HTTP 接口设计

来自多个线上系统的实践

版本：1.0

肖念青 2020-01-13

内容

01 **概述**
背景和假设

02 **HTTP 协议**
针对 HTTP 协议的规则

03 **JSON**
如何构造请求和响应体

04 **Spring Boot 实现**
一个示例

05 **其他**
一些其他信息

概述

背景和假设

概述

- 本文档描述一种 HTTP API 的设计规范，此规范已经在多个线上项目中使用
- 前端包括PC Web，App，小程序，H5，IoT设备等
- 本文档后端基于 Java 语言
- 包含一个基于 Spring Boot 的参考实现，已开源：https://github.com/flmn/http-api-demo

HTTP 协议

针对 HTTP 协议的规则

HTTP 请求方法

- 所有请求使用 POST 方法
- 理由如下
  - 这份接口设计不是遵循 RESTful 规范的，所以不用纠结读请求用 GET，写请求用 POST，PUT，DELETE 等
  - 使用POST，相对于 GET 的 Query String，可以支持复杂的请求参数，即使是读请求，可以构造复杂的筛选请求结构
  - 便于对请求和响应统一做签名、加密、日志等处理

URL 规则

- URL 中的只能含有英文，使用英文单词或简称，不要使用汉语拼音。
- 所有字符使用小写字母。
- 多个单词间使用“-”（ hyphen，连字符）分隔，如create-user，不要使用 createuser、 createUser 或者 create_user 。
- URL 的 path 部分，使用 系统/模块/操作 的格式，如 app/account/login。
  - 系统，表示这个接口是给谁用的，比如：app、web、h5、weapp 等。命名可使用简称。
  - 模块，表示系统的子模块，比如：account（账户）、project（项目）、contract（合同）等。模块名字使用名词全称，且使用单数形式。
  - 操作，表示这个模块的具体的接口，比如 create-user（创建用户）、list-users（用户列表）等。使用动词+名词的形式，需要考虑单复数。对于常用的操作，可以使用习惯词语，比如 login（登录）、logout（登出）。

HTTP 头

- 将具体接口业务无关的数据放在HTTP Headers
- 后端系统可以在不涉及请求和响应体的情况下，处理一些公用逻辑
- 例如
  - X-Access-Token：身份认证
  - X-App-Os：客户端 OS
  - X-App-Version：客户端版本
  - X-Network：客户端网络模式，Wi-Fi，蜂窝网络
  - X-Sign：请求签名

请求和响应体

- 使用 UTF-8 编码
- JSON 格式
- 如果有加密功能，可以将正常的 JSON 加密后，使用 Base64 编码

HTTP 状态码

- 业务的处理结果不体现在 HTTP 状态码，由响应体的错误码字段表示
- 只使用部分 HTTP 状态码来表示一些业务无关的响应
- 例如
  - 200：业务已处理，但是业务处理成功还是失败由响应体表示
  - 400：错误的请求，多用在请求验证，客户端开发要保证向服务器提交正确格式的请求，400 错误不该在生产环境出现
  - 401：认证失败，一般是没有 Access Token 或者已过期
  - 403：无权限，指没有权限调用这个接口，客户端应该在界面上将用户无权限的操作隐藏
  - 500：服务器发生了未处理的异常， 500 错误不该在生产环境出现

JSON

如何构造请求和响应体

字段命名

- 由于 JSON (JavaScript Object Notation) 来自于 JavaScript 语言，所以字段命名遵循 JavaScript 语言，采用 lowerCamelCase (小骆驼拼写法)
- 不要采用 snake_case

数据类型

- 常用数据类型映射
  - bool：映射为 string，使用 Y 表示 true，N 表示 false
  - int：映射为 number
  - long：映射为 string
  - float，double，decimal：映射为 string
  - 日期、时间：映射为 string
- 特殊说明
  - 表示 ID 概念的字段，统一使用 string
  - long 映射为 string，是因为 JavaScript 的 number 能处理的数值范围不够，导致各种奇怪的问题

空值处理

- 在数据传输时，如果某个字段是空值，则直接省略此字段不传，减少网络开销

嵌套 vs. 平铺

- 嵌套 > 平铺
- 举例，Project 对象有个创建者，前端要显示创建者的名字和头像

Bad

```
{
  "id": "id",
  "name": "项目名称",
  "creatorId": "111",
  "creatorName": "创建者",
  "creatorAvatar":
"https://xxx.xxx.xxx/avatar.png"
}
```

Good

```
{
  "id": "id",
  "name": "项目名称",
  "creator": {
    "id": "111",
    "name": "创建者",
    "avatar":
"https://xxx.xxx.xxx/avatar.png"
  }
}
```

响应体

```
{
  "code": "OK",
  "message": "操作成功",
  "data": {},
  "errors": {
    "title": "不能为空"
  }
}
```

code

错误码，任何情况下必须返回

message

错误信息，可选

data

业务数据，可选，该字段是一个 Map

errors

错误表，发生 HTTP 400 错误时，存在此字段，该字段是一个 Map，Key 为发生错误的请求字段，Value 为错误描述

错误码

- code 字段表示业务处理的错误码
- 如果业务处理成功，必须返回 OK
- 如果业务处理失败，使用不同的错误码区分不同的错误，错误码使用简短的能够体现错误种类的英文单词来表示，使用大写字母，使用下划线分隔单词
- 为什么使用 string 而不是 int？传统上经常使用一个数字表示错误码，但是如果错误种类比较多，维护错误码表就是一个很大的工作，因为相近的错误码数字最好连续，但是后期添加就可能导致已经定义的错误码的改动。另外，数字错误码很难一看就知道发生了什么错误；使用字符串的话，能够从文字上看出错误类型，而且错误码的添加和更新也相对独立

Spring Boot 实现

一个示例

GitHub

- https://github.com/flmn/http-api-demo
- 此实现只包含接口处理部分，没有数据库访问，并不是一个完整的后端服务
- 代码中关于 Controller 位置的约定：
  - Controller 放在 api 包里
  - 每个接口放在自己单独的 Controller 里，并把接口的请求体定义放在同一个文件里
  - Java 包结构要反映 URL 结构，这样便于定位代码

一个🌰

```
package tech.jitao.httpapidemo.api.app.account;

@RestController(Login.PATH)

public class Login {

    static final String PATH = "/app/account/login";

    @Autowired

    private AccountService accountService;

    @PostMapping(PATH)

    public ApiResult process(@Validated @RequestBody Request request) {

        …

    }

    private static class Request {

        @NotBlank

        @Size(max = 16)

        private String username;

        @NotBlank

        @Size(max = 40)

        private String password;

        …

    }

}
```

其他

一些其他信息

未来的计划

- 签名？
- 加密？
- 你来说

对于本文档的意见和建议

微信

公众号

Thanks
谢谢

姬涛（肖念青）
jitao@jitao.tech
https://www.jitao.tech