# 基于swagger构建api文档

## 什么是 Swagger?

Swagger™的目标是为REST APIs 定义一个标准的，与语言无关的接口，使人和计算机在看不到源码或者看不到文档或者不能通过网络流量检测的情况下能发现和理解各种服务的功能。当服务通过Swagger定义，消费者就能与远程的服务互动通过少量的实现逻辑。类似于低级编程接口，Swagger去掉了调用服务时的很多猜测。

## 如何集成Swagger-springmvc到我们的项目中

依赖:maven

<!-- swagger-springmvc -->

<dependency>

<groupId>com.mangofactory</groupId>

<artifactId>swagger-springmvc</artifactId>

<version>1.0.2</version>

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.4.2</version>

</dependency>

<!-- swagger-mvc -->

使用：要最快捷地启动swagger-springmvc项目并且使用默认设置，推荐的方式是使用SwaggerSpringMvc插件

**Spring Java Configuration**



**Spring xml Configuration**

**在spring.xml配置**

<mvc:annotation-driven />

<bean class="com.mangofactory.swagger.configuration.SpringSwaggerConfig" />

<context:component-scan base-package=*"com.mangofactory.swagger"* >

<context:exclude-filter type=*"annotation"* expression=*"org.springframework.stereotype.Controller"*/>

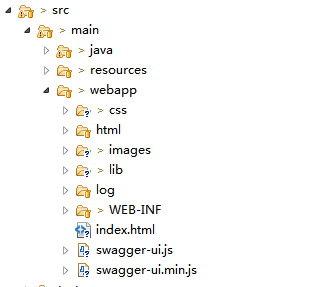
<context:exclude-filter type=*"annotation"* expression=*"org.springframework.web.bind.annotation.RestController"*/>

</context:component-scan>

**在dispather.xml配置**

*<context:component-scan base-package="com.mangofactory.swagger.controllers" />*

**将swagger-ui集成到项目中**

****

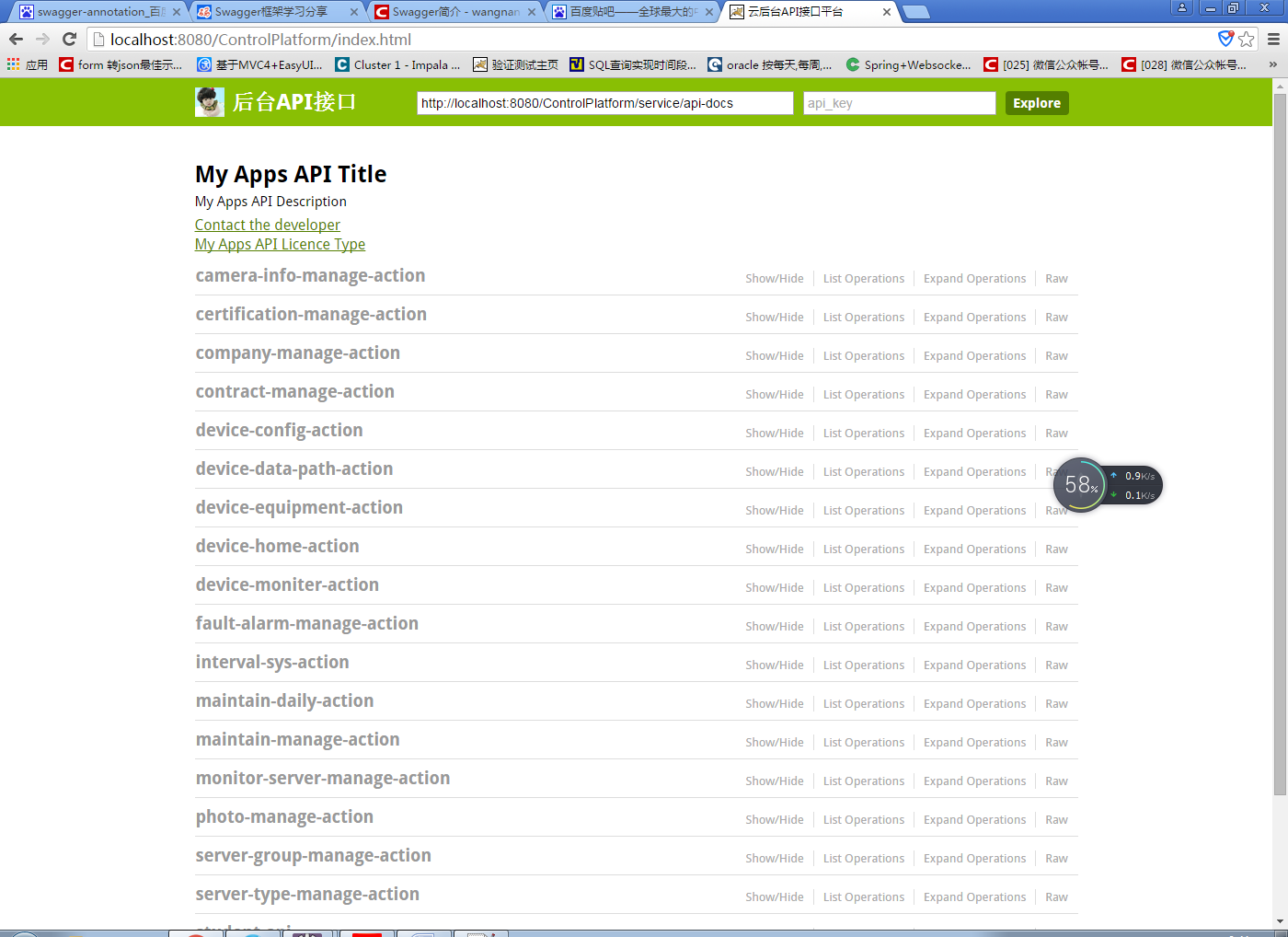
**修改index.html**

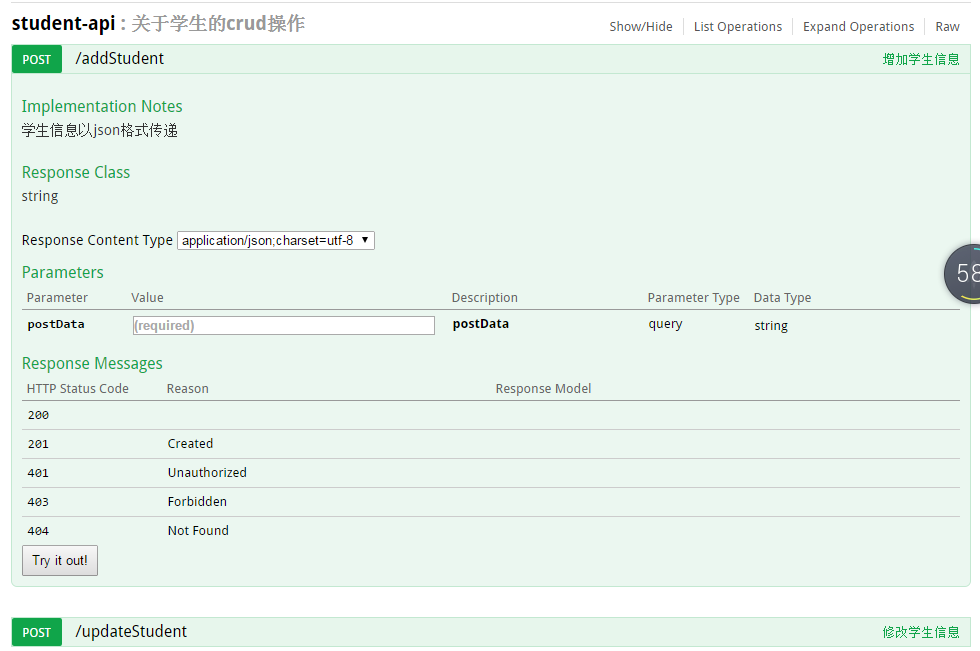
*将ur改成*

*url = "http://localhost:8080/ControlPlatform/service/api-docs";*

## 运行

<http://localhost:8080/ControlPlatform/index.html>





@Api(value="student api",description = "关于学生的crud操作",position =4 )

@Controller

**public** **class** StudentController {

**private** **static** **final** Log ***LOGGER*** = LogFactory.*getLog*(StudentController.**class**);

@RequestMapping(value = "updateStudent", method = RequestMethod.***POST***, produces = "application/json; charset=utf-8")

@ApiOperation(value = "修改学生信息", httpMethod = "POST", notes = "学生信息以json格式传递", response = String.**class**,position = 2)

**public** @ResponseBody String updateStudent(@ApiParam(required = **true**) @RequestParam String postData)

{

***LOGGER***.debug(String.*format*("enter function, %s", postData));

**return** "";

}

@RequestMapping(value = "addStudent", method = RequestMethod.***POST***, produces = "application/json; charset=utf-8")

@ApiOperation(value = "增加学生信息", httpMethod = "POST", notes = "学生信息以json格式传递", response = String.**class**,position = 1)

**public** @ResponseBody String addStudent(@ApiParam(required = **true**) @RequestParam String postData)

{

***LOGGER***.debug(String.*format*("enter function, %s", postData));

**return** "";

}

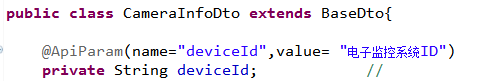
**注解说明:**

**@api** API表示一个开放的API，可以通过description简要描述该API的功能,value就是显示的名称

**@ApiOperation** @api可以有多个@ApiOperation，主要描述当前此方法的信息,value 显示此方法的名称，notes 此方法的描述，httpMethod此方法的请求方式,response 返回类型

**@ApiParam**

**如果我们的请求参数是bean,那么我们在bean里面配置如下**

****

****

**未解决的问题**

**点击try it out 不会调用**

**原因：**

<http://localhost:8080/ControlPlatform/addStudent?postData=%E6%88%91%E6%98%AF%E5%A5%BD%E5%AD%A6%E7%94%9F>缺少service这一层