

RESTful -Representational State Transfer

- 1 什么是RESTful特征的API
- 2 如何设计符合RESTful 特征的API
 - 2.1 关于URL
 - 2.2 HTTP动词语义
 - 2.2.1 说明
 - 3.3 查询参数
 - 3.4 HTTP状态码
- 3 常用注解
 - 1) @RequestMapping 注解
 - 2) 限定请求方式的注解
 - 2.1 @GetMapping 注解
 - 2.2 @PostMapping 注解
 - 2.3 @PutMapping 注解
 - 2.4 @DeleteMapping 注解
 - 3) @PathVariable 注解
 - 3.1 用法
 - 3.2 使用示例
- 4 JSON

RESTful -Representational State Transfer

1 什么是RESTful特征的API

- 每一个URI代表一种资源
- 客户端和服务端之间传递着资源的某种表现
- 客户端通过HTTP的几个动作 对 资源进行操作，发生 状态转化

2 如何设计符合RESTful 特征的API

2.1 关于URL

- 协议 - http/https
- 域名
域名中体现出api字样，如
<https://api.example.com>
<https://example.org/api/>
- 版本
<https://api.example.com/v1/>
- 路径
路径中避免使用动词，资源用名词表示，案例如下

```
https://api.example.com/v1/users
https://api.example.com/v1/animals
```

2.2 HTTP动词语义

2.2.1 说明

- HTTP动词语义

请求动词	说明
GET (SELECT)	从服务器取出资源（一项或多项）
POST (CREATE)	在服务器新建一个资源
PUT (UPDATE)	在服务器更新资源
DELETE (DELETE)	从服务器删除资源

具体案例如下：

请求动作	请求资源	说明
GET	/zoos	列出所有动物园
POST	/zoos	新建一个动物园
GET	/zoos/ID	获取某个指定动物园的信息
PUT	/zoos/ID	更新某个指定动物园的信息
DELETE	/zoos/ID	删除某个动物园
GET	/zoos/ID/animals	列出某个指定动物园的所有动物
DELETE	/zoos/ID/animals/ID	删除某个指定动物园的指定动物

3.3 查询参数

巧用查询参数

```
?type_id=1: 指定筛选条件
?limit=10: 指定返回记录的数量
?offset=10: 指定返回记录的开始位置。
```

3.4 HTTP状态码

用HTTP响应码表达 此次请求结果，例如

响应码	说明
200 OK - [GET]	服务器成功返回用户请求的数据
404 NOT FOUND - [*]	用户发出的请求针对的是不存在的记录
500 INTERNAL SERVER ERROR - [*]	服务器发生错误

3 常用注解

1) @RequestMapping 注解

`@RequestMapping` 用于指定处理请求的 URL，它可以标注在类和方法上；

可以通过 `method` 参数限定处理 GET、POST、PUT、DELETE 等HTTP的请求方法，比如：

- 处理 GET 请求

`@RequestMapping(value = "/v1/users", method = RequestMethod.GET)`

- 处理 POST 请求

`@RequestMapping(value = "/v1/users", method = RequestMethod.POST)`

处理 PUT DELETE 方式的HTTP请求同理。

2) 限定请求方式的注解

2.1 @GetMapping 注解

`@GetMapping` 只会处理 HTTP GET 请求；

是 `@RequestMapping(method = RequestMethod.GET)` 的缩写。

如果限定为处理GET请求，则发送其他方式请求时**HTTP状态码为 405**

2.2 @PostMapping 注解

`@PostMapping` 只会处理 HTTP POST 请求；

是 `@RequestMapping(method = RequestMethod.POST)` 的缩写。

如果限定为处理POST请求，则发送其他方式请求时**HTTP状态码为 405**

2.3 @PutMapping 注解

@PutMapping 只会处理 HTTP PUT 请求；

是 @RequestMapping(method = RequestMethod.PUT) 的缩写。

如果限定为处理PUT请求，则发送其他方式请求时**HTTP状态码为 405**

2.4 @DeleteMapping 注解

@DeleteMapping 只会处理 HTTP DELETE 请求；

是 @RequestMapping(method = RequestMethod.DELETE) 的缩写。

如果限定为处理DELETE请求，则发送其他方式请求时**HTTP状态码为 405**

3) @PathVariable 注解

@PathVariable 注解用于接受 RESTful API 中的 URL 中的变量；

通常与请求注解一起使用，可以将 URL 中的变量映射到 Controller 中的方法参数上。

3.1 用法

假如有一个 RESTful API: /v1/users/{id},

其中 {id} 是一个变量，表示用户 ID。

可以这样定义一个处理该请求的控制器方法

```
@GetMapping("/v1/users/{id}")
public String getUserById(@PathVariable Integer id) {
    // 根据 ID 查询用户，并返回用户信息
}
```

3.2 使用示例

根据 id 查询用户，并返回该用户信息

- 地址: /v1/users/{id} id为用户id
- 请求方法: GET
- 查询参数: 无
- 响应类型: 用户对象user

实现

使用_052MVC-BOOT02 工程

- 根据 id 查询用户，并返回该用户信息（GET请求: /v1/users/{id}）

```
@GetMapping("{id}")
public User selectById(@PathVariable int id){
    // 自己定义相对应的接口方法
    return userMapper.selectById(id);
}
```

4 JSON

JSON (JavaScript Object Notation) 是一种数据交换格式，常用于不同系统间的数据传输。

在使用SpringMVC框架时，常常需要将Java对象转换成JSON格式，然后返回给前端。

JSON格式的数据通常由一对花括号 {} 括起来，在花括号中包含键值对，键值对之间使用逗号分隔。

比如：

```
{
    "name": "Tom",
    "age": 18,
    "interests": ["reading", "music"],
    "address": {
        "city": "Beijing",
        "street": "xxx road"
    }
}
```