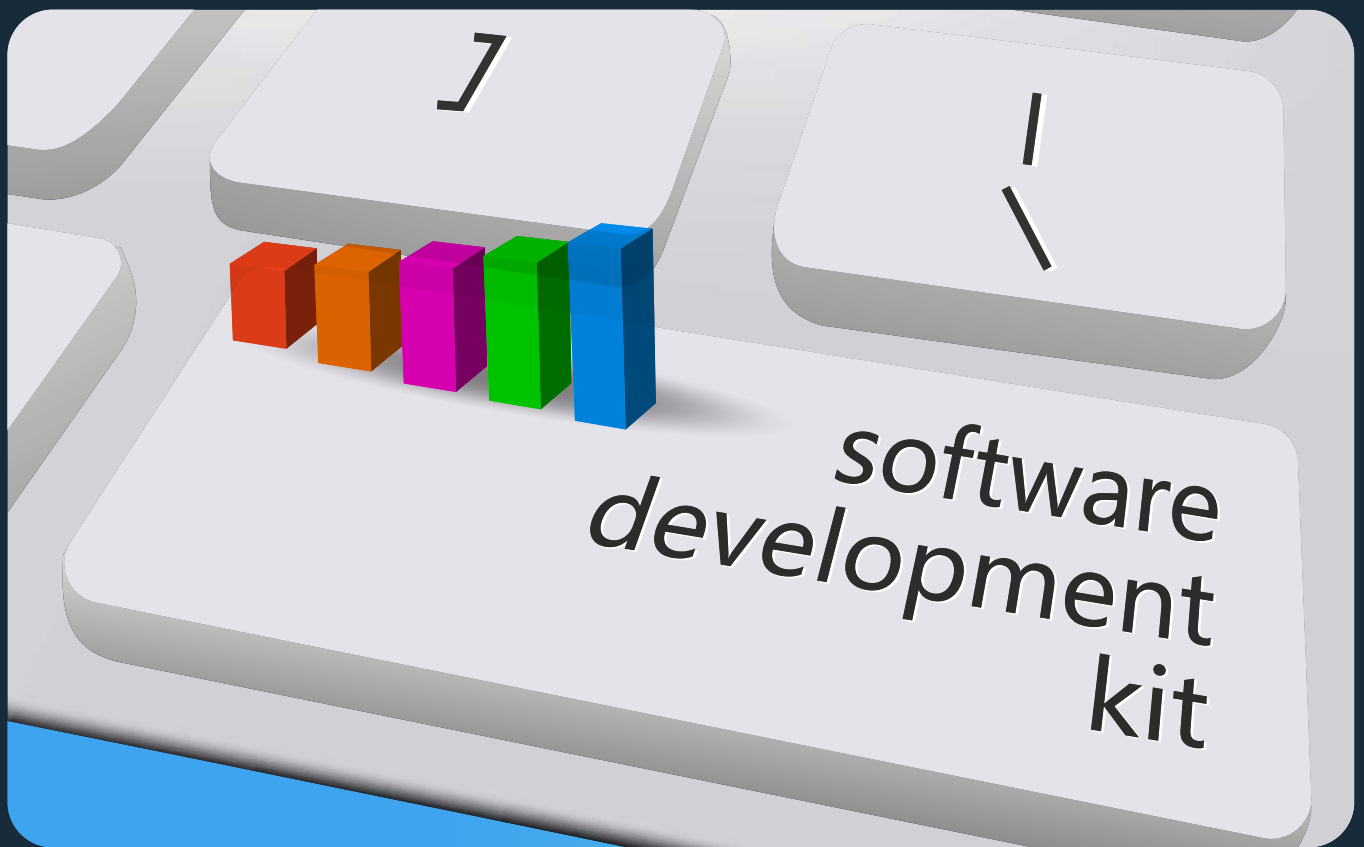


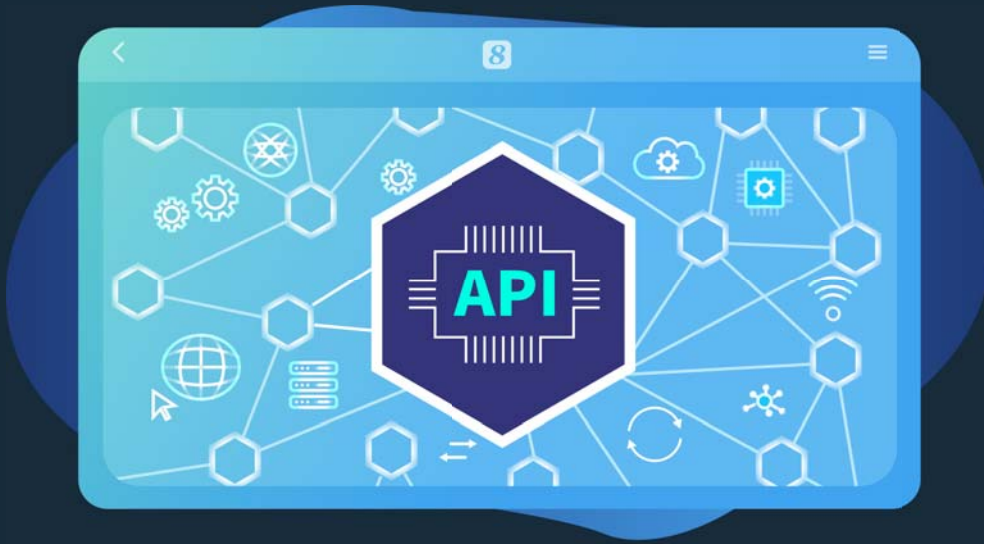
8Manage 软件开发工具包 (SDK)

加速软件开发和集成

SDK



8Manage 为您的团队提供 SDK，以将他们的应用程序集成到 8Manage 服务中。8Manage SDK 包括文档、应用程序编程接口 (API)、代码示例、库和 Java 开发工具包，以及您的团队可以学习如何集成到他们的应用程序中的教程。



应用程序编程接口 (API)

8Manage API 帮助团队利用 8Manage 实施来完成繁琐的扩展和二次开发工作，从而节省时间。这有助于减少开发人员需要创建的代码量，并促进 8Manage 平台的应用程序间实现更高的一致性。

8Manage API 可以控制对 8Manage 平台资源的访问。

代码库

8Manage 代码库是功能齐全的 8Manage 软件，开发人员可通过 8Manage API 进行访问。



文档

8Manage SDK 开发人员指南提供了一系列指导方针，帮助开发人员在 8Manage 平台上创建高质量、一致性高且易于维护的代码。

JDK

JAVA

DEVELOPMENT

KIT

Java 开发工具包

8Manage Java 开发工具包 (JDK) 是一个使用 Java 编程语言构建应用程序、小程序和组件的开发环境。该 JDK 集成了一系列有用的工具，可用于开发和测试采用 Java 编程语言编写并在 8Manage 平台上运行的程序。

代码示例

8Manage 代码示例作为初阶示范，用于说明执行特定任务的步骤并帮助开发人员快速入门。



教程视频

8Manage SDK 教程视频提供了一系列分步指导，讲解如何利用其对 8Manage 平台进行功能扩展或二次开发。

8Manage SDK

开发者指南

修订版 1.8

高亚科技（广州）有限公司

目 录

1	概述	1
2	Web Service 界面外部数据集成	1
3	SDK 定制功能	2
3.1	自定义角色权限	2
3.1.1	概述	2
3.1.2	实现方法	2
3.1.3	实现示例	4
3.1.4	注意事项	5
3.2	请求回调机制	5
3.2.1	概述	5
3.2.2	定制回调机制	6
3.2.3	常用辅助方法	9
3.2.3.1	错误信息	9
3.2.3.1.1	HTTP 请求上下文是否存在错误信息	9
3.2.3.1.2	保存错误信息到 HTTP 请求上下文	9
3.2.3.2	获取当前用户	9
3.2.3.3	获取当前时间	10
3.2.3.4	获取 Hibernate Session	10
3.2.3.5	获取页面权限信息	10
3.2.4	实现示例	10
3.2.4.1	在业务对象的列表页面增加字段	10
3.2.4.1.1	在采购计划列表页面增加显示审批人	10
3.2.4.1.2	在营销活动列表页面增加显示其它字段	12
3.2.4.1.3	在机会/策略列表页面增加显示其它字段	13
3.2.4.1.4	在合约列表页面增加显示其它字段	13
3.2.4.1.5	在内部用户列表页面增加显示其它字段	14
3.2.4.1.6	在客户列表页面增加显示其它字段	14
3.2.4.2	在业务对象的基本信息页面增加字段	14
3.2.4.2.1	在合同基本信息页面增加显示供应商字段	14
3.2.4.2.2	在营销活动基本信息页面增加显示其它字段	32
3.2.4.2.3	在机会/策略基本信息页面增加显示其它字段	33
3.2.4.3	同步业务对象信息到外部数据库	33
3.2.4.3.1	同步用户信息到外部数据库	33
3.2.5	8Manage 常用 actionPoint	38
3.3	用 8Manage 界面组件定制用户界面	43
3.3.1	概述	43
3.3.2	实现步骤	43
3.4	添加自定义菜单到系统菜单	44
3.4.1	概述	44
3.4.2	实现方法	44
3.5	客制化字段名	47
3.5.1	概述	47
3.5.2	实现方法	48
3.5.3	实现示例	48
3.5.4	注意事项	50
3.6	自定义代码表	50
3.6.1	概述	50
3.6.2	实现步骤	51
3.7	数据权限检查机制	52
3.7.1	概述	52
3.7.2	调用数据权限检查	52
3.7.3	类型列表	53
3.8	搭建 SDK 远程 debug 环境	54

3.8.1 概述	54
3.8.2 以远程 Debug 方式启动 8Manage	54
3.9 无刷新多文件上传及文档链接功能的使用	59
3.9.1 概述	59
3.9.2 使用步骤	59
3.10 通用审计跟踪记录功能的实现	63
3.10.1 概述	63
3.10.2 log 标准表结构	63
3.10.3 java 类及 hibernate 配置文件	64
3.10.4 用 SQL, trigger 把记录写到 log 表,及写资源文件 key-value 值	64
3.10.5 列表查询语句, 及表格 JSP 代码	67
3.11 发送自定义邮件	69
3.12 自定义查询条件	71
3.13 自定义论坛名称	72
4 8Manage 目录结构	73
4.1 8Manage 现有目录介绍	73
4.2 客制化目录建议	77
5 客制化代码部署方式	78
5.1 JSP 代码部署	78
5.2 JAVA 代码部署	78
5.3 配置文件部署	79

1 概述

本文档的目的是向开发人员介绍 8Manage SDK (软件开发工具), 包括支持需要编程的 8Manage 功能定制, 以及实现这些功能用到的方法和代码实例。

总体而言, 8Manage 支持两种定制系统功能的方式:

1. 基于 Web Service 的外部数据集成;
2. 利用 SDK 编程的 API 来实现功能定制。

本文档只提供基于 SDK 编程 API 来实现功能定制的指引和说明,

2 Web Service 界面外部数据集成

8Manage 支持外部应用系统通过 Web Service 界面将外部数据导入或在 8Manage 中创建数据,

例如, 用导入的.mpp 格式的项目文件中 8Manage 系统中创建新的项目记录。

请参考文档《8Manage Web Service 集成界面》了解如何实现外部应用系统与 8Manage 集成的详细操作说明。在 8Manage 的发布 CD 中的自述说明文档中的清单如包含此文档, 您即可在此发布 CD 中找到此文档。

3 SDK 定制功能

8Manage 支持通过 API 编程来实现功能定制，此节内容介绍如何通过 SDK 编程实现定制 8Manage 系统功能的默认逻辑，如何提高可用性，以及在数据模型和交换方面的特别要求。

3.1 自定义角色权限

3.1.1 概述

8Manage 通过为每个用户分配一个或多个系统“角色”来分配用户权限。每个角色都包含一组访问 8Manage 系统功能的权限选项。现有用于定义角色权限的权限选项也许不能满足用户的要求。

此时，可通过 8Manage 的 SDK 编程来实现对这些选项的定制。

此节内容介绍如何在 8Manage 中实现自定义角色权限。

3.1.2 实现方法

1) 把功能模块数据加入 func 表

8Manage 使用 func 表来存储系统角色权限信息。8Manage 中 func 表的表结构如下：

字段名	字段类型	允许空	描述
GroupDispOrder	int	否	功能模块组显示序号
GroupName	nvarchar(256)	否	功能模块组名称。应该是系统定义的多语言信息的 key 值。多语言信息的设置见下文中的例子。
FuncId	int	否	主键，功能模块 id。8Manage 保留 10000 之前的 FuncId，二次开发请从 10000 开始编号，以免冲突。
FuncType	varchar(11)	否	功能类型。取值为“VIEW”或

字段名	字段类型	允许空	描述
			“EDIT” 两种。
FuncDispOrder	int	否	功能模块显示序号
ResourceName	nvarchar(256)	否	功能模块名称。应该是系统定义的多语言信息的 key 值。多语言信息的设置见下文中的例子。

如果二次开发的模块可以归入 8Manage 的模块组中，您可以使用 8Manage 现有功能模块组。

8Manage 现有功能模块组名称的显示序号：

功能模块组显示序号 (GroupDispOrder)	功能模块组名称 (GroupName)	描述
0	display.menu.search.global	查询
10	display.menu.TOPMENU	顶部主菜单
15	name.func.topmenu.hr	人事管理
20	display.menu.ENTERPRISE	企业
25	display.menu.MEMBERSHIP.MANAGEMENT	会员管理
30	display.menu.version.productSystem	产品
40	display.menu.clientBusiness	客户业务
50	display.menu.supplierBusiness	供应商相关
90	display.menu.PROJECT	项目与支持
120	display.menu.DOCUMENT.LIBRARY	文档库
160	display.menu.SYSTEMADMIN	系统管理

2) 把功能模块角色权限数据加入 func_url 表

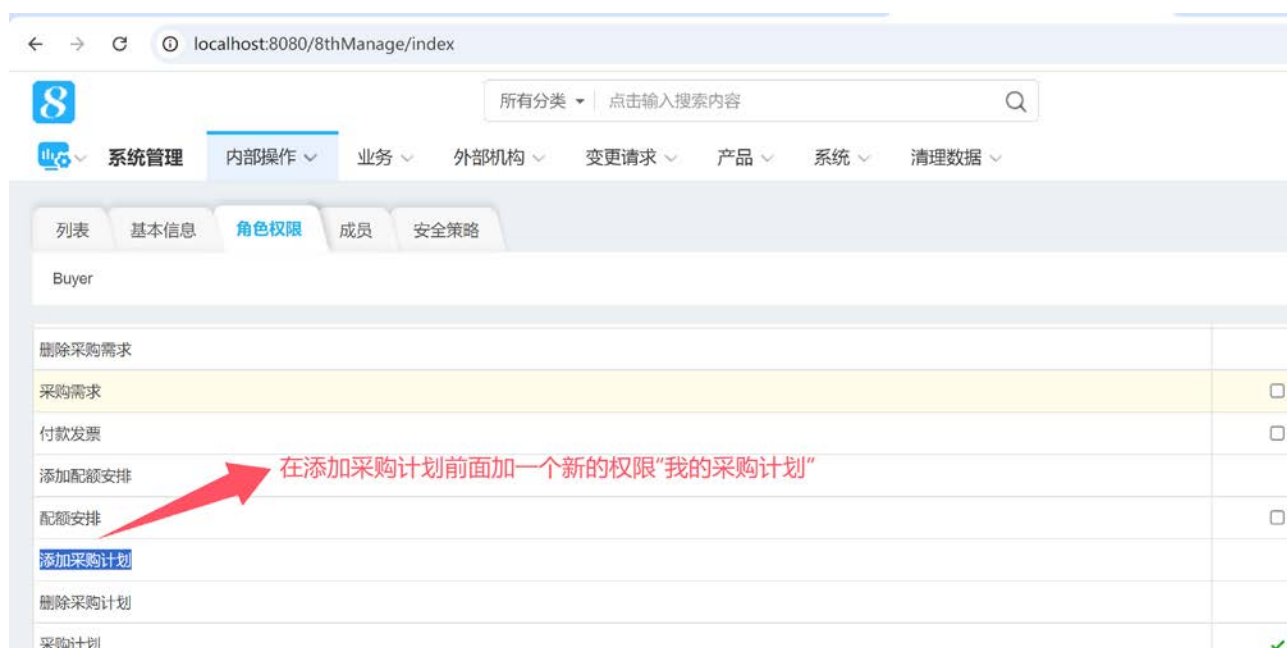
8Manage 使用 func_url 来存储系统中已经授权的 Request URI。8Manage 中的 func_url 表的表结

构如下：

字段名	字段类型	允许空	描述
FuncId	int	否	功能模块 id，关联到 Func 表的外键。
ActionName	nvarchar(64)	否	功能模块的 Request URI。如 Request URL http://cary:8080/8thManage/projectManagement/deal.do?method=listStrategy 中，Request URI 为 /projectManagement/deal.do
MethodName	nvarchar(64)	否	功能模块的方法名。如上述的 Request URL 中，MethodName 为 listStrategy。

3.1.3 实现示例

目标：在“供应商相关”角色权限组中的“添加采购计划”前面加入“我的采购计划”的角色权限数据。



1) 把功能模块数据加入 func 表

```
INSERT INTO func (GroupDispOrder, GroupName, FuncId, FuncType, FuncDispOrder,
ResourceName) VALUES (50, 'display.menu.supplierBusiness', 10000, 'VIEW', 19198,
'project.label.purchasePlan.newPurchasePlan.my');
```

```
INSERT INTO func (GroupDispOrder, GroupName, FuncId, FuncType, FuncDispOrder,
ResourceName) VALUES (50, 'display.menu.supplierBusiness', 10001, 'EDIT', 19199,
'project.label.purchasePlan.newPurchasePlan.my');
```

可通过以下 SQL 在 func 表中查询“供应商相关”和“添加采购计划”的 FuncDispOrder:

```
select * from func where funcId = 148
```

新增的“我的采购计划”角色权限数据的 FuncDispOrder 只要在这两者之间即可。

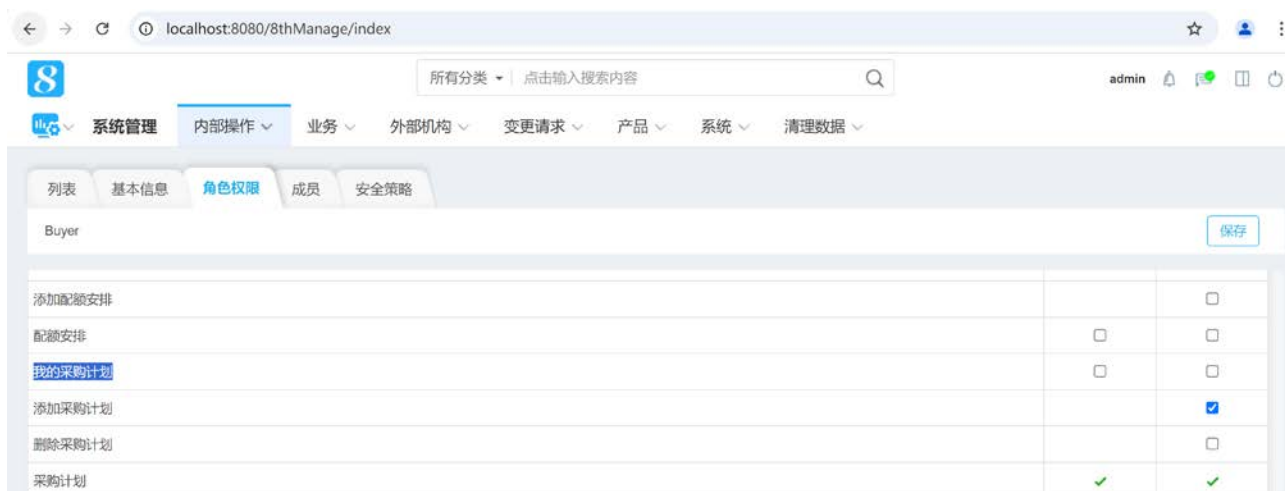
请记得在 `$(siteHome)\customize\resources` 目录下的所有资源文件中增加以下 Key-value 值:

```
project.label.purchasePlan.newPurchasePlan.my=我的采购计划 (这个值请按不同语言翻译)
```

2) 把功能模块角色权限数据加入 func_url 表

```
INSERT INTO func_url (FuncId, ActionName, MethodName) VALUES (10000, 'projectManagement/myPurchasePlan.do', 'doShowList');
```

最后的角色权限页面效果如下:



3.1.4 注意事项

8Manage 保留 10000 之前的 FuncId，二次开发请从 10000 开始编号，以免冲突。

3.2 请求回调机制

3.2.1 概述

由于可能存在客户化的需求，需要在服务器端附加处理。8Manage 系统支持在前台页面发送出请求后，在服务器端回调客户化处理的方法。例如，用户添加的自定义字段在保存的时候可同时作为数据库事务的一部分来保存到系统中。用户也可以修改在列表页面显示的字段。

此节内容介绍如何在 8Manage 中实现个性化回调机制。

3.2.2 定制回调机制

个性化回调配置的核心基于 CustomizedCallback.xml 配置文件。你可以在\${siteHome}/

customize/callback 下找到这个文件。其中\${siteHome}表示 8thManage 的主目录。

CustomizedCallback.xml 配置文件的格式如下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<customized>
  <actions>
    <action actionPoint="actionName|methodName" location="pre|post"
      sameTransaction="true|false" useForward="true|false"
      actionClass="CustomizedActionClass" actionMethod="CustomizedMethod" />
    <action actionPoint="actionName|methodName" location="pre|post"
      sameTransaction="true|false" useForward="true|false"
      actionClass="CustomizedActionClass" actionMethod="CustomizedMethod" />
  </actions>
</customized>
```

- **customized**

customized 节点为 CustomizedCallback.xml 的根结点。

该节点不可缺少。

- **actions**

actions 节点为 customized 根结点的直接子节点，表示所有个性化回调 action 配置。

该节点不可缺少。

- **action**

action 节点是 actions 节点的子节点，表示单个个性化回调配置。action 节点有 actionPoint、

location、sameTransaction、useForward、actionClass 和 actionMethod 共六个属性节点。下面是每

个属性节点的具体描述。

该节点不可缺少。

- **actionPoint**

actionPoint 属性表示个性化回调的加入点。它表示 8Manage 运行环境中的某个方法。也就是说，个性化回调将在 actionPoint 表示的某个方法运行时被回调执行。ActionPoint 的值的

8Manage 中常用的 actionPoint 请参考下文中 “8Manage 常用 actionPoint” 。

该属性不可缺少。

- **location**

location 属性表示个性化回调的切入点。它有两个值：pre 或者 post。

pre：表示个性化回调将在加入点 actionPoint 执行之前执行。

post：表示个性化回调将在加入点 actionPoint 执行之后执行。

该属性不可缺少。

- **sameTransaction**

sameTransaction 属性表示个性化回调是否跟加入点 actionPoint 属于同个数据库事务。它有两个值：true 或者 false：

1. true：表示个性化回调和加入点 actionPoint 属于同个数据库事务。这种情况下，个性化回调或者加入点 actionPoint 在执行过程中出现的错误都将造成两者的数据库事务回滚。
2. false：表示个性化回调和加入点 actionPoint 不属于同个数据库事务。这种情况下，个性化回调和加入点 actionPoint 在执行过程中出现的错误将不会相互影响对方的数据库事务。

8Manage 认为以下情况下执行过程中出现错误：

1. 存在向上抛出未捕捉的异常；
2. 调用 `CommonUtil.hasActionErrors(request)` 返回 `true`（关于该方法及相关方法见会详述）。

注意：

如果 `sameTransaction` 为 `true`，即用户想让加入点 `actionPoint` 和个性化回调处于同一个数据库事务，那么个性化回调代码在操作数据库时必须使用以下方法操作数据库：

使用 `SessionUtil.currentSession()` 返回 Hibernate 的 `Session` 对象进行数据库操作；

或者是使用 `SessionUtil.currentSession().connection()` 返回 JDBC connection 进行数据库操作。当用户这样使用时，请注意必须保证此行代码被 `Transaction` 所包着。

个性化回调过程中用户自己创建的 JDBC connection 不会跟加入点 `actionPoint` 处于同个数据库事务。这时，用户需要自己控制回调过程中的数据库事务。

该属性不可缺少。

- **useForward**

`useForward` 属性表示是否把返回页面重新指定为个性化回调方法中指定的页面。它有两个值：`true` 或者 `false`。

1. `true`：把个性化回调中指定的返回页面作为最终的返回页面。
2. `false`：保留加入点 `actionPoint` 中原来的返回页面作为最终的返回页面。

该属性不可缺少。

- **actionClass**

`actionClass` 属性表示个性化回调定制的具体类。这个类必须实现 `com.gearteks.test.customized.Callbackable` 接口。

该属性不可缺少。

- **actionMethod**

actionMethod 属性表示个性化回调定制的具体方法。这个方法入参列表必须如下：

```
ActionMapping mapping  
ActionForm form  
HttpServletRequest request  
HttpServletResponse response
```

该属性不可缺少。

3.2.3 常用辅助方法

3.2.3.1 错误信息

3.2.3.1.1 HTTP 请求上下文是否存在错误信息

请调用 `CommonUtil` 的以下方法：

```
public static boolean hasActionErrors(HttpServletRequest request)
```

3.2.3.1.2 保存错误信息到 HTTP 请求上下文

请调用 `CommonUtil` 的以下方法：

```
public static void saveErrors(HttpServletRequest request, ProcessErrors pes)  
public static void saveErrors(HttpServletRequest request, ActionErrors ae)  
public static void saveErrors(HttpServletRequest request, ActionMessages errors)
```

3.2.3.2 获取当前用户

请调用 `SessionUtil` 的以下方法：

```
public static User getCurrentUser(HttpServletRequest request)
```


3.2.3.3 获取当前时间

请调用 `DateUtil` 的以下方法：

```
public static Date getDBCurrentDate()
public static Timestamp getDBCurrentTimestamp()
```

3.2.3.4 获取 Hibernate Session

请调用 `SessionUtil` 的以下方法：

```
public static Session currentSession() throws HibernateException
```

3.2.3.5 获取页面权限信息

可通过以下方法来获取页面权限信息：

```
boolean readonly = StringUtil.toBoolean
(request.getAttribute(HtmlTagHelper.USE_POLICY_KEY), false);
```

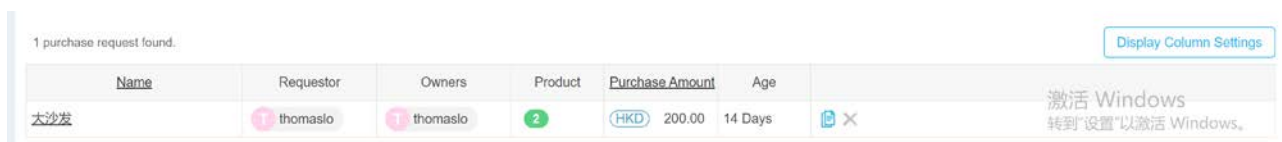
`readonly` 为 `true`，则表示当前页面权限为只读状态，否则为可编辑状态。

3.2.4 实现示例

3.2.4.1 在业务对象的列表页面增加字段

3.2.4.1.1 在采购计划列表页面增加显示审批人

现在采购计划列表页面如下：



Name	Requestor	Owners	Product	Purchase Amount	Age	
大沙发	thomaslo	thomaslo	2	HKD 200.00	14 Days	✕

我们需要在第六个字段 Ag 后面增加显示审批人：

1 purchase request found. Display Column Settings

Name	Requestor	Owners	Product	Purchase Amount	Age	Approvers
大沙发	1 thomaslo	1 thomaslo	2	(HKD) 200.00	14 Days	admin

激活 Windows
转到“设置”以激活 Windows。

步骤如下：

创建 `CustomizedOrderAction` 类。该类实现 `Callbackable` 接口：

```
package com.gearteks.test.customized;

import java.util.Collection;
import java.util.HashMap;
import java.util.Map;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.gearteks.framework.model.approval.Approval;
import com.gearteks.framework.process.assist.DisplayObjAssistProcess;
import com.gearteks.imanage.model.projmgmt.PurchasePlan;
import com.gearteks.imanage.model.quotation.Quotation;
import com.wisagetech.framework.callback.Callbackable;

public class CustomizedPurchasePlanAction implements Callbackable {
    /**
     *
     * @param mapping
     * @param form
     * @param request
     * @param response
     * @return
     * @throws Exception
     */
    public ActionForward doShowList(ActionMapping mapping, ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        Collection<PurchasePlan> planLst = (Collection<PurchasePlan>)
            request.getAttribute("planCol");
        if (planLst != null && !planLst.isEmpty()) {
```

```

Map approverMap = new HashMap();
for (PurchasePlan plan : planLst) {
    Approval appr = plan.getApproval();
    approverMap.put(plan.getPlanId(),
        DisplayObjAssistProcess.getDefaultToolTips(appr.getApprovers(), request));
}
request.setAttribute("approverMap", approverMap);
}
return null;
}
}

```

在 `purchasePlanListTable.jsp` 上显示审批人：

接收 `approverMap` 集合：

```
java.util.Map approverMap = (java.util.Map)request.getAttribute("approverMap");
```

并且在列表的 `Responsible person` 列后面增加显示审批人：

```

<gearTable:column titleKey="name.approval.approvers" sortable="false"
headerClass="sortable" >
    <%= approverMap.get(plan.getPlanId()) != null ? approverMap.get(plan.getPlanId()) :
HtmlTagHelper.EMPTY_DISPLAY_VALUE %>
</gearTable:column>

```

在 `/${siteHome}/customize/callback/CustomizedCallback.xml` 中增加以下配置：

```

<action actionPoint="purchasePlan|doShowList" location="post"
sameTransaction="true" useForward="false"
actionClass="com.gearteks.test.customized.CustomizedPurchasePlanAction"
actionMethod="doShowList" />

```

重启 8Manage。

3.2.4.1.2 在营销活动列表页面增加显示其它字段

限于文档篇幅有限，请参考“在订单列表页面增加显示审批人”。可以通过以下方法获取营销活动

集合列表：

```
Collection campaignLst = (Collection)request.getAttribute(Constants.LIST);
```

在`{siteHome}/customize/callback/CustomizedCallback.xml` 中增加类似以下配置：

```
<action actionPoint="campaign|listStrategy" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedCampaignAction"
        actionMethod="listStrategy" />
```

3.2.4.1.3 在机会/策略列表页面增加显示其它字段

限于文档篇幅有限，请参考“在订单列表页面增加显示审批人”。可以通过以下方法获取机会/策略

集合列表：

```
Collection strategyLst = (Collection)request.getAttribute(Constants.LIST);
```

在`{siteHome}/customize/callback/CustomizedCallback.xml` 中增加类似以下配置：

```
<action actionPoint="strategy|listStrategy" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
        actionMethod="listStrategy" />
```

3.2.4.1.4 在合约列表页面增加显示其它字段

限于文档篇幅有限，请参考“在订单列表页面增加显示审批人”。可以通过以下方法获取合约集合

列表：

```
Collection contractLst = (Collection)request.getAttribute(Constants.LIST);
```

在`{siteHome}/customize/callback/CustomizedCallback.xml` 中增加类似以下配置：

```
<action actionPoint="deal|listStrategy" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedDealAction"
        actionMethod="listStrategy" />
```

3.2.4.1.5 在内部用户列表页面增加显示其它字段

限于文档篇幅有限，请参考“在订单列表页面增加显示审批人”。可以通过以下方法获取内部用户

集合列表：

```
Collection userLst = (Collection)request.getAttribute(Constants. KEY_USERLIST);
```

在`/${siteHome}/customize/callback/CustomizedCallback.xml` 中增加类似以下配置：

```
<action actionPoint="user|doShowList" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedUserAction"
        actionMethod="doShowList" />
```

3.2.4.1.6 在客户列表页面增加显示其它字段

限于文档篇幅有限，请参考“在订单列表页面增加显示审批人”。可以通过以下方法获取客户集合

列表：

```
Collection clientLst = (Collection)request.getAttribute(Constants. KEY_COMPANYLIST);
```

在`/${siteHome}/customize/callback/CustomizedCallback.xml` 中增加类似以下配置：

```
<action actionPoint="company|doShowList" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedCompanyAction"
        actionMethod="doShowList" />
```

3.2.4.2 在业务对象的基本信息页面增加字段

3.2.4.2.1 在合同基本信息页面增加显示供应商字段

以下例子将演示在合同的基本信息页面增加显示“供应商”字段。

现在合同的基本信息页面中的“提议的客户合同”部分如下：

把“供应商”字段加入到 ID 编号之后，效果如下：

实现步骤如下：

1. 在数据库服务器（例如：carson）上创建存储供应商的表：

```
CREATE TABLE customized_members (
  ObjId          id NOT NULL,
  ObjType        type NOT NULL,
  memberType     type NOT NULL,
  memberObjId    id NOT NULL,
  memberObjType  type NOT NULL
);
go
```

```
ALTER TABLE customized_members ADD
    CONSTRAINT PK_customized_members PRIMARY KEY CLUSTERED (ObjId, ObjType,
memberType, memberObjId, memberObjType);
go

ALTER TABLE members ADD
    CONSTRAINT CK1_customized_members
    CHECK (ObjType in ('Deal'));
go

ALTER TABLE customized_members ADD
    CONSTRAINT CK2_customized_members
    CHECK (memberObjType in ('Company', 'Party', 'User'));
go
```

2. 创建 CustomizedMembers 类:

```
/**
 *
 */
package com.gearteks.test.customized.model;

import java.io.Serializable;

import org.apache.commons.lang.builder.EqualsBuilder;
import org.apache.commons.lang.builder.HashCodeBuilder;

import com.gearteks.framework.common.Constants;
import com.gearteks.framework.common.exception.GTSystemException;
import com.gearteks.framework.model.assist.DisplayObjInterFace;
import com.gearteks.framework.object.BusinessObject;
import com.gearteks.framework.process.CompanyProcess;
import com.gearteks.framework.process.UserProcess;

/**
 * @author ericchen
 *
 */
public class CustomizedMembers implements Comparable, Serializable, DisplayObjInterFace {
    private CustomizedMembersPK _comp_id;

    public static class CustomizedMembersPK implements Serializable {
        private Integer _objId;
```

```
private String _objType;
private String _memberType;
private Integer _memberObjId;
private String _memberObjType;

public CustomizedMembersPK() {
    ;
}

public CustomizedMembersPK(Integer objId, String objType, String memberType,
Integer memberObjId, String memberObjType) {
    this._objId = objId;
    this._objType = objType;
    this._memberType = memberType;
    this._memberObjId = memberObjId;
    this._memberObjType = memberObjType;
}

/* equals/hashCode */
public boolean equals(Object other) {
    if (this == other) {
        return true;
    }
    if (other == null) {
        return false;
    }
    if (!(other instanceof CustomizedMembersPK)) {
        return false;
    }
    CustomizedMembersPK castOther = (CustomizedMembersPK) other;

    return new EqualsBuilder().append(this.getObjId(), castOther.getObjId())
        .append(this.getObjType(), castOther.getObjType())
        .append(this.getMemberType(), castOther.getMemberType())
        .append(this.getMemberObjId(), castOther.getMemberObjId())
        .append(this.getMemberObjType(), castOther.getMemberObjType())
        .isEquals();
}

public int hashCode() {
    return new HashCodeBuilder().append(this.getObjId())
        .append(this.getObjType())
        .append(this.getMemberType())
        .append(this.getMemberObjId())
        .append(this.getMemberObjType())
        .toHashCode();
}
```



```
}

/* Getter/Setter */
public Integer getMemberObjId() {
    return _memberObjId;
}
public void setMemberObjId(Integer memberObjId) {
    _memberObjId = memberObjId;
}
public String getMemberObjType() {
    return _memberObjType;
}
public void setMemberObjType(String memberObjType) {
    _memberObjType = memberObjType;
}
public String getMemberType() {
    return _memberType;
}
public void setMemberType(String memberType) {
    _memberType = memberType;
}
public Integer getObjId() {
    return _objId;
}
public void setObjId(Integer objId) {
    _objId = objId;
}
public String getObjType() {
    return _objType;
}
public void setObjType(String objType) {
    _objType = objType;
}
}

public CustomizedMembers() {
    ;
}
public CustomizedMembers(CustomizedMembersPK comp_id) {
    this._comp_id = comp_id;
}
public CustomizedMembers(Integer objId, String objType, String memberType, Integer
memberObjId, String memberObjType) {
    CustomizedMembersPK comp_id = new CustomizedMembersPK(objId, objType,
memberType, memberObjId, memberObjType);
```

```

        this.setComp_id(comp_id);
    }
    /* (non-Javadoc)
     * @see java.lang.Comparable#compareTo(java.lang.Object)
     */
    public int compareTo(Object o) {
        try {
            BusinessObject otherMem = ((CustomizedMembers)o).getDisplayObj();
            BusinessObject thisMem = this.getDisplayObj();
            if (thisMem instanceof Comparable && otherMem instanceof Comparable) {
                return ((Comparable)thisMem).compareTo(((Comparable)otherMem));
            }
            return thisMem.getName().compareTo(otherMem.getName());
        } catch (Exception e) {
            return 0;
        }
    }
    public BusinessObject getDisplayObj() throws GTSystemException {
        Integer thisId = this.getComp_id().getMemberObjId();
        String type = this.getComp_id().getMemberObjType();

        if (type.equals(Constants.OBJ_TYPE_COMPANY)) {
            return CompanyProcess.getInstance().getCompany(thisId);
        } else if (type.equals(Constants.OBJ_TYPE_USER)) {
            return UserProcess.getInstance().getUser(thisId);
        } else {
            return null;
        }
    }
    /* (non-Javadoc)
     * @see
     com.gearteks.framework.model.assist.DisplayObjInterface#setDisplayType(java.lang.String)
     */
    public void setDisplayType(String displayType) {
        ;
    }
    /**
     * @return the comp_id
     */
    public CustomizedMembersPK getComp_id() {
        return _comp_id;
    }
    /**
     * @param comp_id the comp_id to set
     */
    public void setComp_id(CustomizedMembersPK comp_id) {

```

```

    _comp_id = comp_id;
}
/* (non-Javadoc)
 * @see java.lang.Object#hashCode()
 */
@Override
public int hashCode() {
    return new HashCodeBuilder().append(this.getComp_id()).toHashCode();
}
/* (non-Javadoc)
 * @see java.lang.Object#equals(java.lang.Object)
 */
@Override
public boolean equals(Object obj) {
    if (this == obj) {
        return true;
    }
    if (obj == null) {
        return false;
    }
    if (!(obj instanceof CustomizedMembers)) {
        return false;
    }
    CustomizedMembers castOther = (CustomizedMembers) obj;

    return new EqualsBuilder().append(this.getComp_id(), castOther.getComp_id()).isEquals();
}
}
}

```

3. 创建 CustomizedMembers 类的 Hibernate 映射文件:

```

<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<!-- Generated 2007-9-12 9:48:14 by Hibernate Tools 3.2.0.b9 -->
<hibernate-mapping>
    <class name="com.gearteks.test.customized.model.CustomizedMembers"
table="customized_members">
        <composite-id name="comp_id"
class="com.gearteks.test.customized.model.CustomizedMembers$CustomizedMembersPK">
            <key-property name="objId" type="integer">
                <column name="objId" />

```

```
</key-property>
<key-property name="objType" type="string">
  <column name="ObjType" length="11" />
</key-property>
<key-property name="memberType" type="string">
  <column name="memberType" length="11" />
</key-property>
<key-property name="memberObjId" type="integer">
  <column name="memberObjId" />
</key-property>
<key-property name="memberObjType" type="string">
  <column name="memberObjType" length="11" />
</key-property>
</composite-id>
</class>

</hibernate-mapping>
```

4. 在\${8thManageHome}/WEB-INF/classes/hibernate/hibernate.cfg.xml 中增加类似以下配置：

```
<mapping resource="hibernate/CustomizedMembers.hbm.xml" />
```

5. 创建 CustomizedContractAction 类。该类实现 Callbackable 接口：

```
/**
 *
 */
package com.gearteks.test.customized;

import java.util.ArrayList;
import java.util.Collection;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.gearteks.framework.common.Constants;
import com.gearteks.framework.common.exception.GTSystemException;
```

```
import com.gearteks.framework.common.util.CommonUtil;
import com.gearteks.framework.common.util.SessionUtil;
import com.gearteks.framework.model.Company;
import com.gearteks.framework.model.Party;
import com.gearteks.framework.model.User;
import com.gearteks.framework.object.BusinessObject;
import com.gearteks.framework.process.ProcessErrors;
import com.gearteks.framework.process.assist.MatchObjectAssist;
import com.gearteks.imanage.controller.projmgmt.StrategyForm;
import com.gearteks.imanage.model.projmgmt.Strategy;
import com.gearteks.test.customized.model.CustomizedMembers;
import com.gearteks.test.customized.util.DbUtil;
import com.wisagetech.framework.callback.Callbackable;

/**
 * @author ericchen
 *
 */
public class CustomizedContractAction implements Callbackable {
    /**
     *
     * @param mapping
     * @param form
     * @param request
     * @param response
     * @return
     * @throws Exception
     */
    public ActionForward viewStrategy(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response)
        throws Exception {
        StrategyForm strategyForm = (StrategyForm)form;

        CustomizedMembers oldMember = this.getMember(strategyForm.getStrategyId(),
strategyForm.getObjType());
        if (oldMember != null) {
            BusinessObject bo = oldMember.getDisplayObj();
            if (bo != null) {
                request.setAttribute("suppliers", bo.getName());
            }
        }

        return null;
    }
}
/**
 *
 */
```

```

* @param mapping
* @param form
* @param request
* @param response
* @return
* @throws Exception
*/
public ActionForward saveStrategy(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response)
throws Exception {
    String suppliers = request.getParameter("suppliers");
    request.setAttribute("suppliers", suppliers);

    if (CommonUtil.hasActionErrors(request)) {
        return null;
    }

    Strategy strategy = this.getStrategy(form, request);
    if (strategy == null) {
        return null;
    }

    ProcessErrors errors = new ProcessErrors();
    if (suppliers != null && !suppliers.trim().equals("")) {
        BusinessObject supplier =
(BusinessObject)MatchObjectAssist.matchCompanyByType(suppliers,
Company.COMPANY_TYPE_SUPPLIER, errors, null,
SessionUtil.getMessage("quotation.label.supplier"),
SessionUtil.getMessage("quotation.label.supplier"));
        if (supplier != null) {
            CustomizedMembers oldMember = this.getMember(strategy.getId(),
strategy.getObjType());
            CustomizedMembers newMember = this.toMember("Supplier", supplier, strategy);
            this.saveCustomizedMembers(oldMember, newMember);
        }
    }

    if (errors.isHaveErrors()) {
        CommonUtil.saveErrors(request, errors);
    }

    return null;
}
/**
*
* @param form

```

```
* @param request
* @return
* @throws GTSystemException
*/
private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws
GTSystemException {
    StrategyForm strategyForm = (StrategyForm)form;
    Strategy strategy = (Strategy)request.getAttribute("strategy");
    if (strategy == null) {
        strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true);
    }
    return strategy;
}
/**
*
* @param memberType
* @param bo
* @return
*/
private CustomizedMembers toMember(String memberType, BusinessObject bo, Strategy
strategy) {
    String memberObjType = null;
    if (bo instanceof User) {
        memberObjType = Constants.OBJ_TYPE_USER;
    } else if (bo instanceof Party) {
        memberObjType = Constants.OBJ_TYPE_PARTY;
    } else if (bo instanceof Company) {
        memberObjType = Constants.OBJ_TYPE_COMPANY;
    }
    if (memberObjType != null) {
        return new CustomizedMembers(strategy.getId(), strategy.getObjType(), memberType,
bo.getId(), memberObjType);
    } else {
        return null;
    }
}
/**
*
* @param objId
* @param objType
* @return
* @throws GTSystemException
*/
private CustomizedMembers getMember(Integer objId, String objType) throws
GTSystemException {
    Collection<CustomizedMembers> col = this.getMembers(objId, objType);
```

```

        if (col != null && !col.isEmpty()) {
            return col.iterator().next();
        }
        return null;
    }
}
/**
 *
 * @param objId
 * @param objType
 * @return
 * @throws GTSystemException
 */
private Collection<CustomizedMembers> getMembers(Integer objId, String objType)
throws GTSystemException {
    return this.getMembers(objId, objType, null, null, null);
}
/**
 *
 * @param objId
 * @param objType
 * @param memberType
 * @param memberObjId
 * @param memberObjType
 * @return
 * @throws GTSystemException
 */
private Collection<CustomizedMembers> getMembers(Integer objId, String objType, String
memberType, Integer memberObjId, String memberObjType) throws GTSystemException {
    StringBuffer sb = new StringBuffer();
    sb.append("select m from CustomizedMembers m where 2 > 1 ");

    ArrayList<Object> paramList = new ArrayList<Object>();
    if (objId != null) {
        sb.append(" and m.comp_id.objId = ? ");
        paramList.add(objId);
    }
    if (objType != null && !"".equals(objType.trim())) {
        sb.append(" and m.comp_id.objType = ? ");
        paramList.add(objType);
    }
    if (memberType != null && !"".equals(memberType.trim())) {
        sb.append(" and m.comp_id.memberType = ? ");
        paramList.add(memberType);
    }
    if (memberObjId != null) {
        sb.append(" and m.comp_id.memberObjId = ? ");
    }
}

```



```
        paramList.add(memberObjId);
    }
    if (memberObjType != null && !"".equals(memberObjType.trim())) {
        sb.append(" and m.comp_id.memberObjType = ? ");
        paramList.add(memberObjType);
    }

    return DbUtil.getQueryData(sb.toString(), paramList);
}
/**
 *
 * @param oldMember
 * @param newMember
 * @throws GTSystemException
 */
private void saveCustomizedMembers(CustomizedMembers oldMember,
CustomizedMembers newMember) throws GTSystemException {
    ArrayList<CustomizedMembers> oldCol = new ArrayList<CustomizedMembers>();
    if (oldMember != null) {
        oldCol.add(oldMember);
    }
    ArrayList<CustomizedMembers> newCol = new ArrayList<CustomizedMembers>();
    if (newMember != null) {
        newCol.add(newMember);
    }
    this.saveCustomizedMembers(oldCol, newCol);
}
/**
 *
 * @param oldCol
 * @param newCol
 * @return
 * @throws GTSystemException
 */
private boolean saveCustomizedMembers(Collection<CustomizedMembers> oldCol,
Collection<CustomizedMembers> newCol) throws GTSystemException {
    if (newCol == null) {
        return false;
    }
    boolean changed = false;
    if (oldCol == null || oldCol.isEmpty()) {
        for (CustomizedMembers newMember : newCol) {
            DbUtil.add(newMember);
            changed = true;
        }
    } else {
```

```

    for (CustomizedMembers newMember : newCol) {
        if (!oldCol.contains(newMember)) {
            DbUtil.add(newMember);
            changed = true;
        }
    }
    for (CustomizedMembers oldMember : oldCol) {
        if (!newCol.contains(oldMember)) {
            DbUtil.delete(oldMember);
            changed = true;
        }
    }
}
return changed;
}
}

```

6. 在 strategy.jsp 上显示供应商字段:

```

<td width="130" class="fieldName" >
<%=SessionUtil.getMessage(request, "quotation.label.supplier")%>:
</td>
<td width="3">&nbsp;</td>
<td>
<% if (!readonly) { %>
<input type="text" name="suppliers" id="suppliers" class="text"
value=" <%=StringUtil.isEmpty(request.getAttribute("suppliers")) ? "" :
request.getAttribute("suppliers")%>" />
',true,'true',
null, null, 'companyType=Supplier&amp;hideCompanyType=true');return false;"
title=" <%=SessionUtil.getMessage(request, "prompt.title.companySelect.company")%>" />
<% } else { %>
<%=StringUtil.isEmpty(request.getAttribute("suppliers")) ? "" :
request.getAttribute("suppliers")%>
<input type="hidden" name="suppliers" id="suppliers"
value=" <%=StringUtil.isEmpty(request.getAttribute("suppliers")) ? "" :
request.getAttribute("suppliers")%>" />
<% } %>
</td>

```

注意: 系统会自动将 web 应用的根目录 (通过 `getContextPath()`) 加入到定制菜单/字段的 URL 中,

无须手动添加。

7. 在 `strategyJavaScript.jsp` 上增加以下 JavaScript 方法：

```
function setSupplier(objId, objName, otherParam, type, contact) {
    if (objId == "" || objName == "") {
        return;
    }
    var oSupplier = document.getElementById("suppliers");
    oSupplier.value = objName;
    return false;
}
```

8. 创建用于数据库操作的工具类：

```
/**
 *
 */
package com.gearteks.test.customized.util;

import java.io.Serializable;
import java.util.Collection;
import java.util.Iterator;

import org.hibernate.HibernateException;
import org.hibernate.Query;
import org.hibernate.Session;

import com.gearteks.framework.common.exception.GTObjectNotFoundException;
import com.gearteks.framework.common.exception.GTSystemException;
import com.gearteks.framework.common.util.SessionUtil;

/**
 * @author ericchen
 *
 */
public abstract class DbUtil {
    /**
     *
```

```
* @param currClass
* @param pk
* @param throwException
* @return
* @throws GTSYSTEMException
*/
public static Object find(Class currClass, Serializable pk, boolean throwException)
    throws GTSYSTEMException {
    try {
        Session session = SessionUtil.currentSession();
        Object ug = session.get(currClass, pk);

        if (ug == null) {
            if (throwException) {
                throw new GTOBJECTNotFoundException(currClass, pk);
            }
        }
        return ug;
    } catch (HibernateException e) {
        throw new GTSYSTEMException(e);
    }
}
/**
 *
 * @param obj
 * @return
 * @throws GTSYSTEMException
 */
public static boolean add(Object obj) throws GTSYSTEMException {
    try {
        Session session = SessionUtil.currentSession();
        session.save(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSYSTEMException(e);
    }
}
/**
 *
 * @return
 */
public static boolean update(Object obj) throws GTSYSTEMException {
    try {
        Session session = SessionUtil.currentSession();
        session.update(obj);
    }
}
```

```
        return true;
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
/**
 *
 * @param obj
 * @return
 * @throws GTSystemException
 */
public static boolean delete(Object obj) throws GTSystemException {
    try {
        Session session = SessionUtil.currentSession();
        session.delete(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
/**
 *
 * @param obj
 * @return
 * @throws GTSystemException
 */
public static boolean saveOrUpdate(Object obj) throws GTSystemException {
    try {
        Session session = SessionUtil.currentSession();
        session.saveOrUpdate(obj);
        return true;
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
/**
 *
 * @param sql
 * @param paramLst
 * @return
 * @throws GTSystemException
 */
public static Collection getQueryData(String hql, Collection paramLst) throws
GTSystemException {
    return getQueryData(hql, paramLst, false);
}
```

```

/**
 *
 * @param sql
 * @param list
 * @param useQueryCache
 * @return
 * @throws GTSystemException
 */
public static Collection getQueryData(String hql, Collection paramLst, boolean
useQueryCache) throws GTSystemException {
    if (hql == null || hql.trim().equals("") || paramLst == null) {
        throw new IllegalArgumentException("Null name or empty name is found, cannot
perform " + "getNamedQueryData(String s) function.");
    }

    try {
        Session session = SessionUtil.currentSession();
        Query query = session.createQuery(hql);
        if (useQueryCache) {
            query.setCacheable(true);
        }

        int i = 0;
        for (Iterator it = paramLst.iterator(); it.hasNext(); ) {
            query.setParameter(i++, it.next());
        }
        return query.list();
    } catch (HibernateException e) {
        throw new GTSystemException(e);
    }
}
}
}

```

9. 在`/${siteHome}/customize/callback/CustomizedCallback.xml` 中增加以下配置：

```

<action actionPoint="deal|viewStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedContractAction"
actionMethod="viewStrategy" />

<action actionPoint="deal|saveStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedContractAction"

```

```
actionMethod="saveStrategy" />
```

10. 重启 8thManage。

3.2.4.2.2 在营销活动基本信息页面增加显示其它字段

限于文档篇幅有限，请参考“在合约基本信息页面增加显示供应商字段”。可以通过以下方法获取

营销活动对象：

```
/**
 *
 * @param form
 * @param request
 * @return
 * @throws GTSystemException
 */
private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws
GTSystemException {
    StrategyForm strategyForm = (StrategyForm)form;
    Strategy strategy = (Strategy)request.getAttribute("strategy");
    if (strategy == null) {
        strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true);
    }
    return strategy;
}
```

在\${siteHome}/customize/callback/CustomizedCallback.xml 中增加类似以下配置：

```
<action actionPoint="campaign|viewStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedCampaignAction"
    actionMethod="viewStrategy" />

    <action actionPoint="campaign|saveStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedCampaignAction"
    actionMethod="saveStrategy" />
```

3.2.4.2.3 在机会/策略基本信息页面增加显示其它字段

限于文档篇幅有限，请参考“在合约基本信息页面增加显示供应商字段”。可以通过以下方法获取

机会/策略对象：

```
/**
 *
 * @param form
 * @param request
 * @return
 * @throws GTSYSTEMException
 */
private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws
GTSYSTEMException {
    StrategyForm strategyForm = (StrategyForm)form;
    Strategy strategy = (Strategy)request.getAttribute("strategy");
    if (strategy == null) {
        strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true);
    }
    return strategy;
}
```

在`/${siteHome}/customize/callback/CustomizedCallback.xml` 中增加类似以下配置：

```
<action actionPoint="strategy|viewStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
    actionMethod="viewStrategy" />

    <action actionPoint="strategy|saveStrategy" location="post"
        sameTransaction="true" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
        actionMethod="saveStrategy" />
```

3.2.4.3 同步业务对象信息到外部数据库

3.2.4.3.1 同步用户信息到外部数据库

在使用 8Manage 创建用户时，需要把用户信息同步到外部数据库，例如：carson 数据库的 users

表中。回调过程跟加入点 `actionPoint` 不需要在同一个数据库事务中，以保证同步过程中的错误不影响到 8Manage 的使用。步骤如下：

在数据库服务器（例如：carson）上创建新的用户表：

```
CREATE TABLE users (  
    Address1          nvarchar(256) NULL,  
    CreatorId        int NOT NULL,  
    CreatorTS        datetime CONSTRAINT [DF1_users] DEFAULT (getdate()) NOT NULL,  
    Description       nvarchar(2000) NULL,  
    Email            nvarchar(64) NULL,  
    Fax              nvarchar(64) NULL,  
    FirstName        nvarchar(64) NOT NULL,  
    LastName         nvarchar(64) NOT NULL,  
    MiddleName       nvarchar(64) NULL,  
    Password         nvarchar(64) NULL,  
    Phone            nvarchar(64) NULL,  
    Status           varchar(11) NOT NULL,  
    TS               datetime CONSTRAINT [DF2_users] DEFAULT (getdate()) NOT NULL,  
    TSUserId         int NOT NULL,  
    UserId           int NOT NULL,  
    UserName         nvarchar(64) NOT NULL,  
    Zip              nvarchar(64) NULL,  
    Sex              varchar(11) NULL,  
    CONSTRAINT [CK1_users] CHECK (([Status] = 'Suspend' or ([Status] = 'Deleted' or [Status] =  
'Active'))  
);  
go  
  
ALTER TABLE users ADD  
    CONSTRAINT PK_users PRIMARY KEY CLUSTERED (UserId);  
go  
  
CREATE NONCLUSTERED INDEX AK1_users ON users (UserName);  
go
```

创建 `CustomizedUserAction` 类。该类实现 `Callbackable` 接口：

```
package com.gearteks.test.customized;  
  
import java.sql.Connection;  
import java.sql.DriverManager;
```

```
import java.sql.SQLException;
import java.sql.Statement;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;

import com.gearteks.framework.common.util.CommonUtil;
import com.gearteks.framework.controller.UserForm;
import com.wisagetech.framework.callback.Callbackable;

/**
 * @author ericchen
 *
 */
public class CustomizedUserAction implements Callbackable {
    /**
     *
     * @param mapping
     * @param form
     * @param request
     * @param response
     * @return
     * @throws Exception
     */
    public ActionForward doSync(ActionMapping mapping, ActionForm form,
        HttpServletRequest request, HttpServletResponse response)
        throws Exception {
        UserForm uf = (UserForm)form;

        if (CommonUtil.hasActionErrors(request)) {
            return null;
        }

        Connection conn = null;
        Statement stmt = null;
        try {
            StringBuffer sb = new StringBuffer();
            sb.append(" insert into users (Address1, CreatorId, CreatorTS, Description, Email, Fax,
                FirstName, LastName, MiddleName, Password, Phone, Status, TS, TSUserId, UserId, UserName,
                Zip, Sex) ")
                .append(" values (")
                .append(")").append(uf.getAddress1()).append(", ")

```

```

        .append("").append(uf.getCreator().getUserId()).append(", ")
        .append("").append(uf.getCreateTime()).append(", ")
        .append("").append(uf.getDescription()).append(", ")
        .append("").append(uf.getEmail()).append(", ")
        .append("").append(uf.getFax()).append(", ")
        .append("").append(uf.getFirstName()).append(", ")
        .append("").append(uf.getLastName()).append(", ")
        .append("").append(uf.getMiddleName()).append(", ")
        .append("").append(uf.getPassword()).append(", ")
        .append("").append(uf.getPhone()).append(", ")
        .append("").append(uf.getStatus()).append(", ")
        .append("").append(uf.getTs()).append(", ")
        .append("").append(uf.getTsUser().getUserId()).append(", ")
        .append("").append(uf.getUserId()).append(", ")
        .append("").append(uf.getUserName()).append(", ")
        .append("").append(uf.getPostCode()).append(", ")
        .append("").append(uf.getSex()).append("")
        .append(" ");
    conn = this.getConnection();
    stmt = conn.createStatement();
    stmt.execute(sb.toString());
} finally {
    this.closeQuietly(stmt);
    this.closeQuietly(conn);
}
return null;
}
/**
 *
 * @return
 */
private Connection getConnection() {
    String driverClass = "net.sourceforge.jtds.jdbc.Driver";
    String connUrl =
"jdbc:jtds:sqlserver://carson:1433/ericchen141;lastUpdateCount=true;instance=";
    String dbUser = "dbuser";
    String dbPwd = "db1234";

    try {
        Class.forName(driverClass);
        return DriverManager.getConnection(connUrl, dbUser, dbPwd);
    } catch (Exception ex) {
        ;
    }
    return null;
}

```

```
/**
 *
 * @param stmt
 */
private void closeQuietly(Statement stmt) {
    if (stmt != null) {
        try {
            stmt.close();
        } catch (SQLException e) {
            ;
        }
    }
}

/**
 *
 * @param conn
 */
private void closeQuietly(Connection conn) {
    if (conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            ;
        }
    }
}
}
```

在`/${siteHome}/customize/callback/CustomizedCallback.xml` 中增加以下配置:

```
<action actionPoint="user|doAdd" location="post"
        sameTransaction="false" useForward="false"
        actionClass="com.gearteks.test.customized.CustomizedUserAction"
        actionMethod="doSync" />
```

重启 8Manage。

3.2.5 8Manage 常用 actionPoint

ActionName	MethodName	插入点描述	是否有其它参数作为流程流转	代码示例
campaign	listStrategy	打开营销活动列表页面	N	1、在个性化回调方法中获取营销活动列表信息： Collection campaignLst = (Collection)request.getAttribute(Constants.LIST);
campaign	viewStrategy	打开营销活动页面	N.	1、在个性化回调方法中获取营销活动对象： Strategy campaign = (Strategy)request.getAttribute("strategy");
campaign	saveStrategy	保存营销活动	N. 参数有 "isUpdatedSpecial" 会只保存部分内容；参数有 "isAddJobStrategy" 会增加 JobStrategy 关联记录；参数有 "msgId" 会增加 StrategyObjMember 关联记录。	1、在个性化回调方法中获取营销活动对象： private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GTSysException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strategy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; }
strategy	listStrategy	打开机会/策略列表页面	N	1、在个性化回调方法中获取机会/策略列表信息： Collection strategyLst = (Collection)request.getAttribute(Constants.LIST);
strategy	viewStrategy	打开机会/策略页面	N.	1、在个性化回调方法中获取机会/策略对象： Strategy strategy = (Strategy)request.getAttribute("strategy");
strategy	saveStrategy	保存机会/策略	N. 参数有 "isUpdatedSpecial" 会只保存部分内容；参数有 "isAddJobStrategy"	1、在个性化回调方法中获取机会/策略对象： private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws

ActionName	MethodName	插入点描述	是否有其它参数作为流程流转	代码示例
			" 会增加 JobStrategy 关联记录; 参数有 "msgId" 会增加 StrategyObjMember 关联记录。	<pre>GSystemException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strategy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; }</pre>
deal	listStrategy	打开合约列表页面	N	1、在个性化回调方法中获取合约列表信息: <pre>Collection contractLst = (Collection)request.getAttribute(Constants.LIST);</pre>
deal	viewStrategy	打开合约页面	N.	1、在个性化回调方法中获取合约对象: <pre>Strategy contract = (Strategy)request.getAttribute("strategy");</pre>
deal	saveStrategy	保存合约	N. 参数有 "isUpdatedSpecial" 会只保存部分内容; 参数有 "isAddJobStrategy" 会增加 JobStrategy 关联记录; 参数有 "msgId" 会增加 StrategyObjMember 关联记录。	1、在个性化回调方法中获取合约对象: <pre>private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GSystemException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strategy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; }</pre>
quotation	doShowList	打开订单列表页面	N	1、在个性化回调方法中获取订单列表信息: <pre>Collection<Quotation> quotationLst = (Collection<Quotation>)request.getAttribute(Constants.LIST);</pre>
quotation	doShowInfo	打开订单页面	N	1、在个性化回调方法中获取订单对象: <pre>Quotation quotation =</pre>

ActionName	MethodName	插入点描述	是否有其它参数作为流程流转	代码示例
				(Quotation)request.getAttribute("quotation");
quotation	doSave	保存订单	N	<p>1、在个性化回调方法中获取订单对象：</p> <pre>private Quotation getQuotation(ActionForm form, HttpServletRequest request) throws GTSYSTEMException { QuotationForm quotationForm = (QuotationForm)form; Quotation quotation = (Quotation)request.getAttribute(" quotation"); if (quotation == null) { quotation = (Quotation)DbUtil.find(Quotation. class, quotationForm.getQuotationId(), true); } return quotation; }</pre>
ActionName	MethodName	动作	是否有其它参数作为流程流转	描述
campaign	listStrategy	打开营销活动列表页面	N	<p>1、在个性化回调方法中获取营销活动列表信息：</p> <pre>Collection campaignLst = (Collection)request.getAttribute(Constants.LIST);</pre>
campaign	viewStrategy	打开营销活动页面	N.	<p>1、在个性化回调方法中获取营销活动对象：</p> <pre>Strategy campaign = (Strategy)request.getAttribute("strateg y");</pre>
campaign	saveStrategy	保存营销活动	N. 参数有 "isUpdatedSpecial" 会只保存部分内容；参数有 "isAddJobStrategy" 会增加 JobStrategy 关联记录；参数有 "msgId" 会增加 StrategyObjMember 关联记录。	<p>1、在个性化回调方法中获取营销活动对象：</p> <pre>private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GTSYSTEMException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strateg y"); if (strategy == null) {</pre>

ActionName	MethodName	插入点描述	是否有其它参数作为流程流转	代码示例
				<pre> strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; } </pre>
strategy	listStrategy	打开机会/策略列表页面	N	<p>1、在个性化回调方法中获取机会/策略列表信息:</p> <pre> Collection strategyLst = (Collection)request.getAttribute(Constants.LIST); </pre>
strategy	viewStrategy	打开机会/策略页面	N.	<p>1、在个性化回调方法中获取机会/策略对象:</p> <pre> Strategy strategy = (Strategy)request.getAttribute("strategy"); </pre>
strategy	saveStrategy	保存机会/策略	N. 参数有 "isUpdatedSpecial" 会只保存部分内容; 参数有 "isAddJobStrategy" 会增加 JobStrategy 关联记录; 参数有 "msgId" 会增加 StrategyObjMember 关联记录。	<p>1、在个性化回调方法中获取机会/策略对象:</p> <pre> private Strategy getStrategy(ActionForm form, HttpServletRequest request) throws GTSystemException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strategy"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; } </pre>
deal	listStrategy	打开合约列表页面	N	<p>1、在个性化回调方法中获取合约列表信息:</p> <pre> Collection contractLst = (Collection)request.getAttribute(Constants.LIST); </pre>
deal	viewStrategy	打开合约页面	N.	<p>1、在个性化回调方法中获取合约对象:</p> <pre> Strategy contract = (Strategy)request.getAttribute("strategy"); </pre>
deal	saveStrategy	保存合约	N. 参数有 "isUpdatedSpecial	<p>1、在个性化回调方法中获取合约对象:</p> <pre> private Strategy </pre>

ActionName	MethodName	插入点描述	是否有其它参数作为流程流转	代码示例
			“ 会只保存部分内容; 参数有 “isAddJobStrategy” 会增加 JobStrategy 关联记录; 参数有 “msgId” 会增加 StrategyObjMember 关联记录。	<pre> getStrategy(ActionForm form, HttpServletRequest request) throws GTSysException { StrategyForm strategyForm = (StrategyForm)form; Strategy strategy = (Strategy)request.getAttribute("strateg y"); if (strategy == null) { strategy = (Strategy)DbUtil.find(Strategy.class, strategyForm.getStrategyId(), true); } return strategy; } </pre>
quotation	doShowList	打开订单列表页面	N	1、在个性化回调方法中获取订单列表信息: <pre> Collection<Quotation> quotationLst = (Collection<Quotation>)request.getAtt ribute(Constants.LIST); </pre>
quotation	doShowInfo	打开订单页面	N	1、在个性化回调方法中获取订单对象: <pre> Quotation quotation = (Quotation)request.getAttribute("quot ation"); </pre>
quotation	doSave	保存订单	N	1、在个性化回调方法中获取订单对象: <pre> private Quotation getQuotation(ActionForm form, HttpServletRequest request) throws GTSysException { QuotationForm quotationForm = (QuotationForm)form; Quotation quotation = (Quotation)request.getAttribute(" quotation"); if (quotation == null) { quotation = (Quotation)DbUtil.find(Quotation. class, quotationForm.getQuotationId(), true); } return quotation; } </pre>

限于篇幅有限，这里不能一一列举 8Manage 中其他 actionPoint 信息，如果有需要的话，请联系 8Manage 的客服人员。

注意：

当同一个 actionPoint 有多个 Custom Actions 的时候，这些 Custom Action 都会起作用。当一个 actionPoint 按业务逻辑的要求做多个操作时，actionPoint 的名称是以 action|method 的形式命名的，以便相互区别。

3.3 用 8Manage 界面组件定制用户界面

3.3.1 概述

8Manage 支持用户通过创建新的页面或修订现有的页面来扩展 8Manage 的功能。8Manage 提供所有用于实现用户界面的源代码，方便用户修改和定制任何页面。用户可在现有页面中添加新的字段，或甚至添加一个全新的页面。8Manage 同时提供用户界面组件（例如，日历、用户选项框、自动匹配等），使得用户自定义的功能与默认功能在外观风格上一模一样。

在为一个自定义的功能开发用户界面时，你总是可选择通过重用现有用户界面组件来最小化开发工作量。

3.3.2 实现步骤

请参考 3.2.4.2 的示例。

3.4 添加自定义菜单到系统菜单

3.4.1 概述

8Manage 的菜单链接是各主要功能模块的进入点（起始点）。

8Manage 提供的 SDK 接口可让用户在菜单中添加更多现有的功能链接，同时也可以添加用户自定义功能的链接。通过修改菜单的链接，用户也可在菜单中添加链接到其它应用系统的链接，或者进入其它 8Manage 功能的快捷方式链接。

本节内容介绍如何在 8Manage 中实现在系统菜单加入自定义菜单。此功能实现客户往 8Manage 系统菜单加入客户自定义菜单，以进入客户自定义的界面等二次开发功能。

3.4.2 实现方法

1) 客户自定义菜单都加入到 `customizedMenus-8thManage.xml` 配置文件。

你可以在 `{siteHome}/customize/menu` 下找到这个文件。其中 `{siteHome}` 表示 8Manage 的主目录。`customizedMenus-8thManage.xml` 配置文件的格式如下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<application id="8thManage" >
  <menu-group id="myAttentionCustomizedMenu" >
    <menu-item id="plusMenu1" key="WisageTech" location="http://www.8manage.com"
  />
    <menu-item id="plusMenu2" key="Google" location="http://www.google.com" />
  </menu-group>
</application>
```

- **application**

application 节点为 customizedMenus-8thManage 的根节点。

该节点不可缺少。

- **menu-group**

menu-group 节点为 application 节点的子节点，表示一组自定义菜单的配置。

customizedMenus-8thManage.xml 文件中默认已经有了若干个 menu-group 节点，二次开发仅支持在在

menu-group 中加 menu-item 节点，不支持增加或修改 menu-group 节点。menu-group 节点只有 id

这一个属性节点，该属性用来唯一标识此组自定义菜单的 id。该 ID 为内置的，不允许修改，系统

会根据定义好的 id，决定显示在左边菜单的哪个位置，或者显示为 More 菜单(详见“定制‘更多’

菜单下面的菜单项”)。

该节点不可缺少。

- **menu-item**

menu-item 节点是 menu-group 节点的子节点，表示单个自定义菜单的配置。menu-item 节点

有 id、key 和 location 共三个属性节点。下面是每个属性节点的具体描述。

该节点不可缺少。

- **id**

id 属性表示菜单的唯一性 id。此 id 要保证在系统里所有菜单的 id 标识不能重复。所以命名此 id 时请尽量使用一种规则来确保此 id 不会重复。原则上 8Manage 要求以 plusMenu 开头来命名。

该属性节点不可缺少。

- **key**

key 属性表示菜单的名称。此名称会出现在 8Manage 系统的左菜单。此名称也可以使用资源文件的 key 值，如果使用的是资源文件的 key 值，则系统会取此 key 值对应的 value 作为此菜单的名称显示在系统左菜单。

该属性节点不可缺少。

- **location**

location 属性表示菜单的 URL。当用户点击此菜单时，系统的主显示区就会链接进入此 URL 所指向的地址，显示此自定义菜单的界面。注意 **location** 中的值如果含有 “&” 符号，应该以 “&” 代替。当指向的域名跟 8Manage 相同的时候，可以省略域名部分，假设 8Manage 服务器为本机，当 **location** 为

“<http://localhost:8080/8thManage/projectManagement/job.do?method=doShowList>”

时，其中的 <http://localhost:8080> 可以省略，直接为

“/8thManage/projectManagement/job.do?method=doShowList” 即可。

该属性节点不可缺少。

2) 自定义菜单文件 menu-group 的 id 所对应的显示位置

以下为文件中各 menu-group 的 id 所对应的 8Manage 系统里左侧菜单的位置。在以下 ID 的 menu-group 节点下添加的 menu-item 节点，将显示在左侧菜单的相关位置。

id	模块	位置
myAttentionCustomizedMenu	我需留意	我需留意菜单的结尾
enterpriseAdminCustomizedMenu	企业	导入和替换菜单的结尾
enterpriseInfoCustomizedMenu	企业	信息菜单的结尾
enterpriseTrackingCustomizedMenu	企业	跟踪分布菜单的结尾
clientBusinessCustomizedMenu	工作	客户业务菜单的结尾
supplierBusinessCustomizedMenu	工作	供应商业务菜单的结尾
internalBusinessCustomizedMenu	工作	内部工作菜单的结尾
myProjectsCustomizedMenu	工作	项目与支持菜单的结尾
internalOperationCustomizedMenu	系统管理	内部操作菜单的结尾
projectOpportunityCustomizedMenu	系统管理	工作菜单的结尾
clientAdminCustomizedMenu	系统管理	客户和供应商菜单的结尾
changeOrderAdminCustomizedMenu	系统管理	变更请求菜单的结尾
productSystemCustomizedMenu	系统管理	产品与服务菜单的结尾
purgeSettingsCustomizedMenu	系统管理	清理数据菜单的结尾
systemAdminCustomizedMenu	系统管理	系统菜单的结尾

3.5 客制化字段名

3.5.1 概述

在 8Manage 中，每个字段的字段名都是反复考虑而定下来的。但不同的客户存在不同的语言习惯或特殊要求，需要把 8Manage 中已经存在的字段名另外起另一个名字。8Manage 支持这种需求。

3.5.2 实现方法

- 1) 找到要更改名称的字段所在的页面的 JSP 文件
- 2) 找到该字段所用的多语言信息的 key 值
- 3) 在 `{siteHome}\customize\resources` 目录下的所有资源文件中增加 Key-value 值，Key 值跟上
面找到的 Key 值一致
- 4) 如果是非英文语言，请用 JDK 中的 `native2ascii` 命令进行编码转换
- 5) 重启 8Manage

3.5.3 实现示例

目标：合同中的“编号”字段（如下图），中文版改为“合同编号”，英文版改为“ID”：

提议的客户合同

* 名称: 测试合同	状态: 未提交
编号: 1001	供应商: XXX供应商
* 负责人: admin	* 负责内部机构: Engineering
* 销售代表: admin	* 法人实体: Corporate
* 默认审批人: 用户自定义	* 审批人: admin
协助内部机构:	协助人员:

- 1) 找到合约的页面，也就是“ID 编号”所在的页面，如上图。
- 2) 在浏览器中查看源代码，会在源代码比较靠前的地方发现下面的代码：

```
<base href="http://localhost:8080/8thManage/projectManagement/strategy.jsp">
```

从而找到相应的 jsp 文件 `strategy.jsp`

- 3) 在 `strategy.jsp` 中找到“ID 编号”的 Key 值 `strategy.label.strategyNo`

```

<table border="0" cellspacing="0" cellpadding="0"
<%=setting != null && setting.isOnlyShowInShowAll() ? "id=\"dynic\" name=\"dynic\"
alt=\"strategyNo\" style=\"display:none\" : ""%>>
<tr><td height="3" colspan="8"></td></tr>
<tr>
<td width="<%=String.valueOf(HtmlTagHelper.getWidth(request, width_en, width_zh))%">
class="fieldName">
<% if (setting != null && !setting.isAutoGen() && !setting.isOptional()) { %><span
class="star">*</span><% } %>
    <%=SessionUtil.getMessage(request, "strategy.label.strategyNo")%></td>
<td width="3">&nbsp;</td>
    <td>
        <% if (setting != null && !setting.isAutoGen()) { %>
            <gearTag:text property="strategyNo" size="90" maxlength="64"
styleClass="text" alt="<%=setting != null && setting.isOptional() ? "" : "notnull"%> "
title="<%=SessionUtil.getMessage(request, "strategy.label.strategyNo")%> "
gearUsePolicy="<%=!isIDCanBeUpdate %>" gearHaveHidden="true"/>
        <% } else { %>
            <html:hidden property="strategyNo"/>
            <%=strategyForm.getStrategyNo() != null ?
strategyForm.getStrategyNo() : HtmlTagHelper.EMPTY_DISPLAY_VALUE %>
            <% } %>
        </td>
    <td colspan="5">&nbsp;</td>
</tr>
</table>

```

4) 在`#{siteHome}customize\resources\ CustomizedResources_en.properties` 文件中添加以

下 Key-value:

```
strategy.label.strategyNo=ID
```

在`#{siteHome}customize\resources\ CustomizedResources_zh_CN.properties` 文件中添加

以下 Key-value:

```
strategy.label.strategyNo=合同编号
```

5) 由于`#{siteHome}customize\resources\ CustomizedResources_zh_CN.properties` 是中文,

需要转换编码。打开命令行窗口, 把以上路径设为当前路径, 再执行以下命令:

```
native2ascii -encoding utf8 CustomizedResources_zh_CN.properties
CustomizedResources_zh_CN2.properties
```


执行以上命令之前需保证 JDK 在你的环境路径里面。

将生成的 CustomizedResources_zh_CN2.properties 文件覆盖

CustomizedResources_zh_CN.properties 文件。

6) 重启 Tomcat

3.5.4 注意事项

并不是每个字段名都这么容易的改到，存在以下情况之下，请慎重改字段名：

1) 整个系统中统一命名的字段，如“负责人”、“描述”、“状态”等，对应的 Key 为

“display.fieldName.responsible”、“display.fieldName.description”、

“display.fieldName.status”，除非您想整个系统统一改成另外一个名字，否则不要改。

2) 某几个模块可能公用一个 key，如上述的例子，“ID 编号”是

campaign/Opportunity/Contract 公用的一个 key，如果想改成“编号”是可以的，如果想

改成是“合约编号”那就不可以，因为 Campaign 和 Opportunity 里面也会出现一个字段叫

“合约编号”，那就很奇怪了。

3.6 自定义代码表

3.6.1 概述

为了保存不同的 8Manage 基础数据，用户可能需要自定义代码表。此节介绍如何实现此定制。

3.6.2 实现步骤

1) 要自定义代码表，用户首先必须定义一个自定义的所有者类的字符串。例如，选用

“CustomCodeType” 作为所有者类型字符串。

2) 将三个自定义属性加入到文件 `customizedResources_xxx.properties`，例如：

```
admin.label.customized.CustomProductType=Customized product type
admin.label.customized.CustomProductType.l=customized product type
admin.label.customized.CustomProductType.l.s= Customized product types
```

属性键值应该遵循以下规则：以 “admin.label.customized” 开头，紧接着是自定义所有者类型。最好，第二和第三个属性以此自定义属性结尾。

3) 为了加入对功能权限跟踪的支持，用户应插入 6 条记录到表 `func_url_param` 当中，例如：

```
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doAdd',      'ownerType=CustomProductType', 9);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doDelete',   'ownerType=CustomProductType', 9);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doShowAdd',  'ownerType=CustomProductType', 9);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doShowInfo', 'ownerType=CustomProductType', 8);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doShowList', 'ownerType=CustomProductType', 8);
insert into func_url_param(funcId, actionName, methodName, params, dependFuncId)
  values (-1, 'adminType', 'doUpdate',   'ownerType=CustomProductType', 9);
```

请注意，“params” 字段的值的格式应该是：“ownerType=YOUR_OWNER_TYPE”。同时，

“dependFuncId” 字段值应该是你的自定义代码表所属的函数 ID。用户可在数据库中

“func” 表中找到函数 ID 编号，或者直接联系高亚科技技术支持或客服寻求帮助。

4) 最好，创建一个链接，例如：

`"/common/adminType.do?method=doShowList&ownerType=CustomProductType"`

。一定要记得将 “ownerType” 替换为您自定义的所有者类型字符串。

至此，自定义代码表的定制即算完成。

3.7 数据权限检查机制

3.7.1 概述

当用户需要读取/编辑单个记录时，系统可能需要进行数据权限的检查。在 8Manage 中，开发人员可以调用数据权限检查并确认当前用户是否有权读取/编辑当前的数据对象。

3.7.2 调用数据权限检查

1) 检查当前用户是否有读取一个记录的权限：

```
com.gearteks.imate.common.DataRightManager.getInstance().isReadAllowed(id, type, context);
```

以下是上述参数的介绍：

- I. id，数据对象的 ID。
- II. type，数据对象类型，请参考类型列表了解更详细的介绍。
- III. context，用户可通过 **SessionUtil.getCurrentAclContext(HttpServletRequest request)** 获取当前上下文。

2) 检查当前用户是否有修改一个记录的权限：

```
com.gearteks.imate.common.DataRightManager.getInstance().isEditAllowed(id, type, context);
```

以下是上述参数的介绍：

- I. id, 数据对象的 ID。
- II. type, 数据对象的类型。请参考类型列表了解更详细的介绍。
- III. context, 用户可通过 **SessionUtil.getCurrentAclContext(HttpServletRequest request)** 获取当前的上下文。

3) 获当前用户用于特定数据对象的过滤条件：

**com.gearteks.imate.common.DataRightManager.getInstance().getFilterCriteria
(type, context, alias, isHql);**

以下是上述参数的介绍：

- I. type, 数据对象类型。请参考类型列表了解更详细的介绍。
- II. context, 用户可通过 **SessionUtil.getCurrentAclContext(HttpServletRequest request)** 获取当前的上下文。
- III. alias, 在 SQL 中的别名, 代表数据对象表。
- IV. isHql, 表示是否返回 Hql 格式的条件, isHql 默认返回的 SQL 格式条件为 false。

3.7.3 类型列表

以下是常用到的一些类型的列表：

类别	类型名称
Client/Agent/Supplier/Partner/Competitor/Other organizations	Company

类别	类型名称
Group	Group
Internal User	InternalUser
External User	ExternalUser
Project	Project
Activity	Activity
Contract	Deal
Opportunity	Strategy
Campaign	Campaign

3.8 搭建 SDK 远程 debug 环境

3.8.1 概述

本文档的目的是向开发人员介绍 8Manage SDK（软件开发工具）远程 Debug 环境的搭建。

3.8.2 以远程 Debug 方式启动 8Manage

- A. 如果你的 8Manage 是安装在 C:\WisageTech 下，请将 startup-remoteDebug.bat 文件放置于 C:\WisageTech_installer\8thManage\tomcat\bin 目录下。



startup-remoteDebug.bat

B. 如果安装 8Manage 的机器端口 8000 是已经被占用的, 请打开 `startup-remoteDebug.bat` 文件,

将里面的这一行中的 8000 转换成你希望远程 Debug 的端口。

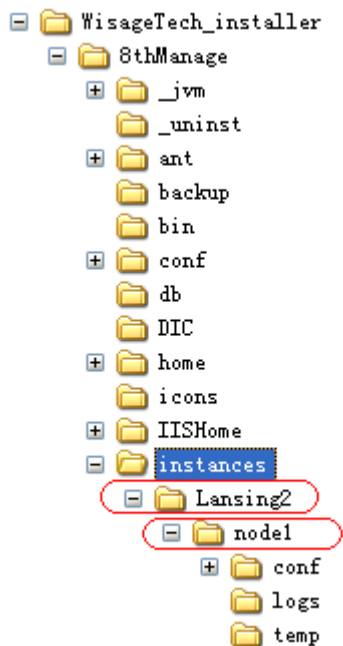
```
set JAVA_OPTS=%JAVA_OPTS% -Xdebug  
-Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=y
```

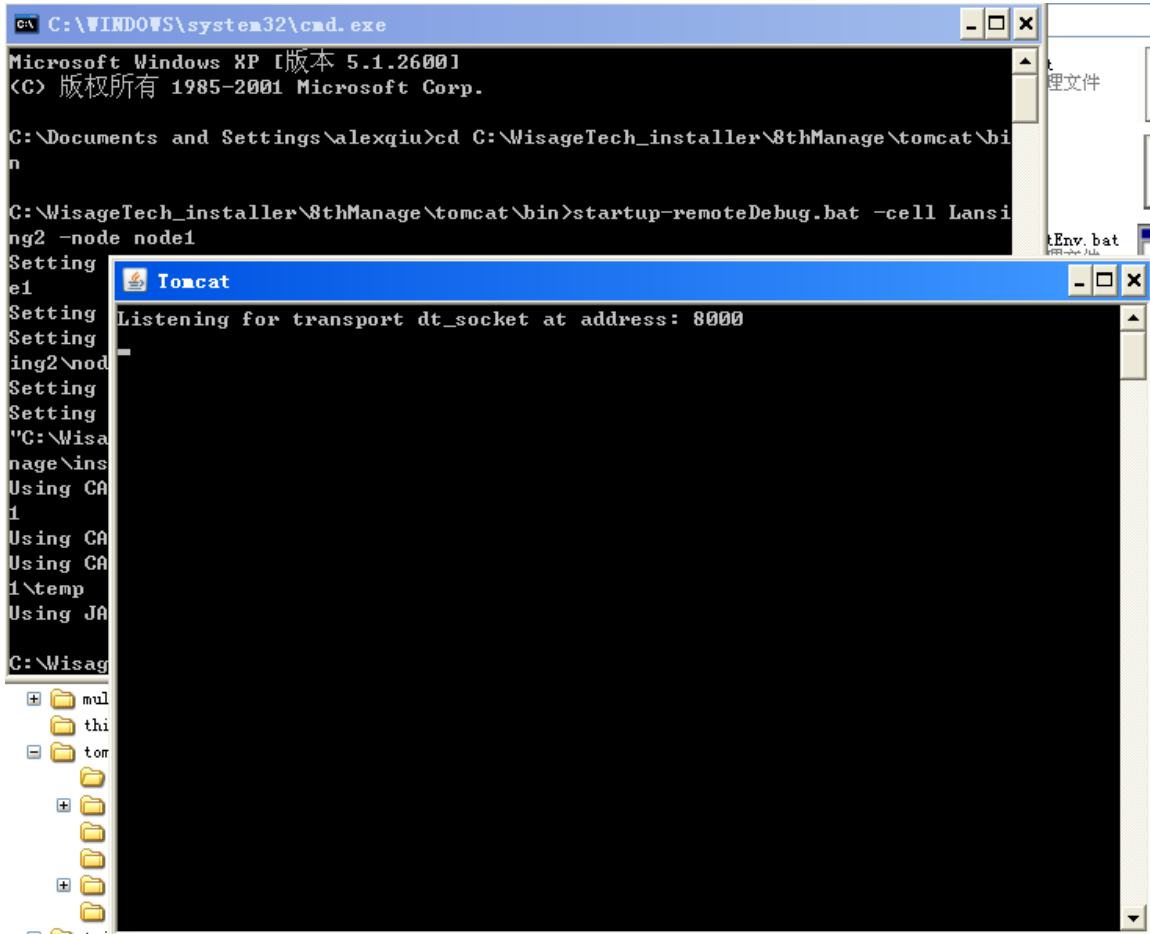
C. 用 `cmd` 打开一个命令行窗口, 来到 `C:\WisageTech_installer\8thManage\tomcat\bin` 下面, 输入

```
startup-remoteDebug.bat -cell $CELLNAME$ -node $NODENAME$
```

以下图为例, 其中 `$CELLNAME$` 为 `Lansing2`, `$NODENAME$` 为 `node1`, 则输入命令为:

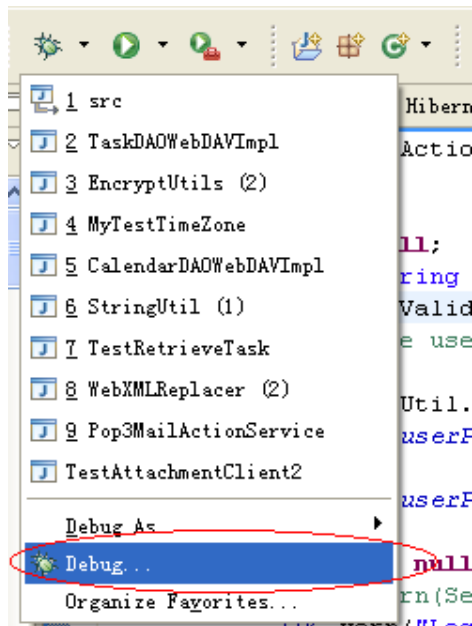
```
startup-remoteDebug.bat -cell Lansing2 -node node1
```



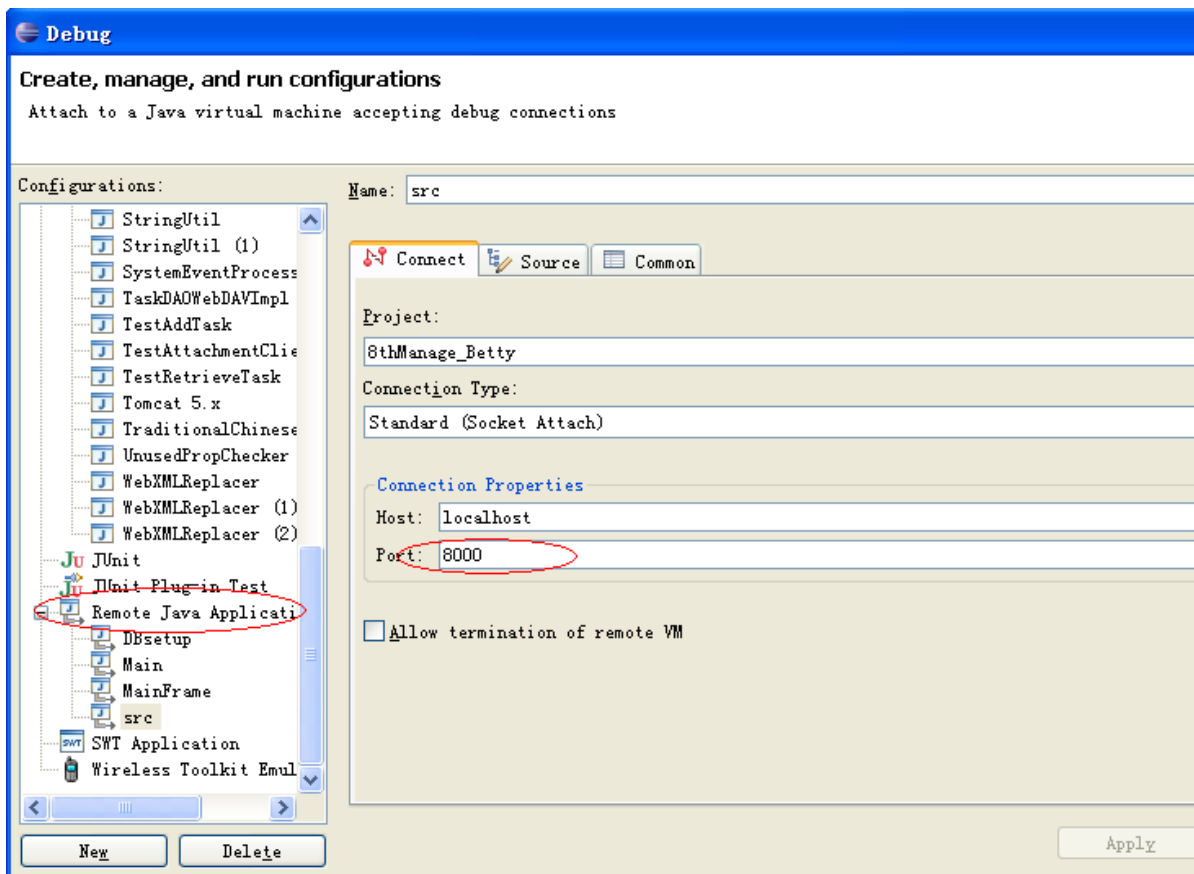


配置 Eclipse 接收远端 Debug 信息

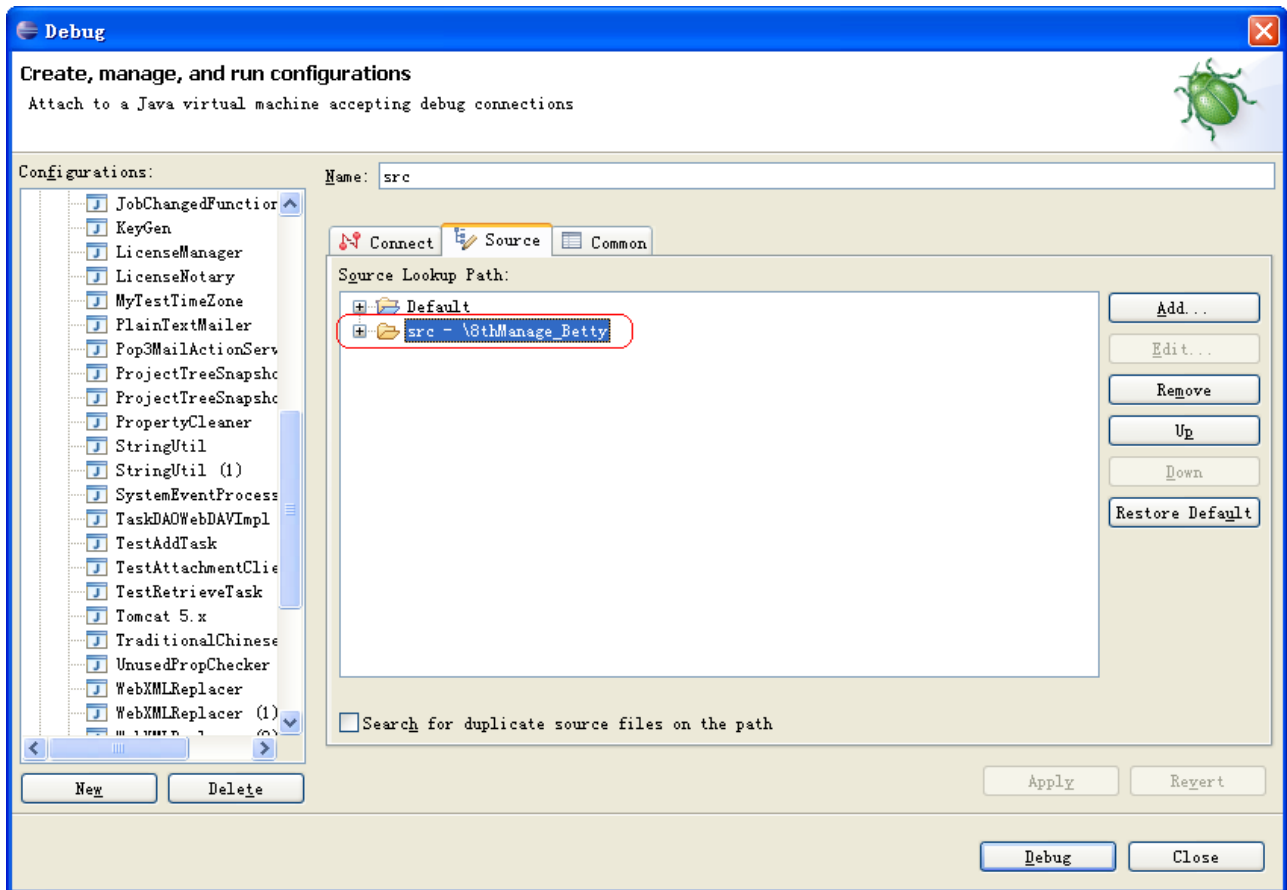
D. 在 Eclipse 中新开一个项目，这里不介意是任何类型的 Java 相关的项目。



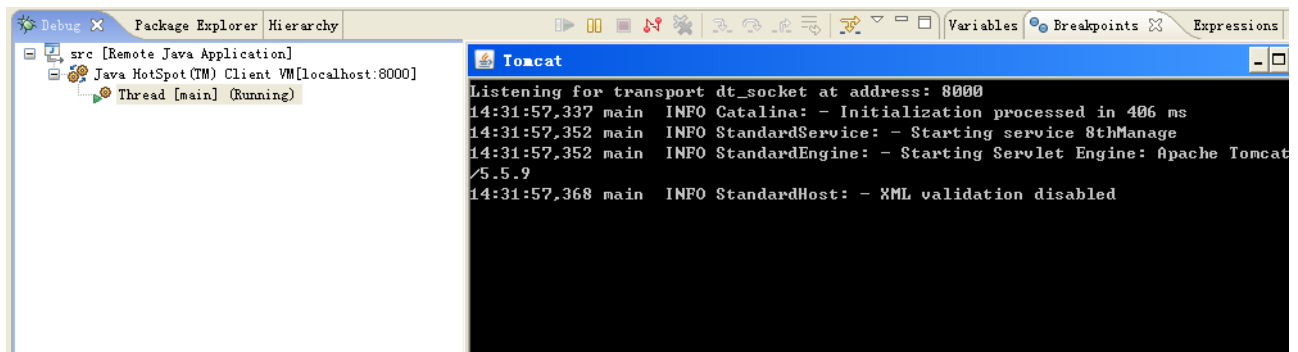
在 Remote Java Application 中，新建一个 Debug 配置。Host 与 Port 配置好，其中 Port 应该是 startup-remoteDebug.bat 中的远程 Debug 端口。

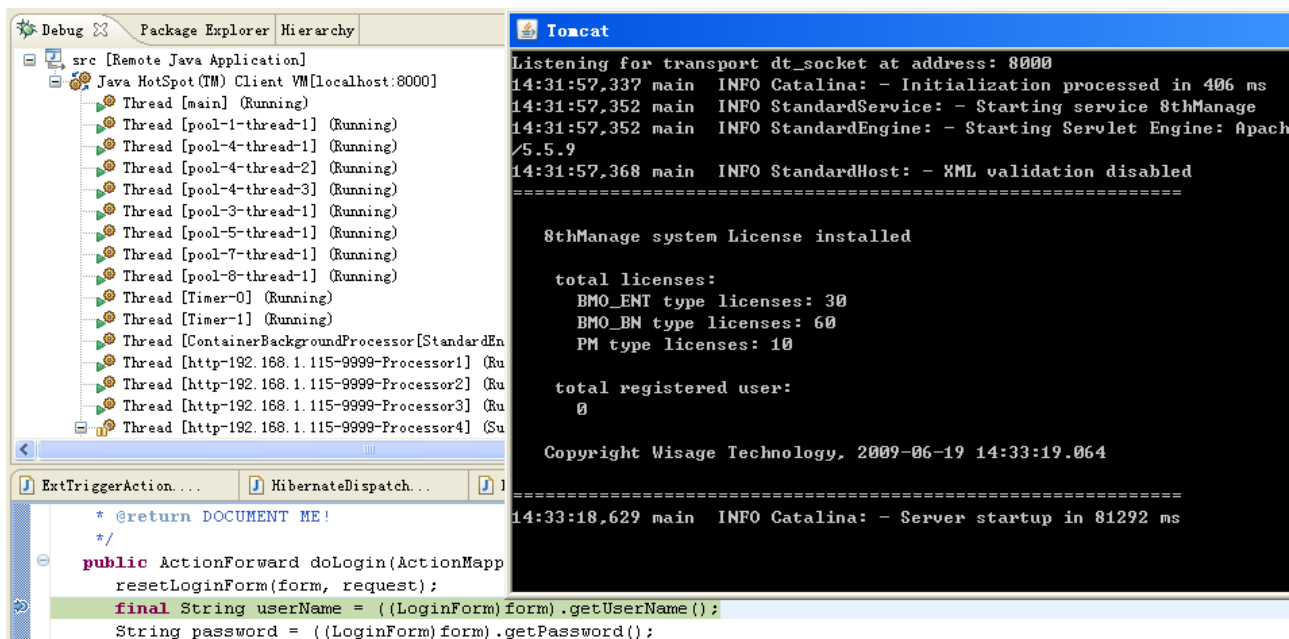


把 8Manage 基础上添加的源代码添加进来。当然，这些源代码编译后的 classes 应该发布到了 C:\WisageTech_installer\8thManage\webApps\8thManage\WEB-INF\classes 下的了。



按下 Debug 按钮后，启动过程如下图。





另外, JSP 的编译成 Java 后, 可以在 `C:\WisageTech_installer`

`\8thManage\instances\Lansing2\node1\conf\8thManage\localhost` 下的 `org` 目录找到。

3.9 无刷新多文件上传及文档链接功能的使用

3.9.1 概述

系统很多地方需要使用到多文件上传功能, 为了避免文件上传过程中的页面刷新以及大文件上传出错后表单信息的获取, 本模块使用 `iframe` 来完成文件的伪 Ajax 上传。

3.9.2 使用步骤

A. 引入相关类及 js 及出错用到的 error table:

```
<%@ include file="/GTcommonJspHeader2.jsp" %>
<%@page import=" com.gearteks.framework.controller.AttachmentHelper"%>
<script type='text/javascript'
src='<%=request.getContextPath()%>/js/GTAjaxTable.js'> </script>
<script type="text/javascript" src="<%=request.getContextPath()%>/js/gtQuery.js"> </script>
```

```
<%@ include file="/errorTable2.inc"%>
```

B. 创建一个 button 用来浏览选取本地文件或链接文档库文件:

上传文件按钮:

```
<%=AttachmentHelper.genAjaxUploadFileButton(String ownerType, Integer ownerId, String title)%>
```

链接文档库文件按钮:

```
<%=AttachmentHelper.genAjaxAddDocButton(String ownerType, Integer ownerId, String title)%>
```

移动附件到文档库按钮:

```
<%=AttachmentHelper.genAjaxMoveToLibButton(String ownerType, Integer ownerId) %>
```

ownerType: 附件所属对象类型。为了避免重复, 请使用公司名加下划线再加名称, 如

WisageTech_xxx

ownerId: 附件所属对象 ID, 如果是新增页面, 请传入 null

title: 按钮名称(不传或传 null 的话, 默认为 New Attachment)。

C. 创建用来显示上传文件的区域:

```
<%=AttachmentHelper.genAjaxUploadFileContent(HttpServletRequest request, String ownerType, Integer ownerId) %>
```

ownerType: 附件所属对象类型。为了避免重复, 请使用公司名加下划线再加名称, 如

WisageTech_xxx

ownerId: 附件所属对象 ID, 如果是新增页面, 请传入 null

D. 如果是新增页面, 请在后台 action 中加上代码用来保存相应的上传文件, 否则不需要调

用:

```
AttachmentHelper.updateMulAttachments(HttpServletRequest request, boolean hasErrors, String ownerType, Integer ownerId)
```

hasErrors: 后台是否有出错信息

ownerType: 附件所属对象类型。为了避免重复, 请使用公司名加下划线再加名称, 如

WisageTech_xxx

ownerId: 附件所属对象 ID

E. 页面文件列表显示:

```
<%=AttachmentHelper.genAjaxAttachmentListTable(HttpServletRequest request, String
ownerType, Integer ownerId, boolean editAllowed, Integer pageSize, boolean
isShowCheckBox)%>
```

ownerType: 附件所属对象类型。为了避免重复, 请使用公司名加下划线再加名称, 如

WisageTech_xxx

ownerId: 附件所属对象 ID

editAllowed: 是否有新增及删除附件的权限

pageSize: 每页显示文件个数

isShowCheckBox: 是否显示移动文档库的文件复选框, 当需要时, 传入 true, 否则传入

false

示例代码, 此示例代码在 8Manage 环境下, 拷贝到一个 JSP 就可以运行

```
<%@ page contentType="text/html; charset=UTF-8"%>
<%@ include file="/GTcommonJspHeader2.jsp"%>
<%@page import=" com.gearteks.framework.controller. AttachmentHelper "%>
<html>
<head>
<%@ include file="/css.inc" %>
<script type='text/javascript'
src='<%=request.getContextPath()%>/js/GTAjaxTable.js'> </script>
<script type="text/javascript" src="<%=request.getContextPath()%>/js/gtQuery.js"> </script>
<script>
</script>
</head>
```

```

<body scroll="no">
<table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0"
class="titleTop">
<tr><td>
  <table width="100%" height="100%" border="0" cellpadding="0" cellspacing="0" >
    <tr><td valign="top">
      <table width="100%" border="0" cellpadding="0" cellspacing="0"
id="tblHead">
        <tr>
          <td width="10"></td>
          <td>&nbsp;</td>
          <td></td>
          <td align="right">
            <table border="0" cellpadding="0" cellspacing="0" >
              <tr>

                <td><%=AttachmentHelper.genAjaxUploadFileButton("WisageTech_User", 1)%></td>
                  <td width="3" ></td>
                  <td width="10" align="center" ></td>
                  <td width="3" ></td>

                <td><%=AttachmentHelper.genAjaxAddDocButton("WisageTech_User", 1)%></td>
                  <td width="3" ></td>
                  <td width="10" align="center" ></td>
                  <td width="3" ></td>

                <td><%=AttachmentHelper.genAjaxMoveToLibButton("WisageTech_User", 1)%></td>
                  </tr>
                </table>
              </td>
              <td width="10"></td>
            </tr>
          </table>
          <table width="100%" border="0" align="center" cellpadding="0"
cellspacing="0" >
            <tr><td class="Linetop"></td></tr><tr><td height="10"></td></tr>
          </table>
          <%@ include file="/errorTable2.inc"%>
          <table width="100%" border="0" cellpadding="0" cellspacing="0">
            <tr>
              <td><%=AttachmentHelper.genAjaxUploadFileContent(request, "
WisageTech_User", 1) %></td>
            </tr>
          </table>
        </tr>
      </table>
    </td>
  </tr>
</table>

```

```

        <td><%=AttachmentHelper.genAjaxAttachmentListTable(request, "
WisageTech_User", 1, true, 20, true)%></td>
        </tr>
    </table>
</td></tr>
</table>
</body>
</html>

```

3.10 通用审计跟踪记录功能的实现

3.10.1 概述

审计跟踪记录是用来记录用户对数据库各字段操作的情况，主要是把操作人，操作时间，操作方式，操作类型，操作前后数据等记录在相应的 log 表里，可以用 trigger 记录进去，也可以用程序记录进去。

3.10.2 log 标准表结构

```

CREATE TABLE strategy_log(
    LogId          int          IDENTITY(1,1) NOT NULL, //自增长 id
    Event          type3       NOT NULL, //操作类型
    StrategyId    id          NOT NULL, //操作主表 id
    FieldType     type3       NULL, //操作数据类型
    FieldName     string2     NULL, //自定义字段名称
    FieldId       id          NULL, //自定义字段 id
    ExtendId      id          NULL, //自定义字段类型 id

    NewValueStr   string3     NULL, //新字符串值
    NewValueNum   float       NULL, //新数字值
    NewValueDate  datetime    NULL, //新日期值
    NewValueId    id          NULL, //新整开值
    NewObjType    string1     NULL, //新对象类型值
    NewObjId      id          NULL, //新对象 id 值

    OldValueStr   string3     NULL, //旧字符串值

```

```

OldValueNum      float      NULL,      //旧数字值
OldValueDate     datetime   NULL,      //旧日期值
OldValueId       id          NULL,      //旧整型值
OldObjType       string1    NULL,      //旧对象类型值
OldObjId         id          NULL,      //旧对象 id 值

TS              ts          CONSTRAINT [DF1_strategy_log] DEFAULT (getdate()) NOT NULL, //操作时
间
TSUserId        id          NOT NULL     //操作用户
)
go
ALTER TABLE strategy_log ADD
    CONSTRAINT PK_strategy_log PRIMARY KEY CLUSTERED (LogId)
go
ALTER TABLE strategy_log ADD CONSTRAINT IF1_strategy_log
    FOREIGN KEY (StrategyId)
    REFERENCES strategy(StrategyId) ON DELETE CASCADE
go
CREATE INDEX IE1_strategy_log ON strategy_log (StrategyId, TS, Event)
go

```

注意：在 company_log,job_log,表中 NewValueStr 对应是 NewValueString, OldValueStr 对应是 OldValueString。

3.10.3 java 类及 hibernate 配置文件

Java 类统一继承超类 `com.gearteks.framework.object.AbstractObjectLog`, 请参考 `StrategyLog`

对应该 hibernate 配置文件也请参考 `StrategyLog.hbm.xml`

3.10.4 用 SQL, trigger 把记录写到 log 表, 及写资源文件

key-value 值

示例代码: SQL 插入代码

```

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueStr, OldValueStr, TS, TSUserId)
values('Mod_StrategyName', 'Str', 1, 'NewName', 'OldName', getdate(), 1);

```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
values('Mod_StrategyProgress', 'Num', 1, 100, 80, getdate(), 1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueDate, OldValueDate, TS,
TSUserId)
values('Mod_StrategyDealDate', 'Date', 1, '2009-10-29', '2009-10-21', getdate(), 1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, NewValueStr,
OldObjType, OldObjId, OldValueStr, TS, TSUserId)
values('Mod_StrategyCurrency', 'Select', 1, 'Currency', 3, 'CNY', 'Currency', 1, 'USD', getdate(),
1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
values('Mod_StrategyIsOutSourcing', 'Yesno', 1, 1, 0, getdate(), 1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, TS, TSUserId)
values('Add_StrategyResp', 'SystemObj', 1, 'User', 17, getdate(), 1);
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, OldObjType, OldObjId, TS, TSUserId)
values('Del_StrategyResp', 'SystemObj', 1, 'User', 5, getdate(), 1);
```

在 trigger 中的使用: Strategy_u.trg

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueStr, OldValueStr, TS, TSUserId)
SELECT 'Mod_StrategyName', 'Str', i.StrategyId, i.StrategyName, d.StrategyName, getdate(),
i.TSUserId
FROM inserted i, deleted d
WHERE i.StrategyId = d.StrategyId AND
i.StrategyName != d.StrategyName;
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
SELECT 'Mod_StrategyProgress', 'Num', i.StrategyId, i.Progress, d.Progress, getdate(),
i.TSUserId
FROM inserted i, deleted d
WHERE i.StrategyId = d.StrategyId AND
i.Progress != d.Progress;
```

```
INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueDate, OldValueDate, TS,
TSUserId)
SELECT 'Mod_StrategyDealDate', 'Date', i.StrategyId, i.DealDate, d.DealDate, getdate(),
i.TSUserId
```



```

FROM inserted i, deleted d
WHERE i.StrategyId = d.StrategyId AND
      i.DealDate != d.DealDate;

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, NewValueStr,
OldObjType, OldObjId, OldValueStr, TS, TSUserId)
  SELECT 'Mod_StrategyCurrency', 'Select', i.StrategyId, 'Currency', i.CurrencyId,
newCurrency.CurrencyName, 'Currency', d.CurrencyId, oldCurrency.CurrencyName, getdate(),
i.TSUserId
  FROM inserted i, deleted d, currency newCurrency, currency oldCurrency
  WHERE i.StrategyId = d.StrategyId AND
        i.CurrencyId != d.CurrencyId AND
        i.CurrencyId = newCurrency.CurrencyId AND
        d.CurrencyId = oldCurrency.CurrencyId;

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewValueNum, OldValueNum, TS,
TSUserId)
  SELECT 'Mod_StrategyIsOutSourcing', 'Yesno', i.StrategyId, i.IsOutSourcing, d.IsOutSourcing,
getdate(), i.TSUserId
  FROM inserted i, deleted d
  WHERE i.StrategyId = d.StrategyId AND
        i.IsOutSourcing != d.IsOutSourcing;

```

strategy_member_i.trg

```

INSERT INTO strategy_log (Event, FieldType, StrategyId, NewObjType, NewObjId, TS, TSUserId)
  SELECT 'Add_StrategyResp', 'SystemObj', i.StrategyId, i.ObjType, i.ObjId, getdate(), s.TSUserId
  FROM inserted i, strategy s
  WHERE i.StrategyId = s.StrategyId
        i.Involvement = 'Resp';

```

strategy_member_d.trg

```

INSERT INTO strategy_log (Event, FieldType, StrategyId, OldObjType, OldObjId, TS, TSUserId)
  SELECT 'Del_StrategyResp', 'SystemObj', d.StrategyId, d.ObjType, d.ObjId, getdate(),
s.TSUserId
  FROM deleted d, strategy s
  WHERE d.StrategyId = s.StrategyId
        d.Involvement = 'Resp';

```

以上所有 Event 请在中英文资源文件里写入对应的 key-value 值，格式如下

```
name.objectLog.Mod_StrategyName=Strategy Name
name.objectLog.Mod_StrategyProgress=Progress
```

请参照本文“4.8 定制化字段名”

注意:

对于引用的是会被删除的对象:

则在往 NewObjType, NewObjId 及 OldObjType, OldObjId 写入值时, 请别忘了往 NewValueStr 及 OldValueStr 里写入当时对象的名称, 如上面例子中的 currency, 这样, 当这被引用的 currency 没有删除时, 程序会用 NewObjType, NewObjId 去取当前的 currency, 如果 currency 被删除了, 则取以前保存的 NewValueStr 作为 currency 的名称。

对于引用的是不会被删除的对象:

则不需要存入 NewValueStr 及 OldValueStr, 如 User, Party, Company。

就像上面 strategy_member_i.trg 及 strategy_member_d.trg, 就没有往 NewValueStr 和 OldValueStr 写入对象的名称。

3.10.5 列表查询语句, 及表格 JSP 代码

一般查询某个对象的审记跟踪记录, 只要查询相应的 log 表, 如 strategy 查询语句示便如下

SQL:

```
select * from strategy_log where StrategyId = ? order by TS desc, LogId desc
```

HQL

```
select sl from StrategyLog sl where sl.strategyId = ? order by sl.ts desc, sl.logId desc
```

列表显示表格代码如下:

```
<gearTable:table name="requestScope.queryResult" id="item"
requestURI="<%=request.getContextPath() +
"/projectManagement/strategy.do?method=doShowLogList"%>" styleClass="displayTag"
pagesize="<%=pageSize%>" defaultsort="2" sort="list" border="0" width="100%"
cellspacing="1" cellpadding="0" needPagination="true">
  <gearTable:column align="left"
title="<%=SessionUtil.getMessage(request,"display.fieldName.date")%>" sortable="true"
headerClass="sortable" width="128" colExtInfo="ts">
  <%=DateUtil.tsFormat(((AbstractObjectLog)item).getTs(), request)%>
```

```

</gearTable:column>
  <gearTable:column align="left"
title=" <%=SessionUtil.getMessage(request,"name.coLog.changedBy")%> " sortable="true"
headerClass="sortable" colExtInfo="tsUser" >
  <%=HtmlUtil.genLink(((AbstractObjectLog)item).getTsUser())%>
</gearTable:column>
  <gearTable:column align="left"
title=" <%=SessionUtil.getMessage(request,"name.job.Event")%> " sortable="true"
headerClass="sortable" colExtInfo="event" >
  <%=((JobLog)item).getEventDisplay()%>
</gearTable:column>
  <gearTable:column align="left"
title=" <%=SessionUtil.getMessage(request,"name.coLog.newValue")%> " sortable="false"
headerClass="sortable" >
  <%=StringUtil.nvl(((AbstractObjectLog)item).getNewValue(), "")%>
</gearTable:column>
  <gearTable:column align="left"
title=" <%=SessionUtil.getMessage(request,"name.coLog.oldValue")%> " sortable="false"
headerClass="sortable" >
  <%=StringUtil.nvl(((AbstractObjectLog)item).getOldValue(), "")%>
</gearTable:column>
  <gearTable:setProperty name="paging.banner.item_name"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.item_name",
null) %> " />
  <gearTable:setProperty name="paging.banner.items_name"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.items_name",
,null) %> " />
  <gearTable:setProperty name="paging.banner.all_items_found"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.all_items_fou
nd",null) %> " />
  <gearTable:setProperty name="paging.banner.some_items_found"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.some_items_
found",null) %> " />
  <gearTable:setProperty name="paging.banner.no_items_found"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.no_items_fo
und",null) %> " />
  <gearTable:setProperty name="paging.banner.one_item_found"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.one_item_fo
und",null) %> " />
  <gearTable:setProperty name="basic.msg.empty_list"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"basic.msg.empty_list",null)
%> " />
  <gearTable:setProperty name="paging.banner.onepage"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.onepage",nu
ll) %> " />
  <gearTable:setProperty name="paging.banner.page.selected"

```

```

value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.page.selected",null) %>" />
        <gearTable:setProperty name="paging.banner.page.link"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.page.link",null) %>" />
        <gearTable:setProperty name="paging.banner.page.separator"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.page.separator",null) %>" />
        <gearTable:setProperty name="paging.banner.last"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.last",null) %>" />
        <gearTable:setProperty name="paging.banner.first"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.first",null) %>" />
        <gearTable:setProperty name="paging.banner.full"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.full",null) %>" />
        <gearTable:setProperty name="paging.banner.placement"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.placement",null) %>" />
        <gearTable:setProperty name="basic.msg.empty_list_row"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"basic.msg.empty_list_row",null) %>" />
        <gearTable:setProperty name="paging.banner.group_size"
value=" <%=TagUtils.getInstance().message(pageContext,null,null,"paging.banner.group_size",null) %>" />
    </gearTable:table>

```

3.11 发送自定义邮件

利用类 `com.gearteks.imate.common.trigger.NotificationHandler`，您可以定制从 8Manage 系统发出给用户的电子邮件。下面的代码例子可定制以下情景的自动电子邮件：当用户点选“**转为正式**”按钮后，系统发送邮件给负责人。

- 配置：

```

<action actionPoint="strategy|saveStrategy" location="post"
    sameTransaction="true" useForward="false"
    actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
    actionMethod="sentEmailWhenOfficial" />

```

● 代码示例:

```
public class CustomizedStrategyAction implements Callbackable {
    public ActionForward sentEmailWhenOfficial(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
        throws ServletException {
        StrategyForm strategyForm = (StrategyForm)form;
        Integer strategyId = strategyForm.getStrategyId();
        Strategy strategy = null;
        if (strategyId != null) {
            strategy = (Strategy)DbObject.find(Strategy.class, strategyId, true);
        }

        Strategy srcStrategy =
        (Strategy)request.getAttribute(StrategyAlertFunction.OLD_STRATEGY);
        String oldStatus = srcStrategy.getStatus();

        if (strategy != null) {
            boolean becomeOfficial = Strategy.STATUS_APPROVE.equals(strategy.getStatus()) &&
            Strategy.adjustAllowed(oldStatus) &&
            strategy.isNoApproval(null, null, null, null, null, null);
            if (becomeOfficial) {
                NotificationHandler nh = new NotificationHandler();
                Message msg = new Message();

                Collection resps = strategy.getResponsibleMembers();
                Collection clients = StrategyLogic.getInstance().getStrategyClients(strategyId);
                String clientName = CollectionUtil.getNames(clients);

                msg.addReceivers(resps);
                msg.setSubject(strategy.getName());
                msg.setContent(strategy.getName() + " - " + clientName);

                nh.sendMessage(msg, null, false, true, false);
            }
        }
        return null;
    }
}
```

下面解释上述代码示例中的参数 `NotificationHandler.sendMessage (Message msg, Template`

`template, boolean filterByUserSetting, boolean withSuspended, boolean filterContact)` API。

- msg - 此参数包含电子邮件主题, 内容, 收件人, 抄送, 秘密抄送, 附件。
- template - 此参数是电子邮件的模板, 如果为空则意味着没有使用电子邮件模板。
- filterByUserSetting - 当此参数值为 false, 表示此邮件的发送无需校验用户通知设置 (始终发送)。
- withSuspended - 当此参数值为 true, 表示邮件也将发送给“被停用”的系统用户。
- filterContact - 当此参数值为 false, 表示此邮件的发送对象无需过滤联系人用户。

3.12 自定义查询条件

用户可自定义查询页面的查询条件。以下代码示例是如果在机会/策略列表页面的查询条件中加入自定义条件。

- 配置:

```
<action actionPoint="strategy|listStrategy" location="pre"
  sameTransaction="true" useForward="false"
  actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
  actionMethod="setCustomizedCriteria" />
```

- 代码示例:

```
public ActionForward setCustomizedCriteria(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response)
  throws GTEException {
  PageInfo pi = SessionUtil.getPageInfo(request);
  if (Strategy.STRATEGY_TYPE.equals(request.getParameter("objType")) &&
  DbFieldValue.STRATEGY_BUSINESSTYPE_SELLING.
  equals(request.getParameter("businessType"))) {
    pi.setHqlConjunction(" exists(select s1.startDate from Strategy s1 where s1.startDate >
'2010-01-01 00:00:00' and s1.strategyId = s.strategyId) ");
    pi.setSqlConjunction(" exists(select s1.startDate from Strategy s1 where s1.startDate >
'2010-01-01 00:00:00' and s1.strategyId = s.strategyId) ");
  }
  return null;
}
```

```
}

```

自定义查询条件功能的注意点：

- 当前版本不支持在查询条件中添加参数
- 因为 8Manage 同时用 SQL 和 HQL 实现查询,所以自定义查询必须用 `setSqlConjunction` 和 `setHqlConjunction` 来定义 SQL 和 HQL 查询。
- SQL 和 HQL 在机会/策略列表页面的别名是 “s” , 但是不同的列表页面中此别名可能会不同。如果您需要自定义其它列表页面的查询条件, 请联系 8Manage 客服支持。

3.13 自定义论坛名称

用户用自定义的逻辑来定制论坛的名称。以下示例展示了如何定制机会/策略审批页面的论坛名称。

- 配置：

```
<action actionPoint="strategy|saveStrategy" location="post"
  sameTransaction="true" useForward="false"
  actionClass="com.gearteks.test.customized.CustomizedStrategyAction"
  actionMethod="resetDiscussionName" />
```

- 代码示例：

```
public ActionForward resetDiscussionName(ActionMapping mapping, ActionForm form,
HttpServletRequest request, HttpServletResponse response)
  throws GTEException {
  if (Strategy.STRATEGY_TYPE.equals(request.getParameter("objType")) &&
    DbFieldValue.STRATEGY_BUSINESSTYPE_SELLING.
    equals(request.getParameter("businessType"))) {
    StrategyForm strategyForm = (StrategyForm)form;
    Integer strategyId = strategyForm.getStrategyId();
    Strategy strategy = null;
    if (strategyId != null) {
      strategy = (Strategy)DbObject.find(Strategy.class, strategyId, true);
    }

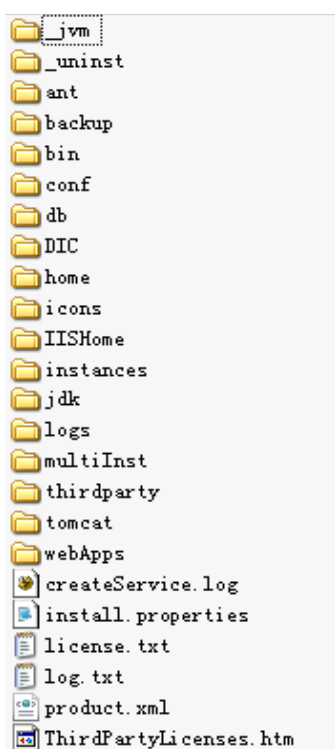
    if (strategy != null) {
```

```
Approval appr = strategy.getCurrentApproval();
if (appr != null) {
    Discussion discussion = ApprovalHelper.getFirstDiscussion(appr);
    discussion.setName("Here is a test...");
}
}
}
return null;
}
```

4 8Manage 目录结构

4.1 8Manage 现有目录介绍

%WisageTech_HOME%\8thManage 目录下主要文件夹如下:



8Manage 下的涉及二次开发的目录:

- Home\localhost\customize\callback: 请求回调机制配置文件

- Home\localhost\customize\customize: 客制化系统属性配置文件
- Home\localhost\customize\menu: 客制化菜单配置文件
- Home\localhost\customize\resources: 客制化资源配置文件
- instances\%CELL_NAME%\%NODE_NAME%\conf\8thManage\localhost: 放置系统配置文件, 包括: Excel 导入/导出的列定义, log4j 设置, 系统属性设置。
- logs: 存放 8Manage 系统的 log
- instances\%CELL_NAME%\%NODE_NAME%\logs: 存放 tomcat 的 log

WebApps 目录介绍



8Manage 的子目录包括:

approval: 系统审批相关页面

attachment: 系统附件相关页面

changeOrder: 变更请求功能相关页面

chat: 即时聊天页面

comment: 注释页面

common: 公用页面, 如提示信息

css: 存放 CSS 样式

customization: 定制化的页面

customizeReport: 定制化的报表页面

discussion: 论坛页面

download: 系统模板

ext: 第三方系统集成页面

help: 帮助手册页面

home: “我的工作台”的几个功能, 包括 Home Dashboard\Timesheet\My Report\Expense

images: 图片

integration: 集成所需页面

js: JS 存放目录

library: 文档库

m: 手机访问涉及的页面

maintain: 系统主要对象, 包括: 用户、部门、产品、资源、工时表、系统设置、审批等。

menu: 菜单

oais: OA 相关功能及 HR 相关功能页面

popup: 弹出框

projectManagement: 项目管理相关页面

psa: 订单、销售单、费用清单等相关页面

strategy: 邮件营销和费用的报表

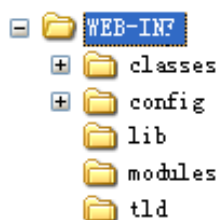
support: 技术支持相关页面

trakingmatrix: 项目矩阵相关页面

vm: 概览页面模板

WEB-INF: 编译后的 JAVA 代码

WEB-INF 目录介绍



classes: JAVA 编译后的 class 文件目录

config: struts、axis2 和菜单配置文件

lib: 8Manage 所有的 jar 包

modules: axis2 所使用到的模块

tld: 标签定义文件, 包括 struts 标签和自定义标签

com 目录介绍: 该目录主要存储 8Manage 的源码, 部分源码在发布时会经过混淆。



gearteks: 该目录存放系统架构源码及各业务模块代码

wisagetech: 该目录存放二次开发框架代码及 WisageTech 实现的客制化代码

xxx: 用于存放客户开发的客制化功能

4.2 客制化目录建议

为避免客制化代码与现有系统代码混淆, 或互相覆盖的情况, 建议客制化代码放在统一目录下。例

如:

com.xxx.xxx.report。同时 JSP 文件也建议放在单独的目录下，例如：webApps\8thManage\xxx。

5 客制化代码部署方式

5.1 JSP 代码部署

- JSP 代码直接复制到%WisageTech_HOME%\8thManage\webApps\8thManage\xxx 目录下。xxx 目录用于存放客制化 JSP 代码的目录。
- 删除%WisageTech_HOME%\8thManage\instances\%CELL_NAME%\%NODE_NAME%\conf\8thManage\localhost\org\apache\jsp\xxx 目录下原有的文件。
- 重启 8Manage 服务。

5.2 JAVA 代码部署

- 将 JAVA 代码编译后得到的 class 字节码复制到%WisageTech_HOME%\8thManage\webApps\WEB-INF\classes\com\xxx 目录下。xxx 目录用于存放客制化 JAVA 代码的目录。
- 重启 8Manage 服务。

5.3 配置文件部署

- 在原有的配置文件基础上修改原有的值或新增新的值。
- 新增新的配置文件。
- 重启 8Manage 服务

8Manage 为企业管理提供标准产品和定制服务的最佳组合，在亚洲有超过 500 家企业正在使用我们的以下模块（包括本地部署或 SaaS）：

8Manage PPM：项目和项目组合管理

8Manage SRM：供应商管理、电子采购和电子招标

8Manage 工时表：资源时间和成本管理

8Manage CRM：企业客户 CRM 和零售业务 CRM

8Manage ITSM：服务管理

8Manage HCM：人力资本管理

8Manage OA：办公自动化

8Manage EDMS：电子文档管理系统

8Manage 看板：可视化的敏捷和精益

8Manage ERP：企业全自动化



扫码关注更多