get("daa424").toString(), DataEnum.DETAILS));

}

export.addData(dataList);

count += dataList.size();

**if** (count > 60000 || rows > dataList.size())

{

**break**;

}

}

ByteArrayOutputStream buffere = **null**;

**if** (exportType.equals("1"))

{

buffere = export.exportToExcel();

} **elseif** (exportType.equals("2"))

{

buffere = export.exportToPDF();

}

Map<String, Object> res = **new** HashMap<String, Object>();

res.put("buffere", buffere);

res.put("fileName", "拆零商品" + Utility.get19BitCurTimeAPP());

res.put("exportType", exportType);

**return** res;

}

}

/\*\*

\* **@filename** TempController.java

\* **@time** 2016-6-16 上午9:38:00

\* **@author** ZQQ

\*/

**package**com.masiinc.outlet.gsp.physicrecall;

**import** java.io.ByteArrayOutputStream;

**import** java.io.UnsupportedEncodingException;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**import** javax.servlet.ServletOutputStream;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** net.sf.json.JSONArray;

**import** net.sf.json.JSONObject;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.ui.Model;

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

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** cn.com.ebsi.platform.common.util.Utility;

**import** cn.com.ebsi.platform.exception.EbsiException;

**import** com.masiinc.common.MasiBaseController;

**import** com.masiinc.common.report.JasperReportControl;

**import** com.masiinc.common.report.MasiJRDataSourceFactory;

**import** com.masiinc.outlet.gsp.physicrecall.service.PhysicRecallService;

**import** com.masiinc.utility.MasiUtil;

/\*\*

\* 药品召回登记

\* **@author**<a href="mailTo:mengkai@masiinc.com">EYEDATA</a>

\* **@time** 2016-8-1 下午2:29:31

\* **@version** 1.0

\* **@since**jdk 1.7.0

\*/

@Controller

@RequestMapping("/physicrecall")

**publicclass**PhysicRecallController**extends** MasiBaseController

{

@Autowired

**private** PhysicRecallService service;

**privatefinalstatic** String *JR\_MAIN\_DATASOURCE* = "jrMainDataSource";

/\*\*

\* [具体使用时功能描述]

\* 访问功能首页(页面)

\* **@return**

\* **@see** index(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("index.html")

**public** String index(HttpServletRequest request, Map<String, Object> params)

{

**this**.authentication(request, params);

**return** retMapper("physicrecall");

}

/\*\*

\* [具体使用时功能描述]

\* ajax访问功能首页(分页列表)

\* **@return**

\* **@see** selectIndexList(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("selectIndexList.html")

@ResponseBody

**public** String selectIndexList(HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

// 获得用户信息

**this**.authentication(request, params);

// 分页

**this**.pagingCalcu(request, params);

params.put("aaa008", request.getParameter("aaa008"));

params.put("aaa013", request.getParameter("aaa013"));

params.put("mgg082S", request.getParameter("mgg082S"));

params.put("mgg082E", request.getParameter("mgg082E"));

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectIndexList(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* [具体使用时功能描述]

\* ajax访问功能首页(打开首页显示在首页的文字)

\* **@return**

\* **@see** selectIndexInfo(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("selectIndexInfo")

@ResponseBody

**public** String selectIndexInfo(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectIndexInfo(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* [具体使用时功能描述]

\* ajax访问打开编辑页面获取数据(默认显示的数据)

\* **@param** params

\* **@return**

\* **@see** selectEditpageDefaultContent(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("selectToEditRecord")

@ResponseBody

**public** String selectEditpageDefaultRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectEditpageDefaultRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* [具体使用时功能描述]

\* ajax保存数据(此方法目标为创建数据,创建同时也可更新数据)

\* **@param** params

\* **@return**

\* **@see** createData(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("saveRecord.html")

@ResponseBody

**public** String createRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.createRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* [具体使用时功能描述]

\* ajax查询批次号

\* **@param** params

\* **@return**

\* **@see** createData(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("selectCallNext.html")

@ResponseBody

**public** String selectCallNext(HttpServletRequest request)

{

List<Map<String, Object>> result = **new** ArrayList<>();

Map<String, Object> params = **new** HashMap<>();

**this**.authentication(request, params);

**this**.pagingCalcu(request, params);

**try**

{

result = service.selectCallNext(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONArray.*fromObject*(result).toString();

}

/\*\*

\* [具体使用时功能描述]

\* ajax更新数据

\* **@param** request

\* **@param** params

\* **@return**

\* **@see** updateDate(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("editById.html")

@ResponseBody

**public** String updateRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.updateRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* [具体使用时功能描述]

\* ajax删除数据

\* **@param** request

\* **@param** params

\* **@see** deleteRecord(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("delById.html")

@ResponseBody

**public** String deleteRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.deleteRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

// =====================================

// 方法名规范:

// 查询select 开头

// 添加create 开头

// 修改update 开头

// 删除delete 开头

// 自定义带url的方法 ---开始

// ...

/\*\*

\* [具体使用时功能描述]

\* ajax访问功能首页(分页列表)

\* **@return**

\* **@see** selectIndexList(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("selectCallDrug.html")

@ResponseBody

**public** String selectCallDrug(HttpServletRequest request, String q)

{

Map<String, Object> params = **new** HashMap<>();

// 获得用户信息

**this**.authentication(request, params);

// 分页

**this**.pagingCalcu(request, params);

Map<String, Object> res = **new** HashMap<>();

params.put("value", q);

**try**

{

res = service.selectCallDrug(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* 导出功能

\*/

@RequestMapping("selectexport.html")

**public** Object export(HttpServletRequest request,

HttpServletResponse response) **throws** EbsiException,

UnsupportedEncodingException

{

Map<String, Object> params = **new** HashMap<>();

**this**.authentication(request, params);

params.putAll(Utility.*toMapByJson*(String.*valueOf*(MasiUtil

.*getParamsByRequest*(request).get("json"))));

params = MasiUtil.*encodeURIdata*(params);

**if** (**null** == params.get("exportType"))

{

**thrownew** EbsiException("MSO-71010");

}

**try**

{

ServletOutputStream outputStream = response.getOutputStream(); // 向浏览器输入的流

Map<String, Object> servleMap = service.export(params,

params.get("exportType").toString());

String fileName = **new** String(servleMap.get("fileName").toString()

.getBytes("UTF-8"), "ISO-8859-1");// 文件名

ByteArrayOutputStream buffere = (ByteArrayOutputStream) servleMap

.get("buffere"); // 文件保存的反冲区域

**if** (servleMap.get("exportType").equals("1"))

{

response.setContentType("application/vnd.ms-Excel"); // 文件类型

response.setHeader("Content-disposition",

"attachment;filename=" + fileName + ".xls"); // 设置头信息

} **else**

{

response.setContentType("application/vnd.ms-PDF");

response.setHeader("Content-disposition",

"attachment;filename=" + fileName + ".pdf");

}

response.setContentLength(buffere.size());// 设置文件大小

buffere.writeTo(outputStream); // 将缓存区的信息写入 输出流

outputStream.flush();

} **catch** (Exception e)

{

e.printStackTrace();

}

**returnnull**;

}

// 打印商品退货库单

@RequestMapping(value = "/selectreport.html", method = RequestMethod.*GET*)

**public** String report(Model model, HttpServletRequest request)

{

**try**

{

// 获取请求中的参数并放到Model中

Map<String, Object> paramMap = MasiUtil.*getParamsByRequest*(request);

Map<String, Object> params = Utility.*toMapByJson*(String

.*valueOf*(paramMap.get("json")));

model.addAllAttributes(params); // 报表参数集

**this**.authentication(request, params);

params = MasiUtil.*encodeURIdata*(params);

Map<String, Object> dataList = service.getListdata(params);

Map<String, Object> exportInfo = **new** HashMap<>();

exportInfo.put("PAA621", "供货商人员管理");

exportInfo.put("PAA623", "1");

exportInfo.put("paa624", "physicrecall.jrxml");

exportInfo.put("paa625", "1");

exportInfo.put("paa626", "04");

exportInfo.put("PAA627", "供货商人员管理");

exportInfo.put(*JR\_MAIN\_DATASOURCE*, MasiJRDataSourceFactory

.*getDataSource*((List<Map<String, Object>>) dataList

.get("dataList")));

model.addAllAttributes(exportInfo);

model.addAllAttributes(dataList);

**return**"reportView";

} **catch** (Exception ex)

{

Logger.*getLogger*(JasperReportControl.**class**.getName()).log(

Level.*SEVERE*, **null**, ex);

}

**returnnull**;

}

/\*\*

\* 计算总金额数量

\* **@param** params

\* **@return**

\* **@see** selectIndexListSumMenoy(..)

\* **@time** 2017-5-10

\*/

@RequestMapping("/selecttotal.html")

@ResponseBody

**public** String selectIndexListSumMenoy(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectIndexListSumMenoy(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

}

/\*\*

\* **@filename** TempService.java

\* **@time** 2016-6-16 上午9:39:41

\* **@author** ZQQ

\*/

**package**com.masiinc.outlet.gsp.physicrecall.service;

**import** java.io.ByteArrayOutputStream;

**import** java.math.BigDecimal;

**import** java.util.ArrayList;

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.WeakHashMap;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** org.springframework.transaction.annotation.Transactional;

**import** cn.com.ebsi.platform.common.util.Utility;

**import** cn.com.ebsi.platform.web.service.EbsiBaseService;

**import** com.masiinc.outlet.gsp.physicrecall.dao.PhysicRecallDao;

**import** com.masiinc.utility.ExportUtil;

**import** com.masiinc.utility.MasiUtil;

/\*\*

\* 药品召回登记

\* **@author**<a href="mailTo:mengkai@masiinc.com">EYEDATA</a>

\* **@time** 2016-8-1 下午2:29:31

\* **@version** 1.0

\* **@since**jdk 1.7.0

\*/

@Service

@Transactional

**publicclass**PhysicRecallService**extends** EbsiBaseService

{

@Autowired

**private** PhysicRecallDao dao;

**privatestaticfinal** Long *rows* = Long.*valueOf*(1000);

/\*\*

\* [具体使用时功能描述]

\* ajax访问功能首页(分页列表)

\* **@param** params

\* **@return**

\* **@see** selectIndexList(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> selectIndexList(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<String, Object>(0);

// 自己的业务逻辑 --开始

// 查询总数

**long** total = dao.selectIndexListTotal(params);

List<Map<String, Object>> list = **new** ArrayList<>(0);

**if** (0 < total)

{

list = dao.selectIndexList(params);

**for** (Map<String, Object> map : list)

{

map.put("daa523",

**null** == map.get("daa523") ? 0 : MasiUtil

.*toBigDecimal*(map.get("daa523").toString()));

map.put("mgg081",

**null** == map.get("mgg081") ? 0 : MasiUtil

.*toBigDecimal*(map.get("mgg081").toString()));

}

}

// 自己的业务逻辑 --结束

res.put("total", total);

res.put("rows", list);

// 返回分页数据

**return** res;

}

/\*\*

\* [具体使用时功能描述]

\* ajax访问功能首页(打开首页显示在首页的文字)

\* **@param** params

\* **@return**

\* **@see** selectIndexInfo(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> selectIndexInfo(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<>(0);

// 自己的业务逻辑 --开始

res = dao.selectIndexInfo(params);

// 自己的业务逻辑 --结束

**return** res;

}

/\*\*

\* [具体使用时功能描述]

\* ajax访问打开编辑页面获取数据(默认显示的数据)

\* **@param** params

\* **@return**

\* **@see** selectEditpageDefaultRecord(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> selectEditpageDefaultRecord(

Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<>(0);

// 自己的业务逻辑 --开始

res = dao.selectEditpageDefaultRecord(params);

// 自己的业务逻辑 --结束

**return** res;

}

/\*\*

\* [具体使用时功能描述]

\* ajax保存数据(此方法目标为创建数据,创建同时也可更新数据)

\* **@param** params

\* **@return**

\* **@see** createRecord(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> createRecord(Map<String, Object> params)

{

**long** code = 0;

// 自己的业务逻辑 --开始

code = dao.createRecord(params);

// 自己的业务逻辑 --结束

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

/\*\*

\* [具体使用时功能描述]

\* ajax查询批次号

\* **@param** params

\* **@return**

\* **@see** createData(..)

\* **@time** 2016-6-16

\*/

**public** List<Map<String, Object>> selectCallNext(Map<String, Object> params)

{

List<Map<String, Object>> res = **new** ArrayList<>();

// 自己的业务逻辑 --开始

res = dao.selectCallNext(params);

// 自己的业务逻辑 --结束

**return** res;

}

/\*\*

\* [具体使用时功能描述]

\* ajax更新数据

\* **@param** params

\* **@return**

\* **@see** updateRecord(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> updateRecord(Map<String, Object> params)

{

**long** code = 0;

// 自己的业务逻辑 --开始

code = dao.updateRecord(params);

// 自己的业务逻辑 --结束

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

/\*\*

\* [具体使用时功能描述]

\* ajax删除数据

\* **@param** params

\* **@return**

\* **@see** deleteRecord(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> deleteRecord(Map<String, Object> params)

{

**long** code = 0;

// 自己的业务逻辑 --开始

code = dao.deleteRecord(params);

// 自己的业务逻辑 --结束

Map<String, Object> res = **new** HashMap<>(0);

res.put("code", code);

**return** res;

}

// =====================================

// 方法名规范:

// 查询select 开头

// 添加create 开头

// 修改update 开头

// 删除delete 开头

// 自定义方法 ---开始

// ...

/\*\*

\* [具体使用时功能描述]

\* ajax访问功能首页(分页列表)

\* **@param** params

\* **@return**

\* **@see** selectIndexList(..)

\* **@time** 2016-6-16

\*/

**public** Map<String, Object> selectCallDrug(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<String, Object>(0);

// 自己的业务逻辑 --开始

// 查询总数

**long** total = dao.selectCallDrugTotal(params);

List<Map<String, Object>> list = **new** ArrayList<>(0);

**if** (0 < total)

{

list = dao.selectCallDrug(params);

}

// 自己的业务逻辑 --结束

res.put("total", total);

res.put("rows", list);

// 返回分页数据

**return** res;

}

// 导出

**public** Map<String, Object> export(Map<String, Object> params,

String exportType)

{

// 创建表格标题

Map<String, Object> titles = **new** LinkedHashMap<>();

titles.put("aaa008", "药品编码");

titles.put("aaa013", "通用名称");

titles.put("aca002", "批次号");

titles.put("aaa017", "召回金额");

titles.put("aaa017", "规格");

titles.put("aaa211", "厂家");

titles.put("mgg082", "召回日期");

titles.put("mgg081", "召回数量");

titles.put("mgg083", "召回原因");

// 创建导出工具实体

ExportUtil export = **new** ExportUtil(

(LinkedHashMap<String, Object>) titles, "供货商人员管理");

params.put("rows", *rows*);

**for** (**long** page = 0; page + 1 > page; page++)

{

**long** start = *rows* \* page;

// page = rows \* page; // 分页查询开始行号

params.remove("page");

params.put("page", start);

List<Map<String, Object>> dataList = dao.ExportIndexList(params);

export.addData(dataList);

**if** (*rows*> dataList.size())

{

**break**;

}

}

ByteArrayOutputStream buffere = **null**;

**if** (exportType.equals("1"))

{

buffere = export.exportToExcel();

} **elseif** (exportType.equals("2"))

{

buffere = export.exportToPDF();

}

Map<String, Object> res = **new** HashMap<String, Object>();

res.put("buffere", buffere);

res.put("fileName", "供货商人员管理");

res.put("exportType", exportType);

**return** res;

}

// 打印

**public** Map<String, Object> getListdata(Map<String, Object> params)

{

Map<String, Object> res = **new** HashMap<>();

params.put("rows", *rows*);

**for** (**long** page = 0; page + 1 > page; page++)

{

**long** start = *rows* \* page;

// page = rows \* page; // 分页查询开始行号

params.remove("page");

params.put("page", start);

List<Map<String, Object>> dataList = dao.ExportIndexList(params);

BigDecimal price = BigDecimal.*ZERO*;

**for** (Map<String, Object> map : dataList)

{

map.put("daa523",

MasiUtil.*toBigDecimal*(map.get("daa523").toString()));

map.put("mgg081",

MasiUtil.*toBigDecimal*(map.get("mgg081").toString()));

price = price.add(MasiUtil.*toBigDecimal*(map.get("daa523")

.toString()));

}

res.put("pricename", Utility.*toRmb*(price));

res.put("dataList", dataList);

**if** (*rows*> res.size())

{

**break**;

}

}

**return** res;

}

// 计算总金额，数量

**public** Map<String, Object> selectIndexListSumMenoy(

Map<String, Object> params)

{

Map<String, Object> map = **new** WeakHashMap<>();

Map<String, Object> res = **new** WeakHashMap<>();

res = dao.selectIndexListSumMenoy(params);

map.put("mgg081",

res == **null** ? BigDecimal.*ZERO* : MasiUtil.*toBigDecimal*(res.get(

"mgg081").toString()));

map.put("daa523",

res == **null** ? BigDecimal.*ZERO* : MasiUtil.*toBigDecimal*(res.get(

"daa523").toString()));

**return** map;

}

}

/\*\*

\* **@filename** TempController.java

\* **@time** 2016-6-16 上午9:38:00

\* **@author** ZQQ

\*/

**package**com.masiinc.outlet.gsp.personnelfoster;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.annotation.Resource;

**import** javax.servlet.http.HttpServletRequest;

**import** net.sf.json.JSONObject;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

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

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** cn.com.ebsi.platform.common.util.Utility;

**import** cn.com.ebsi.platform.exception.EbsiException;

**import** com.masiinc.common.MasiBaseController;

**import** com.masiinc.common.log.IUserSysOperaLog;

**import** com.masiinc.outlet.gsp.facilitymanage.service.FacilityManageService;

**import** com.masiinc.outlet.gsp.hygrothermograph.service.HygrothermographService;

**import** com.masiinc.outlet.gsp.medicalregister.service.MedicalRegisterService;

**import** com.masiinc.outlet.gsp.personnelfoster.service.PersonnelFosterService;

**import** com.masiinc.outlet.template.service.TempService;

/\*\*

\*

\* 员工培训登记

\* **@author**<a href="mailTo:mengkai@masiinc.com">EYEDATA</a>

\* **@time** 2016-8-1 下午4:44:10

\* **@version** 1.0

\* **@since**jdk 1.7.0

\*/

@Controller

@RequestMapping("/personnelfoster")

**publicclass**PersonnelFosterController**extends** MasiBaseController

{

@Autowired

**private** PersonnelFosterService service;

@Resource(name="logService")

**private** IUserSysOperaLog OperaLog;

/\*\*

\* [具体使用时功能描述]

\* 访问功能首页(页面)

\* **@return**

\* **@see** index(..)

\* **@time** 2016-6-16

\*/

@RequestMapping("index.html")

**public** String index(HttpServletRequest request, Map<String, Object> params)

{

**this**.authentication(request, params);

**return** retMapper("personnelfoster");

}

//初始化,查询全部的数据

@RequestMapping("selectindexsList.html")

@ResponseBody

**public** String selectList(@RequestParam("page") String pageStr, @RequestParam("rows") String rowsStr,HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

**this**.authentication(request, params);

params.put("rows", Long.*parseLong*(rowsStr));

params.put("page", (Long.*parseLong*(pageStr) - 1) \* Long.*parseLong*(rowsStr));

params.put("deptname", request.getParameter("deptname"));//部门名称

params.put("userid", request.getParameter("userid"));//用户编号

params.put("username", request.getParameter("username"));//员工姓名

params.put("mgg085", request.getParameter("mgg085"));//培训方式

params.put("starttime", request.getParameter("starttime"));//开始时间

params.put("stoptime", request.getParameter("stoptime"));//结束时间

Map<String, Object> res = **null**;

res = service.selectIndexList(params);

**return** JSONObject.*fromObject*(res).toString();

}

//登记界面的下拉框数据表格

@RequestMapping("selectdlglist.html")

@ResponseBody

**public** String sreachMemberInfo(@RequestParam("page") String pageStr, @RequestParam("rows") String rowsStr,

String q, HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

**this**.authentication(request, params);

params.put("rows", Long.*parseLong*(rowsStr));

params.put("page", (Long.*parseLong*(pageStr) - 1) \* Long.*parseLong*(rowsStr));

params.put("q", q);

Map<String, Object> res = service.selectdlglist(params);

**return** JSONObject.*fromObject*(res).toString();

}

@RequestMapping("savedatacreate.html")

@ResponseBody

**public** String createRecord(HttpServletRequest request, @RequestBody Map<String, Object> params) **throws** EbsiException

{

OperaLog.beginOperaLog("员工培训登记保存操作", "创建保存", request.getParameterMap(),params);

**this**.authentication(request, params);

JSONObject jsono = **new** JSONObject();

**try**

{

jsono.putAll(service.createRecord(params));

**int** succRows = jsono.getInt("code");

jsono.element("code", succRows);

**if** (succRows > 0)

{

jsono.element("msg", Utility.*getMessage*("MSO-71010", "保存"));

OperaLog.endOperaLog(request, Utility.*getMessage*("MSO-71005","员工培训登记保存操作|创建保存"), succRows);

} **else**

{

jsono.element("msg", Utility.*getMessage*("MSO-71011", "保存"));

}

} **catch** (Exception ex)

{

jsono.element("code", "-1");

jsono.element("msg", ex.getMessage());

OperaLog.endOperaLog(request, Utility.*getMessage*("MSO-71006","员工培训登记保存操作|创建保存"), ex);

}

**return** jsono.toString();

}

/\*\*

\* 删除一条数据

\* **@param** request

\* **@param** params

\* **@return**

\* **@throws** EbsiException

\*/

@RequestMapping("delById.html")

@ResponseBody

**public** String deleteRecord(HttpServletRequest request, @RequestBody Map<String, Object> params) **throws** EbsiException

{

OperaLog.beginOperaLog("员工培训人员管理删除", "删除", request.getParameterMap(),params);

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.deleteRecord(params);

OperaLog.endOperaLog(request, Utility.*getMessage*("MSO-71005","员工培训人员管理|删除"), res);

} **catch** (Exception e)

{

e.printStackTrace();

OperaLog.endOperaLog(request, Utility.*getMessage*("MSO-71006","员工培训人员管理|删除"), e);

}

**return** JSONObject.*fromObject*(res).toString();

}

//打开查看界面

@RequestMapping("selectToEditRecord.html")

@ResponseBody

**public** String selectEditpageDefaultRecord(HttpServletRequest request, @RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectEditpageDefaultRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

/\*\*

\* 根据流水号修改一条数据

\* **@param** request

\* **@param** params

\* **@return**

\* **@throws** EbsiException

\*/

@RequestMapping("editById.html")

@ResponseBody

**public** String updateRecord(HttpServletRequest request, @RequestBody Map<String, Object> params) **throws** EbsiException

{

OperaLog.beginOperaLog("员工培训修改", "修改", request.getParameterMap(),params);

**this**.authentication(request, params);

JSONObject jsono = **new** JSONObject();

**try**

{

jsono.putAll(service.updateRecord(params));

**int** succRows = jsono.getInt("code");

jsono.element("code", succRows);

**if** (succRows > 0)

{

jsono.element("msg", Utility.*getMessage*("MSO-71010", "保存"));

OperaLog.endOperaLog(request, Utility.*getMessage*("MSO-71005","员工培训修改|修改"), succRows);

} **else**

{

jsono.element("msg", Utility.*getMessage*("MSO-71011", "保存"));

}

} **catch** (Exception ex)

{

jsono.element("code", "-1");

jsono.element("msg", ex.getMessage());

OperaLog.endOperaLog(request, Utility.*getMessage*("MSO-71006","员工培训修改|修改"), ex);

}

**return** jsono.toString();

}

}

**package**com.masiinc.outlet.gsp.firstbusinessmanage;

**import** java.util.HashMap;

**import** java.util.Map;

**import** javax.annotation.Resource;

**import** javax.servlet.http.HttpServletRequest;

**import** net.sf.json.JSONObject;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

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

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.ResponseBody;

**import** cn.com.ebsi.platform.common.util.Utility;

**import** cn.com.ebsi.platform.exception.EbsiException;

**import** com.masiinc.common.MasiBaseController;

**import** com.masiinc.common.log.IUserSysOperaLog;

**import** com.masiinc.outlet.gsp.firstbusinessmanage.service.FirstBusinessManageService;

@Controller

@RequestMapping("/firstbusinessmanage")

**publicclass**FirstBusinessManageController**extends** MasiBaseController

{

@Autowired

**private** FirstBusinessManageService service;

@Resource(name = "logService")

**private** IUserSysOperaLog OperaLog;

@RequestMapping("index.html")

**public** String index(HttpServletRequest request, Map<String, Object> params)

{

**this**.authentication(request, params);

**return** retMapper("firstbusinessmanage");

}

@RequestMapping("selectIndexList.html")

@ResponseBody

**public** String selectIndexList(HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

// 获得用户信息

**this**.authentication(request, params);

// 分页

**this**.pagingCalcu(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

params.put("aaa350", request.getParameter("aaa350"));

params.put("aaa352", request.getParameter("aaa352"));

params.put("aaa354", request.getParameter("aaa354"));

params.put("aaa361", request.getParameter("aaa361"));

res = service.selectIndexList(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

@RequestMapping("selectIndexInfo")

@ResponseBody

**public** String selectIndexInfo(@RequestBody Map<String, Object> params,

HttpServletRequest request)

{

**this**.authentication(request, params);

Map<String, Object> res = service.selectIndexInfo(params);

**return** JSONObject.*fromObject*(res).toString();

}

@RequestMapping("selectToEditRecord")

@ResponseBody

**public** String selectEditpageDefaultRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.selectEditpageDefaultRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

@RequestMapping("saveRecord.html")

@ResponseBody

**public** String createRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params) **throws** EbsiException

{

OperaLog.beginOperaLog("首营企业信息", "成功/失败", request.getParameterMap(),

params);

**this**.authentication(request, params);

JSONObject jsono = **new** JSONObject();

**try**

{

params.put("datasign", 1);

jsono.putAll(service.createRecord(params));

**int** succRows = jsono.getInt("code");

jsono.element("code", succRows);

**if** (succRows > 0)

{

jsono.element("msg", Utility.*getMessage*("MSO-71010", "保存"));

OperaLog.endOperaLog(request,

Utility.*getMessage*("MSO-71005", "首营企业信息|成功/失败"), jsono);

} **else**

{

jsono.element("msg", Utility.*getMessage*("MSO-71011", "保存"));

}

} **catch** (Exception e)

{

jsono.element("code", "-1");

jsono.element("msg", e.getMessage());

OperaLog.endOperaLog(request,

Utility.*getMessage*("MSO-71006", "首营企业信息|成功/失败"), e);

}

/\* return JSONObject.fromObject(res).toString(); \*/

**return** jsono.toString();

}

@RequestMapping("update.html")

@ResponseBody

**public** String update(HttpServletRequest request,

@RequestBody Map<String, Object> params) **throws** EbsiException

{

OperaLog.beginOperaLog("首营企业信息", "成功/失败", request.getParameterMap(),

params);

**this**.authentication(request, params);

JSONObject jsono = **new** JSONObject();

**try**

{

params.put("datasign", 1);

jsono.putAll(service.update(params));

**int** succRows = jsono.getInt("code");

jsono.element("code", succRows);

**if** (succRows > 0)

{

jsono.element("msg", Utility.*getMessage*("MSO-71010", "修改"));

OperaLog.endOperaLog(request,

Utility.*getMessage*("MSO-71005", "首营企业信息|成功/失败"), jsono);

} **else**

{

jsono.element("msg", Utility.*getMessage*("MSO-71011", "修改"));

}

} **catch** (Exception e)

{

jsono.element("code", "-1");

jsono.element("msg", e.getMessage());

OperaLog.endOperaLog(request,

Utility.*getMessage*("MSO-71006", "首营企业信息|成功/失败"), e);

}

/\* return JSONObject.fromObject(res).toString(); \*/

**return** jsono.toString();

}

@RequestMapping("editById.html")

@ResponseBody

**public** String updateRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params)

{

**this**.authentication(request, params);

Map<String, Object> res = **new** HashMap<>();

**try**

{

res = service.updateRecord(params);

} **catch** (Exception e)

{

e.printStackTrace();

}

**return** JSONObject.*fromObject*(res).toString();

}

@RequestMapping("delById.html")

@ResponseBody

**public** String deleteRecord(HttpServletRequest request,

@RequestBody Map<String, Object> params) **throws** EbsiException

{

OperaLog.beginOperaLog("首营企业删除操作", "成功/失败", request.getParameterMap(),

params);

JSONObject jsono = **new** JSONObject();

**this**.authentication(request, params);

jsono.putAll(service.deleteRecord(params));

**int** succRows = jsono.getInt("code");

jsono.element("code", succRows);

**if** (succRows > 0)

{

jsono.element("msg", Utility.*getMessage*("MSO-71030", "删除"));

OperaLog.endOperaLog(request,

Utility.*getMessage*("MSO-71005", "首营企业删除操作|成功/失败"), jsono);

} **else**

{

jsono.element("msg", Utility.*getMessage*("MSO-71031", "删除"));

OperaLog.endOperaLog(request,

Utility.*getMessage*("MSO-71006", "首营企业删除操作|成功/失败"), jsono);

}

**return** jsono.toString();

}

@RequestMapping("selectdlglist.html")

@ResponseBody

**public** String sreachMemberInfo(@RequestParam("page") String pageStr,

@RequestParam("rows") String rowsStr, String q,

HttpServletRequest request)

{

Map<String, Object> params = **new** HashMap<>();

**this**.authentication(request, params);

params.put("rows", Long.*parseLong*(rowsStr));

params.put("page",

(Long.*parseLong*(pageStr) - 1) \* Long.*parseLong*(rowsStr));

params.put("q", q);

Map<String, Object> res = service.selectdlglist(params);

**return** JSONObject.*fromObject*(res).toString();

}

}