### Http接口请求常用注解

1、GET请求

1、单一参数@RequestMapping(path = "/{id}", method = RequestMethod.GET)

1) public String getUser(@PathVariable String id ) {}

1. @RequestMapping(path="/{depid}/{userid}",method= equestMethod.GET) 可以同时指定多个提交方法

getUser(@PathVariable("depid")String departmentID,@PathVariable("userid") String userid)

3）一个顶俩

@GetMapping = @RequestMapping(method = RequestMethod.GET)

@PostMapping = @RequestMapping(method = RequestMethod.POST)

@PutMapping = @RequestMapping(method = RequestMethod.PUT)

@DeleteMapping = @RequestMapping(method = RequestMethod.DELETE)

4）@RequestParam(value = "name", required = true)

可以设置默认值，比如分页

4)@RequestBody 请求体映射实体类

需要指定http头为 content-type为application/json charset=utf-8

5）@RequestHeader 请求头，比如鉴权

@RequestHeader("access\_token") String accessToken

1. HttpServletRequest request自动注入获取参数

### 常用json框架介绍和Jackson返回结果处理

简介：介绍常用json框架和注解的使用，自定义返回json结构和格式

1、常用框架 阿里 fastjson,谷歌gson等

JavaBean序列化为Json，性能：Jackson > FastJson > Gson > Json-lib 同个结构

Jackson、FastJson、Gson类库各有优点，各有自己的专长

空间换时间，时间换空间

2、jackson处理相关自动

指定字段不返回：@JsonIgnore

指定日期格式：

@JsonFormat(pattern="yyyy-MM-dd hh:mm:ss",locale="zh",timezone="GMT+8")

空字段不返回：@JsonInclude(Include.NON\_NUll)

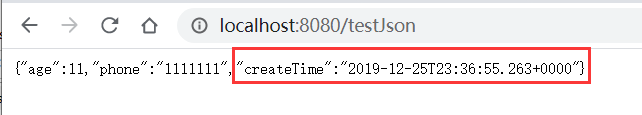
指定别名：@JsonProperty

举例：

|  |
| --- |
| **package** com.example.lchtest.springbootdemo1.domain;   **import** com.fasterxml.jackson.annotation.JsonIgnore; **import** lombok.Getter; **import** lombok.Setter;  **import** java.util.Date;  @Getter @Setter **public class** User {  **private int age**;  *// @JsonIgnore指定的字段不返回* @JsonIgnore  **private** String **pwd**;  **private** String **phone**;  **private** Date **createTime**;   **public** User(){  **super**();  }  **public** User(**int** age, String pwd, String phone, Date createTime) {  **this**.**age** = age;  **this**.**pwd** = pwd;  **this**.**phone** = phone;  **this**.**createTime** = createTime;  } } |
| **package** com.example.lchtest.springbootdemo1.controller;  **import** com.example.lchtest.springbootdemo1.domain.User; **import** org.springframework.web.bind.annotation.RequestMapping; **import** org.springframework.web.bind.annotation.RestController;  **import** java.util.Date; **import** java.util.Map;  @RestController **public class** SampleController {  @RequestMapping(**"/testJson"**)  **public** Object testJson(){  **return new** User(11,**"abc"**,**"1111111"**,**new** Date());  } } |

接口请求结果如下：

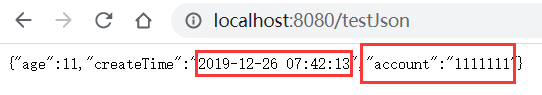
可以看到加了 @JsonIgnore注解的字段没有返回，但是时间的格式不是我想要的格式，



给phone指定一个序列化的别名，给**createTime指定返回的时间格式：**

|  |
| --- |
| @JsonProperty(**"account"**) **private** String **phone**; @JsonFormat(pattern = **"yyyy-MM-dd hh:mm:ss"**,timezone = **"GMT+8"**) **private** Date **createTime**; |

再次请求：



### SpringBoot2.x目录文件结构

##### 1.SpringBoot目录文件结构和官方推荐的目录规范

1、目录讲解

src/main/java：存放代码

src/main/resources

static: 存放静态文件，比如 css、js、image, （访问方式 http://localhost:8080/js/main.js）

templates:存放静态页面jsp,html,tpl

config:存放配置文件,application.properties

resources:

2、引入依赖 Thymeleaf

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

注意：如果不引人这个依赖包，html文件应该放在默认加载文件夹里面，

比如resources、static、public这个几个文件夹，才可以访问

3、同个文件的加载顺序,静态资源文件

Spring Boot 默认会挨个从

META/resources > resources > static > public 里面找是否存在相应的资源，如果有则直接返回。

Why? 这是因为源码中就是这么定义的，参考

<https://blog.csdn.net/qq_34797335/article/details/80194137>

4、默认配置

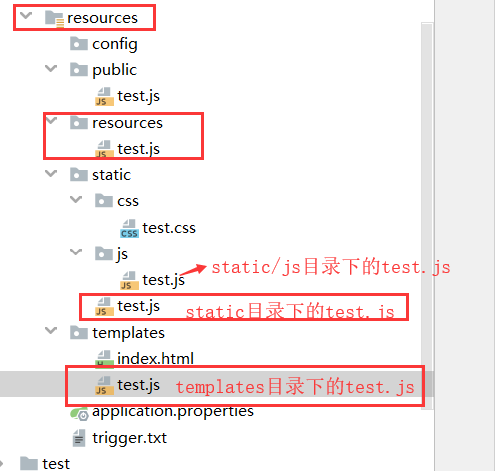
1)官网地址：

https://docs.spring.io/spring-boot/docs/current/reference/html/boot-features-developing-web-applications.html#boot-features-spring-mvc-static-content

2)spring.resources.static-locations= classpath:/META-INF/resources/,classpath:/resources/,classpath:/static/,classpath:/public/

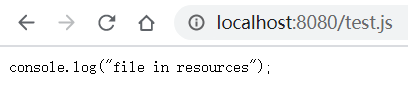
##### 2.资源文件加载目录顺序测试

在src/main/resources目录下创建resources、public目录，分别创建test.js文件，打印所属目录

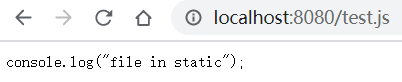


启动项目访问<http://localhost:8080/test.js> 进行测试：

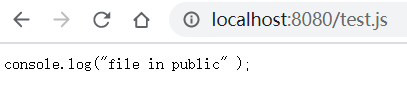
第一次访问：获取到的是src/main/resources/resources目录下的js文件，将 这个文件改名，



第二次访问：获取到的是src/main/resources/static目录下的js文件，将这个文件改名



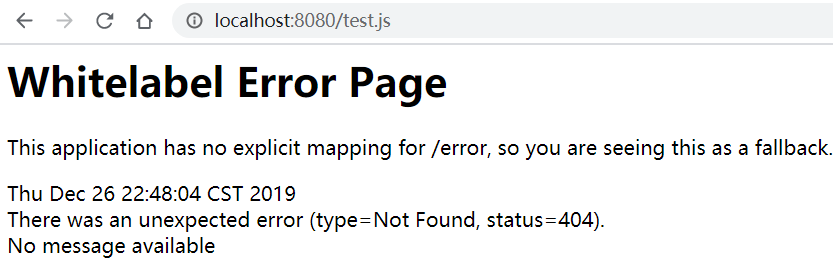
第三次访问：获取到的是src/main/resources/pbulic目录下的js文件



这说明了同名的资源文件，Spring Boot 加载的默认顺序是 resources > static > public

##### 3.访问自定义资源目录下的文件

在前述步骤，将public目录的test.js改名，重新访问：<http://localhost:8080/test.js>，结果如下，发现无法找到文件



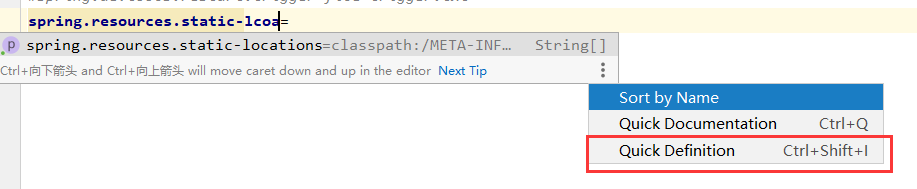
说明访问resources 、static 、public 目录之外的资源，需要手动去配置资源加载路径，在application.properties中添加如下配置，加上**classpath:/templates/**：

|  |
| --- |
| **spring.resources.static-locations**=**classpath:/META-INF/resources/, classpath:/resources/, classpath:/static/, classpath:/public/,classpath:/templates/** |

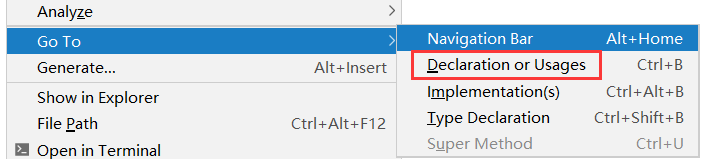
然后再去访问：



Tips：发现一个小技巧，输入配置项值时，停顿一些，会有一个提示，点击三个点，出现Quick Definition,然后在弹出框的界面上，会显示出这个配置项所在的类名，点进去可以查看反编译出来的源码：



或者直接鼠标右键，goto Declartion or usages，就可以打开所在的类ResourceProperties（这是一个配置类）

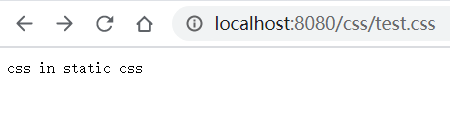


可以看到它里面定义了默认的资源加载路径：

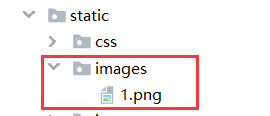
|  |
| --- |
| **private static final** String[] CLASSPATH\_RESOURCE\_LOCATIONS = **new** String[]{**"classpath:/META-INF/resources/"**, **"classpath:/resources/"**, **"classpath:/static/"**, **"classpath:/public/"**}; |

##### 4.访问css/图片

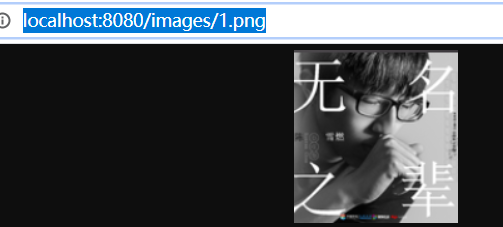
访问css文件： src/main/resources/目录下的resources 和static目录下的文件，访问时，不需要再带上resources static，默认从这两个目录下查找



访问静态图片：在static目录下创建一个images目录（名字任意），访问该目录下的图片：



<http://localhost:8080/images/1.png>



##### 5.访问html页面

如果直接访问<http://localhost:8080/index.html> ，则会报404的错误，这时在pom中 添加模板引擎thymeleaf依赖，刷新maven依赖，

|  |
| --- |
| <**dependency**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-thymeleaf</**artifactId**> </**dependency**> |

添加一个controller：

|  |
| --- |
| @RequestMapping(**"/index"**) **public** Object idnex(){  **return "index"**; } |

访问结果如下，也就是说，依赖了模板引擎，springboot会自动把idnex这个逻辑视图名映射到src/main/resources目录下的index.html中去

