

若依前后端不分离版整合闲鹿 workflow

若依前后端不分离版整合闲鹿 workflow

- 一、本地安装 workflow 依赖
- 二、获取若依最新版
- 三、整合 workflow 步骤
 - 1、新建业务模块 module
 - 2、修改依赖配置 pom.xml
 - 3.2.1 ruoyi-business 依赖 xianlu-activiti
 - 3.2.2 ruoyi-admin 依赖 xianlu-activiti 和 ruoyi-business
 - 3、修改应用配置 application.yml
 - 3.3.1 application-druid.yml
 - 3.3.2 application.yml
 - 4、跨域支持（可选）
 - 5、修改启动类
 - 6、初始化 workflow 表
- 四、业务表单集成流程
 - 1、代码生成表单 CRUD
 - 2、员工转正列表 JS 增强
 - 3、流程设计与部署
 - 4.3.1 流程设计
 - 4.3.2 流程部署
 - 4、表单和流程关联
 - 4.4.1 业务表流程定义关系
 - 4.4.2 业务表页面映射关系
 - 5、发起员工转正流程
 - 6、流程中心
- 五、在线表单集成流程
 - 1、启动表单设计器
 - 5.1.1 安装 fis3
 - 5.1.2 修改后端接口地址
 - 5.1.3 指定端口启动表单设计器前端服务
 - 2、表单设计
 - 3、表单列表
 - 4、流程设计与部署
 - 5、表单和流程关联
 - 5.5.1 业务表流程定义关系(在线表单)
 - 6、发起在线表单流程
 - 7、流程中心
- 六、其他
 - 1、Linux 部署项目，流程图文字乱码
 - 2、Linux 部署项目，功能按钮、发起权限列表报错

若依版本	闲鹿 workflow 版本	作者	联系方式
任意	V1.x	闲鹿软件开发工作室	微信(手机) 13123392160、QQ 1055202292

一、本地安装工作流依赖

执行以下命令即可安装 `xianlu-activiti-1.0-SNAPSHOT.jar` 到本地 maven 仓库。

```
mvn install:install-file -Dfile=xianlu-activiti-1.0-SNAPSHOT.jar -
-DgroupId=com.xianlu.activiti -DartifactId=xianlu-activiti -Dversion=1.0-SNAPSHOT -
-Dpackaging=jar
```

» OS (C:) » devSoft » repository » com » xianlu » activiti » xianlu-activiti » 1.0-SNAPSHOT

名称	修改日期	类型	大小
 _remote.repositories	2022/7/5 21:23	REPOSITORIES ...	1 KB
 maven-metadata-local.xml	2022/7/5 21:23	XML 文件	1 KB
 xianlu-activiti-1.0-SNAPSHOT.jar	2022/7/5 21:23	Executable Jar File	7,523 KB
 xianlu-activiti-1.0-SNAPSHOT.pom	2022/6/4 11:18	POM 文件	12 KB

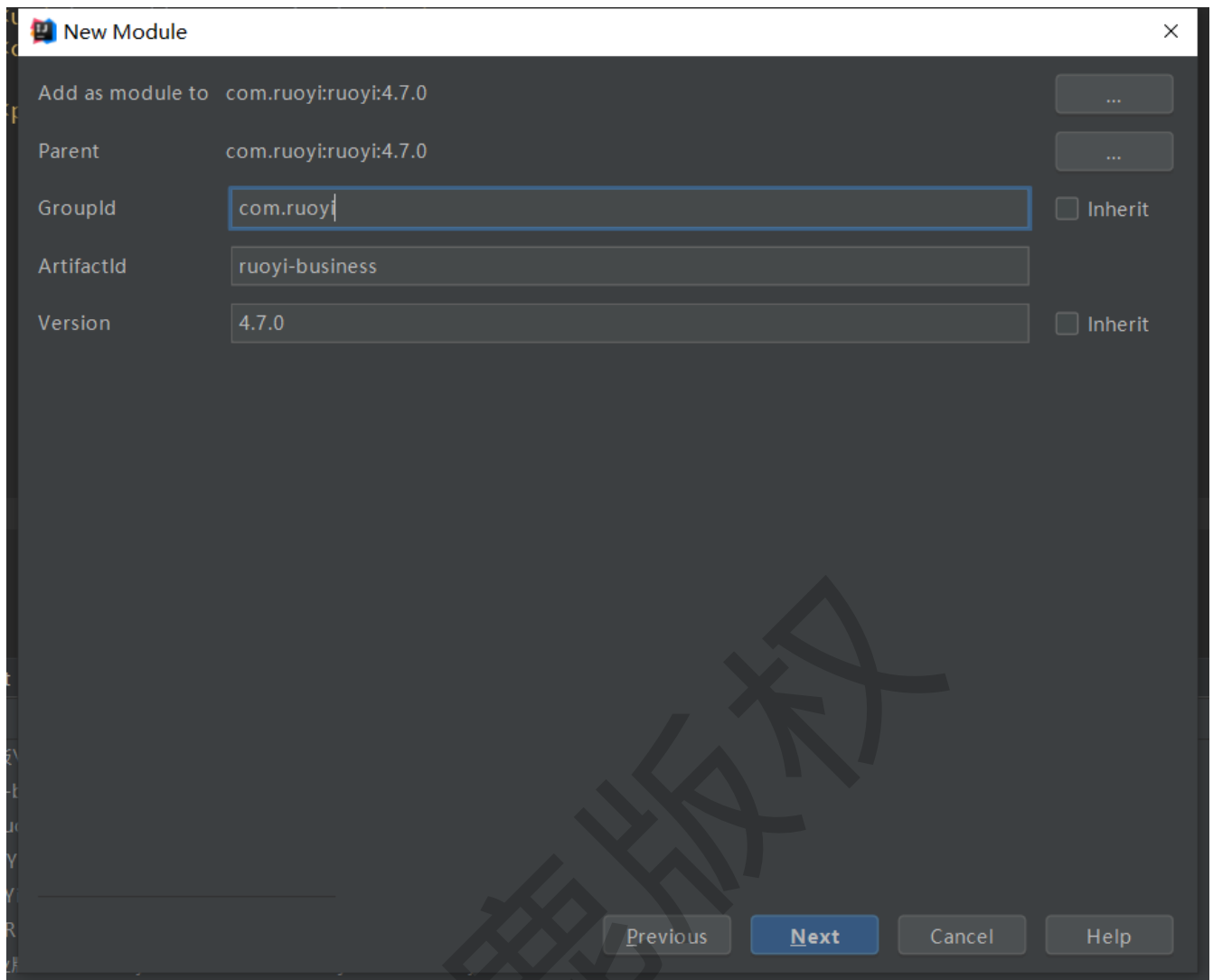
二、获取若依最新版

[码云下载](#)

三、整合工作流步骤

1、新建业务模块 module

新建业务模块 `ruoyi-business`，模块名称随意，或者直接用 `ruoyi-admin` 也行，视具体项目情况而定。



2、修改依赖配置 pom.xml

3.2.1 ruoyi-business 依赖 xianlu-activiti

```
<!-- 核心模块-->
<dependency>
    <groupId>com.ruoyi</groupId>
    <artifactId>ruoyi-framework</artifactId>
</dependency>
<!-- 闲鹿 workflow 依赖 start-->
<dependency>
    <groupId>com.xianlu.activiti</groupId>
    <artifactId>xianlu-activiti</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
```

```
<artifactId>mysql-connector-java</artifactId>
<scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
</dependency>
<!--Shiro核心框架 -->
<dependency>
  <groupId>org.apache.shiro</groupId>
  <artifactId>shiro-core</artifactId>
  <version>1.8.0</version>
</dependency>
<!--客户端 jwt-->
<dependency>
  <groupId>com.auth0</groupId>
  <artifactId>java-jwt</artifactId>
  <version>3.4.0</version>
</dependency>
<!-- spring security 安全认证 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<!-- websocket -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-websocket</artifactId>
</dependency>
<dependency>
  <groupId>org.activiti</groupId>
  <artifactId>activiti-spring-boot-starter-basic</artifactId>
  <version>6.0.0</version>
  <exclusions>
    <exclusion>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<!--activiti modeler 5.22 start-->
<dependency>
  <groupId>org.activiti</groupId>
  <artifactId>activiti-json-converter</artifactId>
  <version>6.0.0</version>
  <exclusions>
    <exclusion>
      <groupId>org.activiti</groupId>
      <artifactId>activiti-bpmn-model</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<!-- xml解析依赖-->
```

```
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-codec</artifactId>
  <version>1.7</version>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-css</artifactId>
  <version> 1.7</version>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-svg-dom</artifactId>
  <version>1.7</version>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-svggen</artifactId>
  <version>1.7</version>
</dependency>
<!-- xml解析依赖-->
<!-- Swagger3依赖 -->
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-boot-starter</artifactId>
  <version>3.0.0</version>
  <exclusions>
    <exclusion>
      <groupId>io.swagger</groupId>
      <artifactId>swagger-models</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<!-- oss -->
<dependency>
  <groupId>com.qiniu</groupId>
  <artifactId>qiniu-java-sdk</artifactId>
  <version>[7.2.0, 7.2.99]</version>
</dependency>
<dependency>
  <groupId>com.aliyun.oss</groupId>
  <artifactId>aliyun-sdk-oss</artifactId>
  <version>2.5.0</version>
</dependency>
<dependency>
  <groupId>com.qcloud</groupId>
  <artifactId>cos_api</artifactId>
  <version>5.5.9</version>
  <exclusions>
    <exclusion>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

```

    </exclusions>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-aop</artifactId>
</dependency>
<dependency>
    <groupId>de.schlichtherle.truelicense</groupId>
    <artifactId>truelicense-core</artifactId>
    <version>1.33</version>
</dependency>
<!-- 闲鹿工作流依赖 end-->

```

3.2.2 ruoyi-admin 依赖 xianlu-activiti 和 ruoyi-business

```

<!-- 闲鹿工作流依赖 start-->
<dependency>
    <groupId>com.xianlu.activiti</groupId>
    <artifactId>xianlu-activiti</artifactId>
    <version>1.0-SNAPSHOT</version>
</dependency>
<dependency>
    <groupId>com.ruoyi.business</groupId>
    <artifactId>ruoyi-business</artifactId>
    <version>4.7.0</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
<!--Shiro核心框架 -->
<dependency>
    <groupId>org.apache.shiro</groupId>
    <artifactId>shiro-core</artifactId>
    <version>1.8.0</version>
</dependency>
<!--客户端 jwt-->
<dependency>
    <groupId>com.auth0</groupId>
    <artifactId>java-jwt</artifactId>
    <version>3.4.0</version>
</dependency>

```

```
<!-- spring security 安全认证 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<!-- websocket -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-websocket</artifactId>
</dependency>
<dependency>
  <groupId>org.activiti</groupId>
  <artifactId>activiti-spring-boot-starter-basic</artifactId>
  <version>6.0.0</version>
  <exclusions>
    <exclusion>
      <groupId>org.mybatis</groupId>
      <artifactId>mybatis</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<!--activiti modeler 5.22 start-->
<dependency>
  <groupId>org.activiti</groupId>
  <artifactId>activiti-json-converter</artifactId>
  <version>6.0.0</version>
  <exclusions>
    <exclusion>
      <groupId>org.activiti</groupId>
      <artifactId>activiti-bpmn-model</artifactId>
    </exclusion>
  </exclusions>
</dependency>
<!-- xml解析依赖-->
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-codec</artifactId>
  <version>1.7</version>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-css</artifactId>
  <version> 1.7</version>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-svg-dom</artifactId>
  <version>1.7</version>
</dependency>
<dependency>
  <groupId>org.apache.xmlgraphics</groupId>
  <artifactId>batik-svggen</artifactId>
  <version>1.7</version>
</dependency>
```

```

</dependency>
<!-- xml解析依赖-->
<!-- Swagger3依赖 -->
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-boot-starter</artifactId>
    <version>3.0.0</version>
    <exclusions>
        <exclusion>
            <groupId>io.swagger</groupId>
            <artifactId>swagger-models</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<!-- oss -->
<dependency>
    <groupId>com.qiniu</groupId>
    <artifactId>qiniu-java-sdk</artifactId>
    <version>[7.2.0, 7.2.99]</version>
</dependency>
<dependency>
    <groupId>com.aliyun.oss</groupId>
    <artifactId>aliyun-sdk-oss</artifactId>
    <version>2.5.0</version>
</dependency>
<dependency>
    <groupId>com.qcloud</groupId>
    <artifactId>cos_api</artifactId>
    <version>5.5.9</version>
    <exclusions>
        <exclusion>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-log4j12</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-aop</artifactId>
</dependency>
<dependency>
    <groupId>de.schlichtherle.truelicense</groupId>
    <artifactId>truelicense-core</artifactId>
    <version>1.33</version>
</dependency>
<!-- 闲鹿工作流依赖 end-->

```

3、修改应用配置 application.yml

3.3.1 application-druid.yml

数据库 url 添加参数配置 `&nullCatalogMeansCurrent=true`

例如：

```
url: jdbc:mysql://localhost:3307/ry_20211019?
useUnicode=true&characterEncoding=utf8&zeroDateTimeBehavior=convertToNull&useSSL=true&serverTimezone=GMT%2B8&nullCatalogMeansCurrent=true
```

3.3.2 application.yml

开启 activiti 日志输出（可选）

```
# 日志配置
logging:
  level:
    # 开启 activiti sql 日志
    org.activiti.engine.impl.persistence.entity: debug
```

activiti 相关配置

```
# Spring配置
spring:
  # activiti 模块
  activiti:
    # 添加这个配置就不会一直调用了
    # 在流程引擎启动就激活AsyncExecutor,异步 true false 关闭 （切记关闭）
    async-executor-activate: false
    # 解决启动报错: class path resource [processes/] cannot be resolved to URL because it
    does not exist
    check-process-definitions: false
    # 检测身份信息表是否存在
    db-identity-used: false
```

其他增强配置

```
xianlu:
  # 附件上传位置
  upload: C:/xianlu-activiti
  # 数据库名
  table-schema: ry_20211019
  # 表单设计器前端域名
  form-designer-origin: http://localhost:8082
  # 消息通知域名
  websocket-origin: ws://localhost:81
```

4、跨域支持（可选）

在模块 ruoyi-business (或者 ruoyi-admin) 下新建类 CORSFilter.java, 以便支持**表单设计器**前端跨域请求。

```
/**
 * 允许跨域
 */
@Component
public class CORSFilter implements Filter {

    @value("#{${xianlu.form-designer-origin}'.split(',')}")
    private List<String> formDesignerOrigin;

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain
chain)
        throws IOException, ServletException {
        HttpServletResponse res = (HttpServletResponse) response;
        res.addHeader("Access-Control-Allow-Credentials", "true");
        String originHeader = ((HttpServletRequest) request).getHeader("Origin");
        if (formDesignerOrigin.contains(originHeader)) {
            res.addHeader("Access-Control-Allow-Origin", originHeader);
        } else {
            res.addHeader("Access-Control-Allow-Origin", "https://***.***.com");
        }
        res.addHeader("Access-Control-Allow-Methods", "GET, POST, DELETE, PUT");
        res.addHeader("Access-Control-Allow-Headers", "Content-Type,X-CAF-Authorization-
Token,sessionToken,X-TOKEN,token,Authorization");
        if (((HttpServletRequest) request).getMethod().equals("OPTIONS")) {
            response.getWriter().println("ok");
            return;
        }
        chain.doFilter(request, response);
    }
    @Override
    public void destroy() {
    }
    @Override
    public void init(FilterConfig filterConfig) {
    }
}
```

5、修改启动类

在 ruoyi-admin 下 RuoYiApplication.java 增加以下注解。

```
@ComponentScan({ "com.ruoyi", "com.xianlu.activiti" })
@SpringBootApplication(exclude = {
    DataSourceAutoConfiguration.class,
    org.activiti.spring.boot.SecurityAutoConfiguration.class,
    org.springframework.boot.autoconfigure.security.servlet.SecurityAutoConfiguration.class
})
public class RuoyiApplication {}
```

6、初始化工作流表

工作流引擎表均为 `act_XXX` 开头，在应用首次启动时候会自动创建。

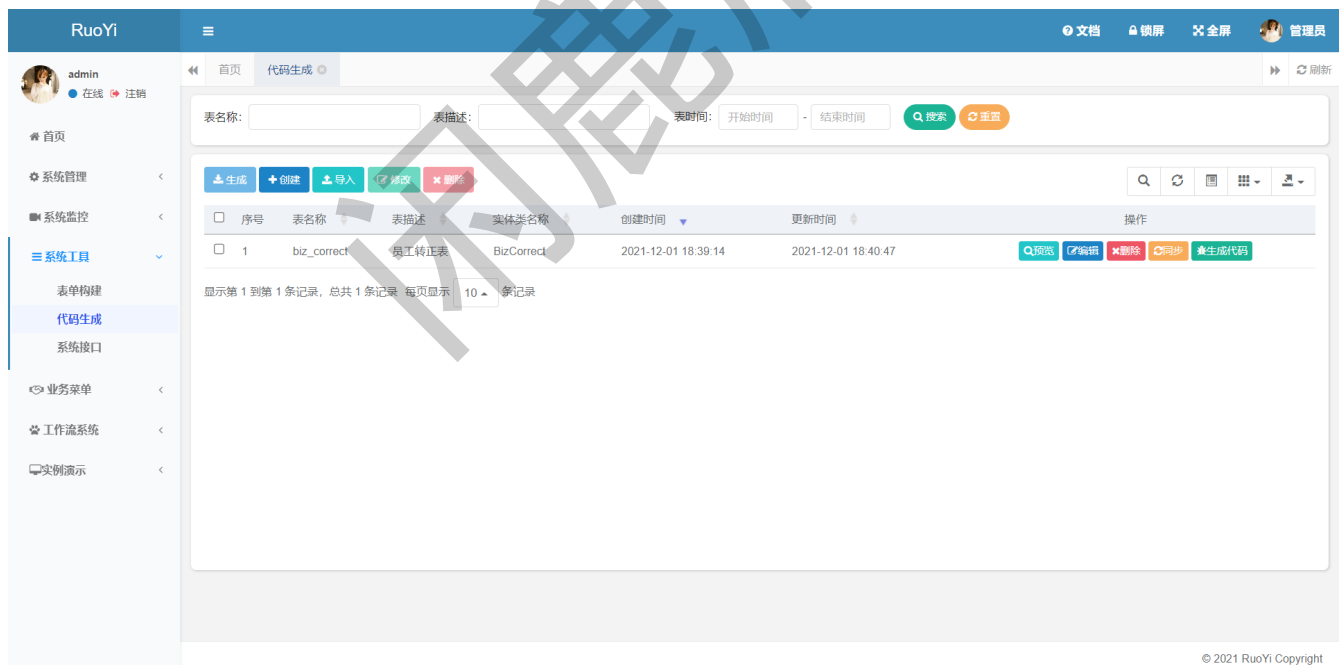
其他表或数据依次执行以下增强 sql 文件即可生成。

```
sql增强.sql
ruoyi_menu增强.sql
```

四、业务表单集成流程

1、代码生成表单CRUD

若依常用功能，不做赘述。如以下生成**员工转正**的业务。



2、员工转正列表JS增强

在员工转正列表 `correct.html` `<script>$(function() {});</script>` 末尾添加工作流增强 js。

```
$.getScript('/activiti/activiti4ry.js');
```

代码示例 leave.html

```
<script th:inline="javascript">
$(function() {
    // ...
    $.table.init(options);

    // 工作流 js 增强
    $.getScript('/activiti/activiti4ry.js');
});
</script>
```

刷新页面，即可查看效果。

员工姓名: 入职日期: 开始时间 - 结束时间 搜索 重置

员工姓名	入职日期	转正理由	流程状态	流程实例ID	操作
张三	2021-12-01	测试委托代理	未启动	-	编辑 删除 提交申请
李四	2021-12-01	测试用户组	未启动	-	编辑 删除 提交申请
1	2022-05-06	测试上级主管2	未启动	-	编辑 删除 提交申请
12	2022-06-07	测试上级主管	未启动	-	编辑 删除 提交申请
admin	-	测试动态SQL用户	未启动	-	编辑 删除 提交申请
测试动态sql	-	测试二	未启动	-	编辑 删除 提交申请
动态用户	-		未启动	-	编辑 删除 提交申请
动态用户三	-		未启动	-	编辑 删除 提交申请
测试动态sql(部门语句)	-		未启动	-	编辑 删除 提交申请

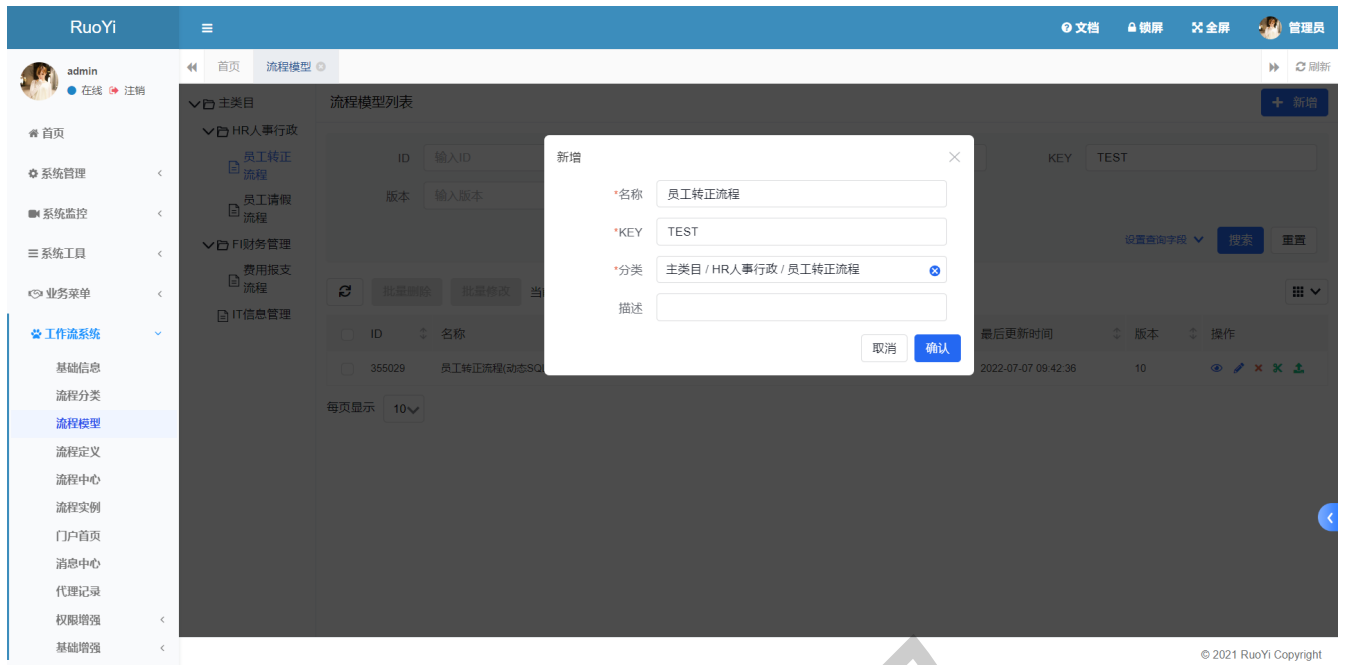
显示第 1 到第 9 条记录, 总共 9 条记录 每页显示 10 条记录

© 2021 RuoYi Copyright

3、流程设计与部署

4.3.1 流程设计

首先需要在 **workflows系统 -> 流程模型**新建流程模型，然后点击**设计**按钮，打开流程在线设计窗口。流程设计好后，别忘了点击左上角保存按钮。





具体的流程设计细节，详见用户使用手册，此处不做过多描述。

4.3.2 流程部署

流程模型设计好后，不能直接使用，需要点击**部署**按钮，发布到流程定义，方能使用。

RuoYi

admin 在线 注销

系统管理 系统监控 系统工具 业务菜单 工作流系统

基础信息 流程分类 流程模型 流程定义 流程中心 流程实例 门户首页 消息中心 代理记录 权限增强 基础增强

流程定义列表

ID 输入ID 名称 输入名称 KEY 输入KEY 版本 输入版本

设置查询字段 搜索 重置

批量删除 当前有 1 条数据。

ID	名称	KEY	分类	版本	状态	部署ID	操作
TEST-2-447542	员工转正流程(动态SQL用户)	TEST	员工转正流程	2	已激活	447539	

每页显示 10

© 2021 Ruoyi Copyright

4、表单和流程关联

为实现业务表单和流程关联，需要在**工作流系统 -> 基础信息**添加以下两条记录。

4.4.1 业务表流程定义关系

新增



*描述

业务表流程定义关系



*KEY

这里填写数据库物理表名

*VALUE

这里填写流程定义KEY

取消

确认

4.4.2 业务表页面映射关系

新增



*描述

业务表页面映射关系



*KEY

这里填写数据库表名

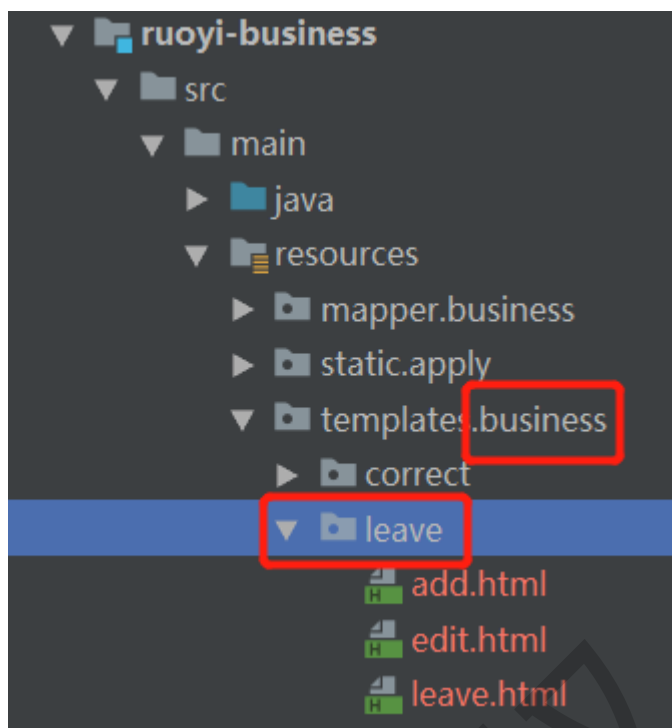
*VALUE

这里填写页面目录路径, 如 /business/leave

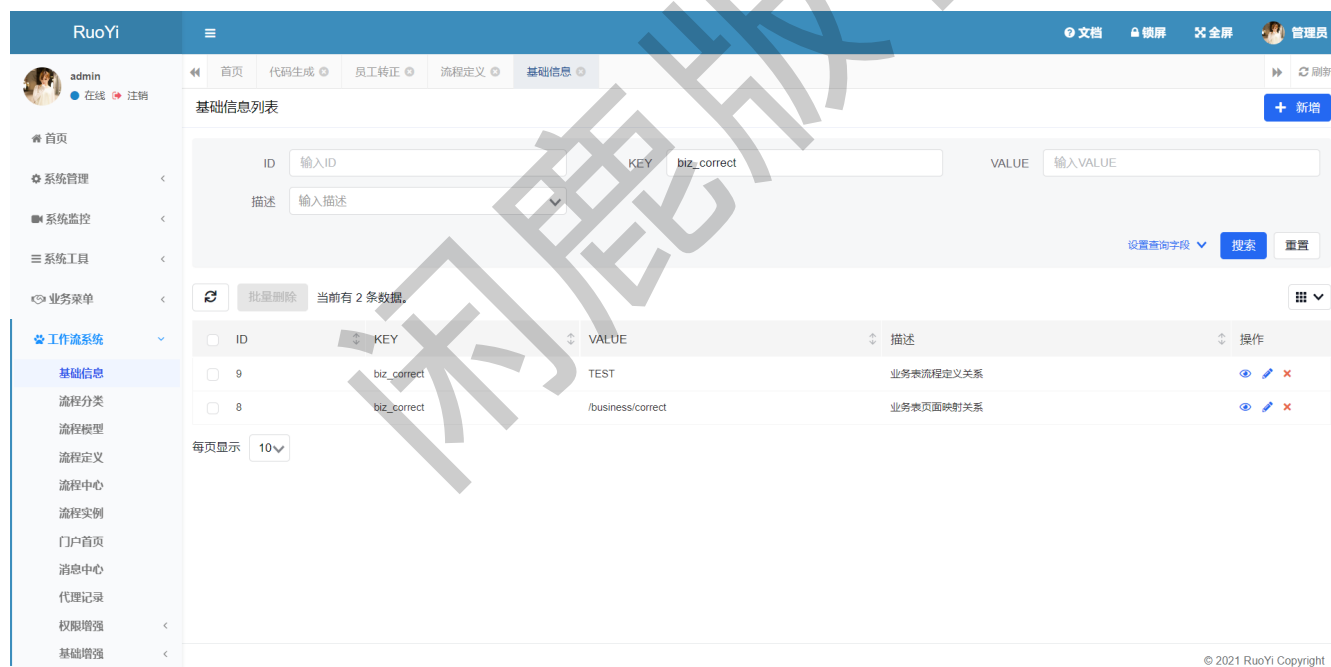
取消

确认

具体页面目录路径如图:

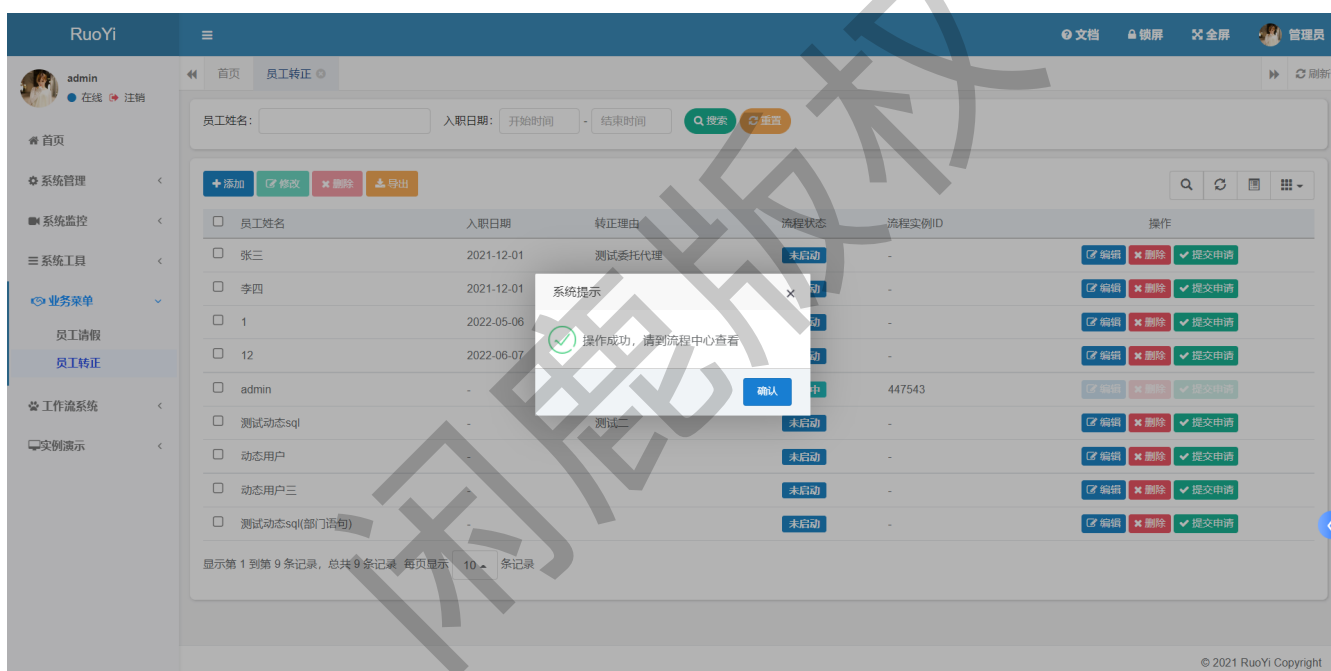
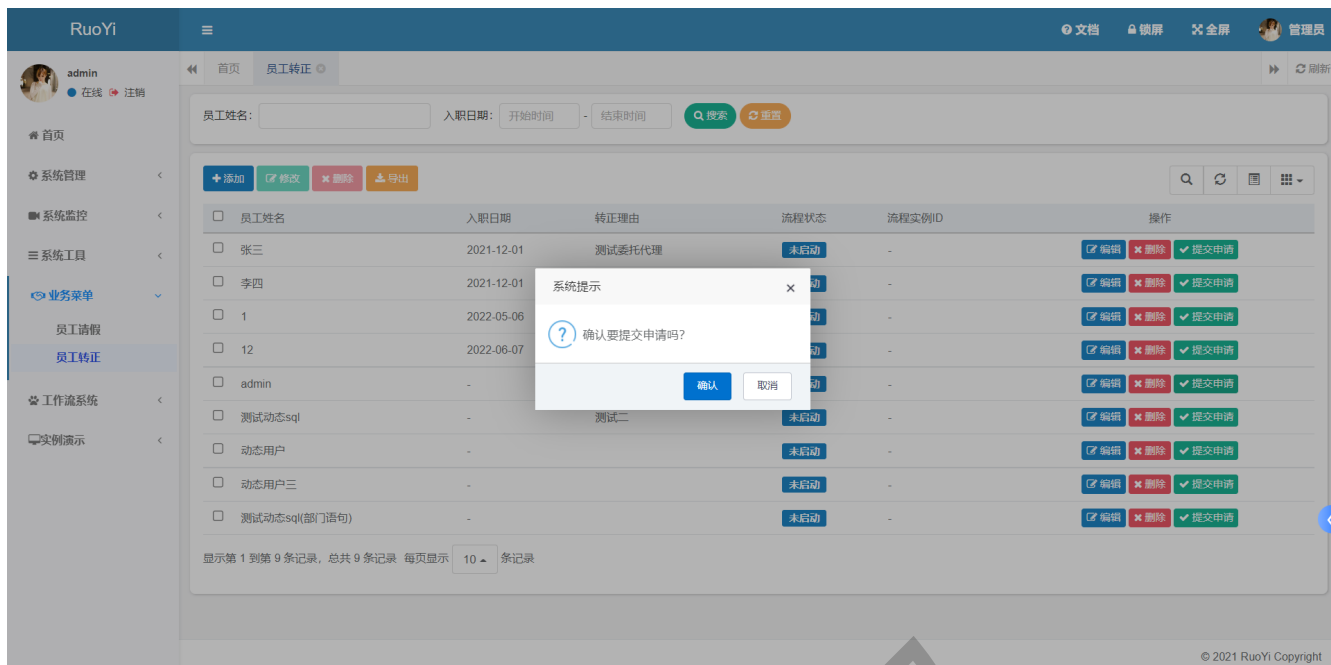


员工转正表单和流程，关联结果参考：



5、发起员工转正流程

点击员工转正列表中的**提交申请**按钮，即可提交流程。



提交成功后, 可以看到流程状态为进行中, 并且得到流程实例ID。流程详情可到[工作流系统 -> 流程中心](#)查看。

6、流程中心

流程中心聚合显示了[我的待办](#)、[我的草稿\(在线表单\)](#)、[我发起的](#)、[我参与的](#)等流程相关信息。



点击办理，可对该流程进行操作，如：同意、退回等。

员工转正流程(动态SQL用户)(动态SQL用户)

详细信息

步骤明细

流程监控图

员工姓名:

admin

入职日期:

yyyy-MM-dd

转正理由:

测试动态SQL用户

*审批意见

附件

上传文件

抄送用户

请输入用户ID，多个则逗号隔开（如 ry,admin）

同意

退回

关闭

五、在线表单集成流程

使用在线表单，需要额外获取表单设计器服务，没有的话联系作者。

1、启动表单设计器

5.1.1 安装 fis3

默认用户已经安装好 node 环境，没有的话自行查找资料安装。

执行以下命令全局安装 fis3 服务

```
npm install -g fis3
```

验证 fis3 已经安装

```
C:\Users\10001392>fis3 -v
```

```
[INFO] Currently running fis3 (D:\devSoft\nodejs\node_global\node_modules\fis3)
```

v3.4.46

—/\/\/\/\/\/\/\/\/—/\/\/\/\/\/\/—/\/\/\/\/—
 —\/\//\//\//\//—\//\//\//—/\/\//\//\//\//—
 —\/\//—\/\//—\/\//—\//—
 —\/\//\//\//—\/\//—\//\//—
 —\/\//\//\//—\/\//—\//\//—
 —\/\//—\/\//—\//\//—
 —\/\//—\/\//—/\/\——\/\//—
 —\/\//——/\/\/\/\/\/\/\—\//\//\//\//\//—
 —\//——\//\//\//\//—\//\//\//\//—

5.1.2 修改后端接口地址

找到表单设计器 public/request 目录下的 helper.js，修改实际的后端地址。（后端地址详见 ruoyi-admin/application.yml，默认端口为 80）

```
// 全局路径
const commonUrl = 'http://localhost:81'
```

5.1.3 指定端口启动表单设计器前端服务

执行以下命令，表单设计器即可运行于 8082 端口。

```

fis3 server start --www public --port 8082 --no-daemon

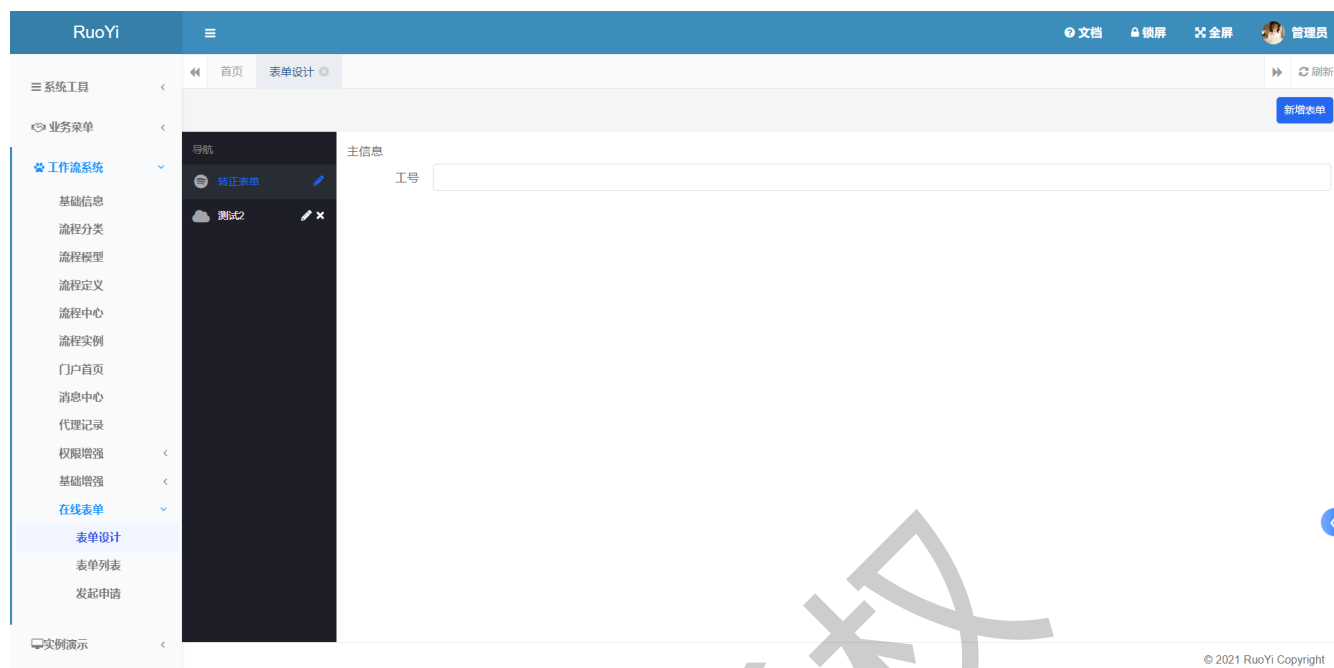
```

表单设计器域名地址，需要与 ruoyi-admin/application.yml 配置的一致。如：

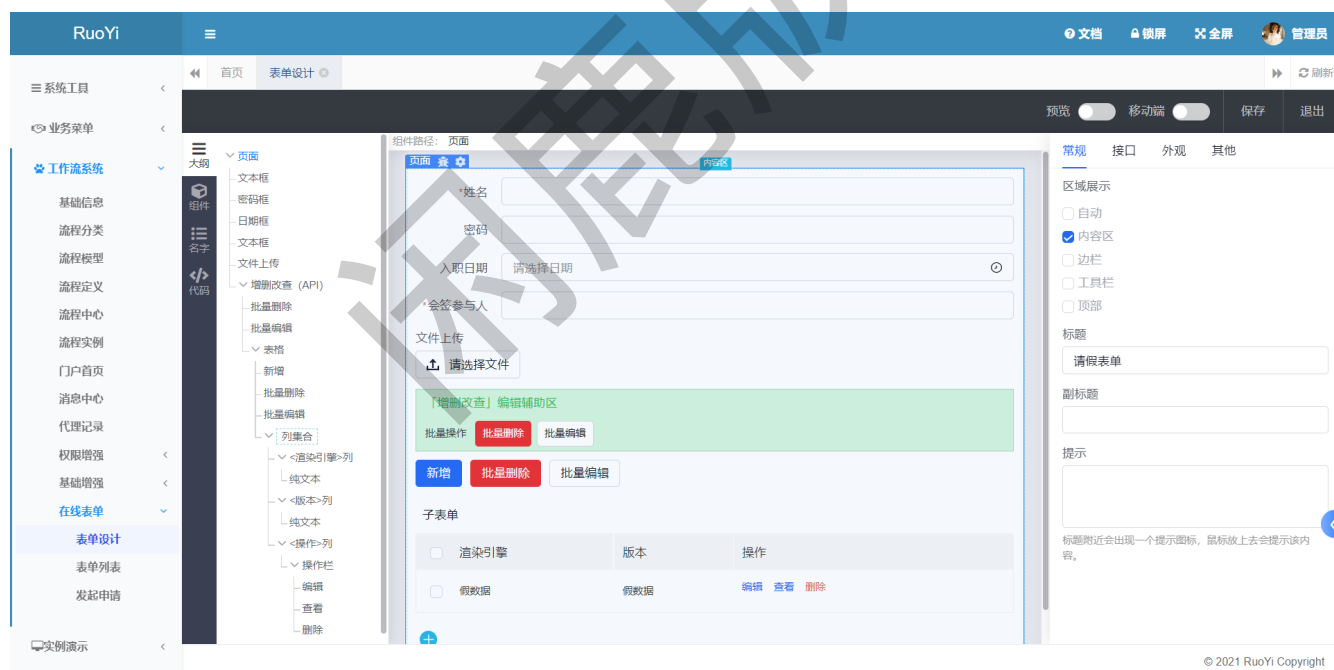
```
xianlu:
# 表单设计器前端域名
form-designer-origin: http://localhost:8082
```

2、表单设计

打开**工作流系统** -> **在线表单** -> **表单设计**。



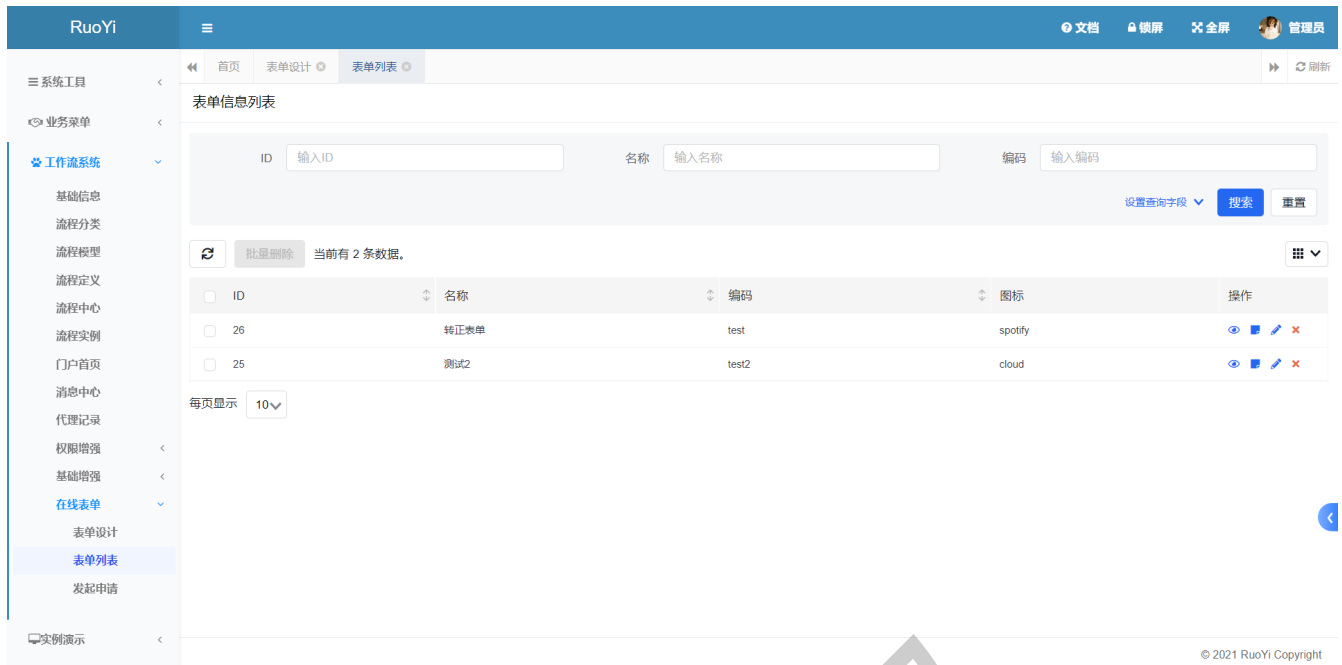
点击左侧小铅笔图标可进入表单编辑界面。表单设计好后，别忘了点击右上角保存按钮。



具体的表单设计细节，详见用户使用手册，此处不做过多描述。

3、表单列表

打开**工作流系统** -> **在线表单** -> **表单列表**，可查看历史表单列表。



4、流程设计与部署

与 4.3 业务表单集成流程/流程设计与部署 同理，不做赘述。

5、表单和流程关联

为实现在线表单和流程关联，需要在**工作流系统 -> 基础信息**添加以下一条记录。

5.5.1 业务表流程定义关系(在线表单)

新增

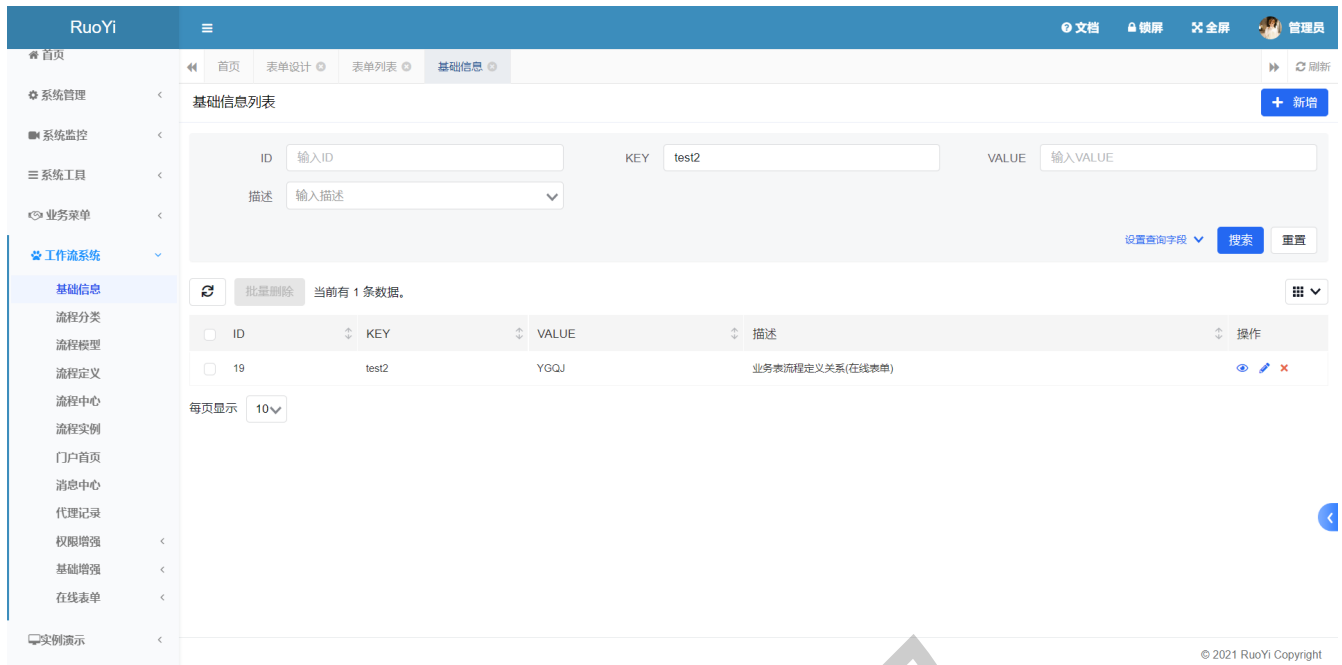
*描述	业务表流程定义关系(在线表单)
*KEY	这里填写表单设计生成的物理表名
*VALUE	这里填写流程定义KEY

取消

确认

在线表单，不需要配置业务表页面映射关系，系统会自动找到对应的页面。

在线表单和流程，关联结果参考：



6、发起在线表单流程

找到 **工作流系统** -> **在线表单** -> **发起申请** 点击员工请假会签版中的**发起**按钮，即可在新窗口打开流程表单界面。



*姓名

密码

入职日期

请选择日期

⊙

*会签参与人

文件上传

📁 请选择文件

新增

批量删除

批量编辑

子表单

渲染引擎	版本	操作
暂无数据		

暂存

提交

关闭

<

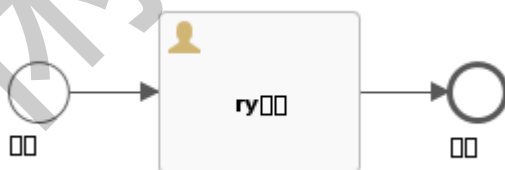
流程表单支持暂存操作，草稿位于 **流程中心 -> 我的草稿**，点击提交按钮即可提交流程。

7、流程中心

与 4.6 业务表单集成流程/流程中心 同理，不做赘述。

六、其他

1、Linux 部署项目，流程图文字乱码



乱码原因是因为 Linux 上没有安装中文字体**宋体**，手动安装即可。

找到附件宋体 `simsun.ttc`，或者 Windows 系统下，依次打开路径 `C:\Windows\Fonts`，找到**宋体 常规**。

将字体 `simsun.ttc` 通过 SFTP 等方式上传到 Linux 服务器上。

找到 jre 字体路径，如：`/usr/local/java/jdk1.8.0_171/jre/lib/fonts`，在 `fonts` 下新建 **fallback**目录，并放入宋体字体。

执行命令参考：

```
cd /usr/local/java/jdk1.8.0_171/jre/lib/fonts
mkdir fallback
cd fallback
mv ~/simsun.ttc ./

mkfontscale
mkfontdir

# 若找不到命令先进行安装相关工具:
yum install -y fontconfig mkfontscale
```

最后重启项目。

重启完项目，还需要进入流程模型，**重新部署到流程定义**，再预览新的流程图，即正常。



参考链接

[activiti流程图在linux上中文乱码问题解决](#)

2、Linux 部署项目，功能按钮、发起权限列表报错

查询列表报错：

```
{"msg":"查询失败: StatementCallback; bad SQL grammar [...]; nested exception is
java.sql.SQLException: Expression #2 of SELECT list is not in GROUP BY clause and
contains nonaggregated column 'a.NAME_' which is not functionally dependent on columns in
GROUP BY clause; this is incompatible with
sql_mode=only_full_group_by","data":null,"status":-1}
```

解决问题 暂时性修改sql_mode

set

```
@@GLOBAL.sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_
ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
```

缺点是下次重启mysql还会有这个问题

修改mysql配置文件 修改my.cnf文件来指定sql_mode的属性（my.cnf位于etc文件夹下），在最后一行追加下面的语句

sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

然后重启mysql

参考链接 ["this is incompatible with sql_mode=only_full_group_by" 问题解决](#)

(文档介绍完毕，如有疑问请联系作者：闲鹿软件开发工作室，微信(手机) 13123392160、QQ 1055202292)

闲鹿版权