

课程目标



工作日常



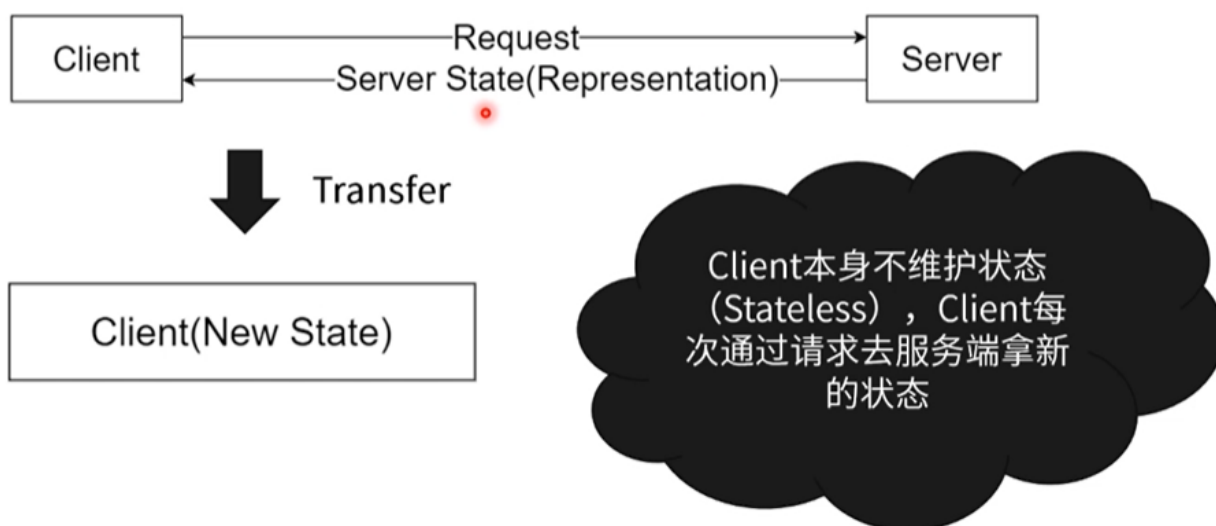
面试重点



大厂生态

RESTFUL

restful全称:表现层 状态 转化 Representational state transfer



在restful的约定里面 client端发送一个request到服务端，服务端的返回的数据，我们称之为表现Representation。因为在服务端是有一个业务状态的，比如订单状态，商品状态，但是服务端返回的是这些状态表现出来的样子。client端每次请求服务端的时候，客户端有自己的表现，也就是客户端的状态，也就是服务端状态的一种表示，请求回来后，客户端的状态会发生Transfer转化，客户端进入新的状态，浏览器会根据这个新的状态，重新渲染页面。客户端本身不维护状态,Client每次通过请求去服务端拿的状态。

资源 表示和转化

在restful协议中，有这样几个概念，资源 表示和转化。restful认为（资源-服务端拥有的是资源，拥有URL），客户端的拥有的不是资源本身，客户端在展现页面的时候，并不需要完整的数据，只需要数据的某个编号。（表示-服务端的资源在客户端的表示，客户端拥有操作服务端资源的方法），（转换-客户端收到新的表示，从而向新用户展示新的内容）

约定

资源命名: (1.名词性词组 比如.下单:/makeOrder ->restful会替换成POST/order 用户登录:/user/login ->restful会替换成/token,名词性的, 根据restFul协议。获取token用GET 一般认为使用GET参数都在URL不安全, 但是GET也可以body传参, 不一定要URL传参。)

状态 : (客户端无状态(资源表示-展示), 比如产品列表:GET/product ->产品列表
get/product 在服务端这是资源, 服务端并没有把资源传过来, 只是资源的表示, 就是能够直接渲染产品列表的部分数据, 最终展示出来的, 有可能是的客户端操作后的数据)

统一: restful希望客户端和服务端有统一的协议。比如: 资源ID 直接放在api的后面
/product/:id , 不使用query去表示。描述性强:遵守HTTP状态码, 规范的协议。

CURD约定

- GET /product?{query} get条件查询产品使用query
- GET /product/123 get查询某一个产品, 使用 api+ / + 资源ID
- POST /product <body> 创建某个产品, post +body传参
- PUT /product/123 <body> 修改某个产品, put api + / + 资源id body传参
- DELETE /product/123 删除某个产品 delete api + / + 资源id

Restful的优点

- 客户端无状态(简化开发, 状态都由服务端去管理, 客户端只看到资源的表示)
- 资源独立 接口间独立(服务端现在的架构是领域驱动, 符合现在潮流趋势。缓存好设计)
- 对协议依赖不严重(可迁移)

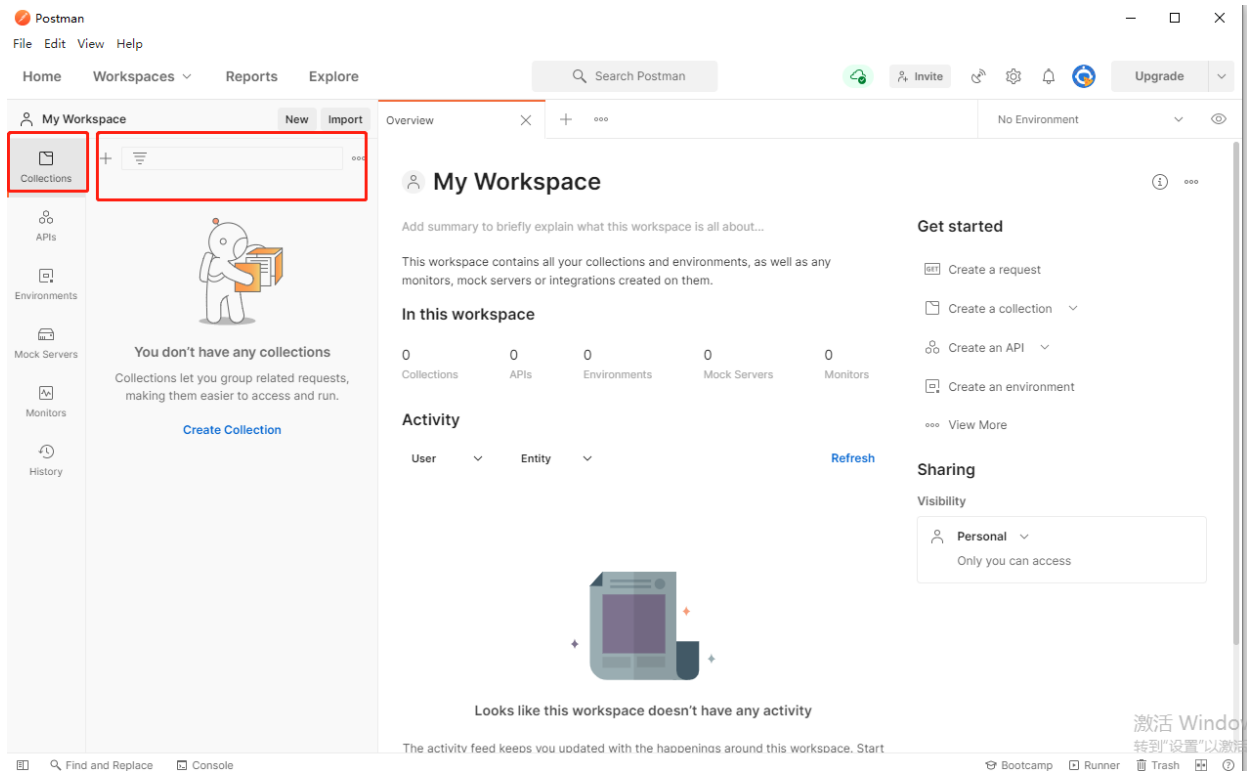
Restful的缺点

服务端建设不够领域化(导致:相同资源不同字段)

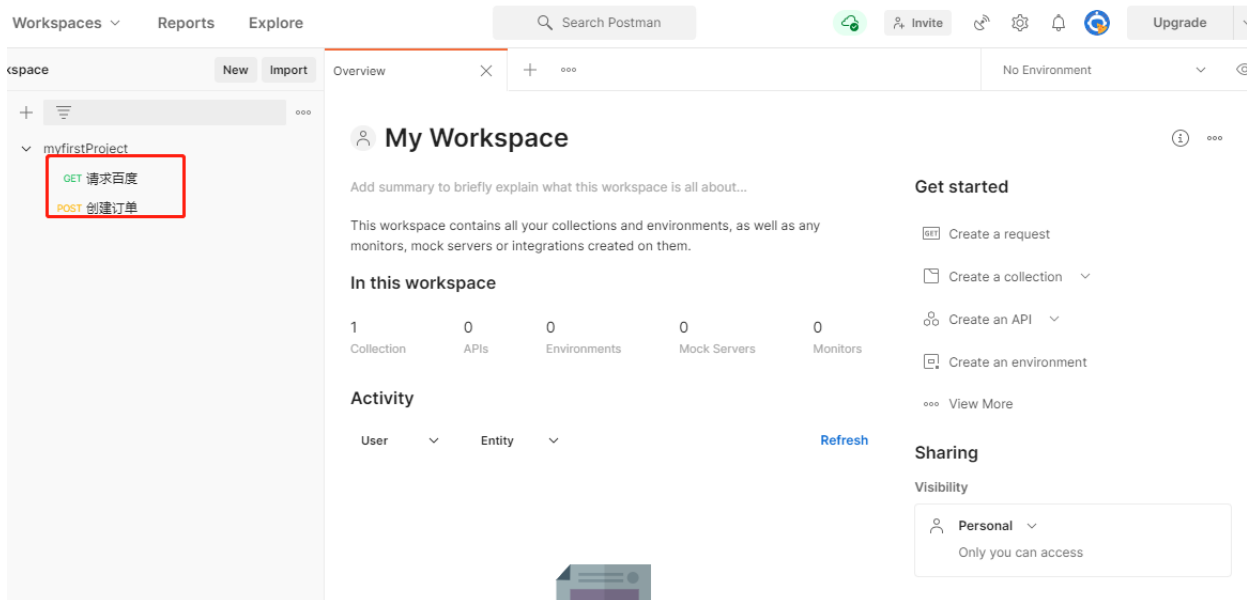
Postman使用方法

postman是一个api开发的协作工作

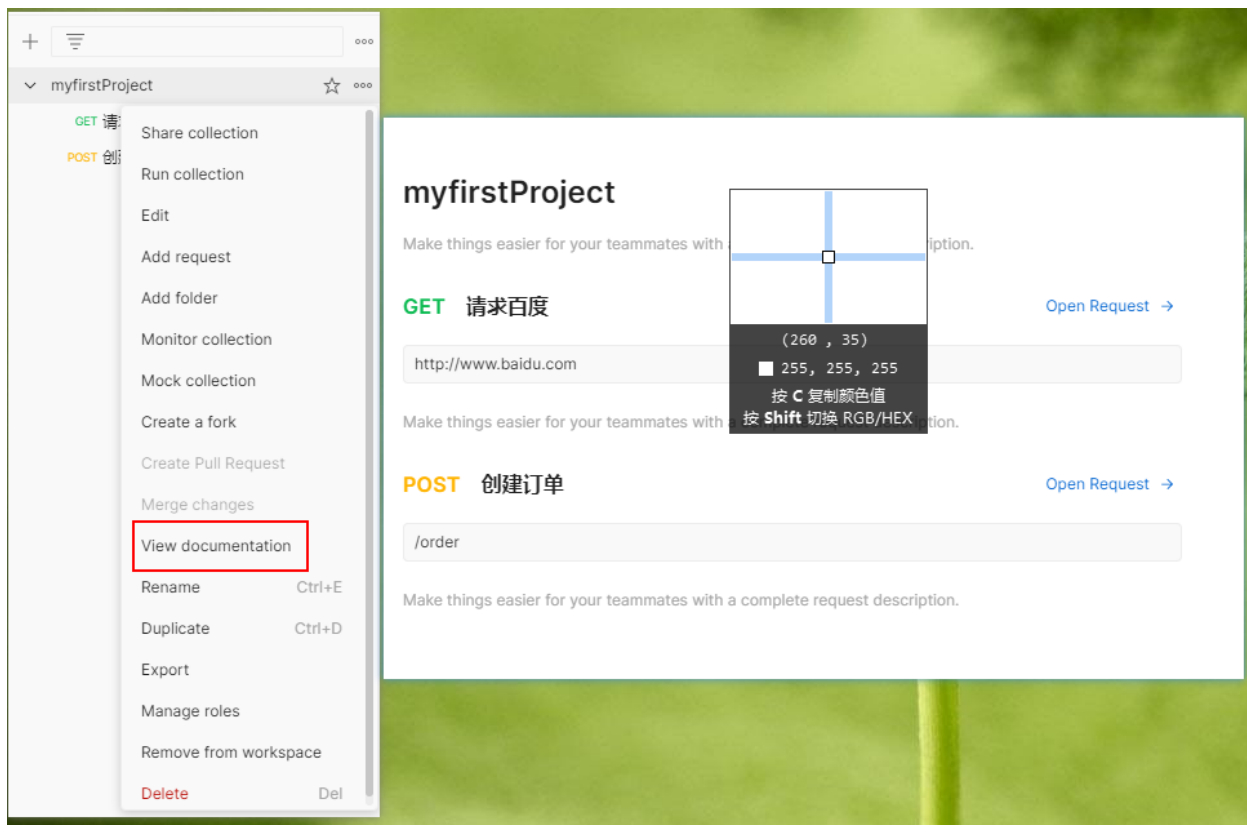
- 创建一个集合就如同创建一个项目



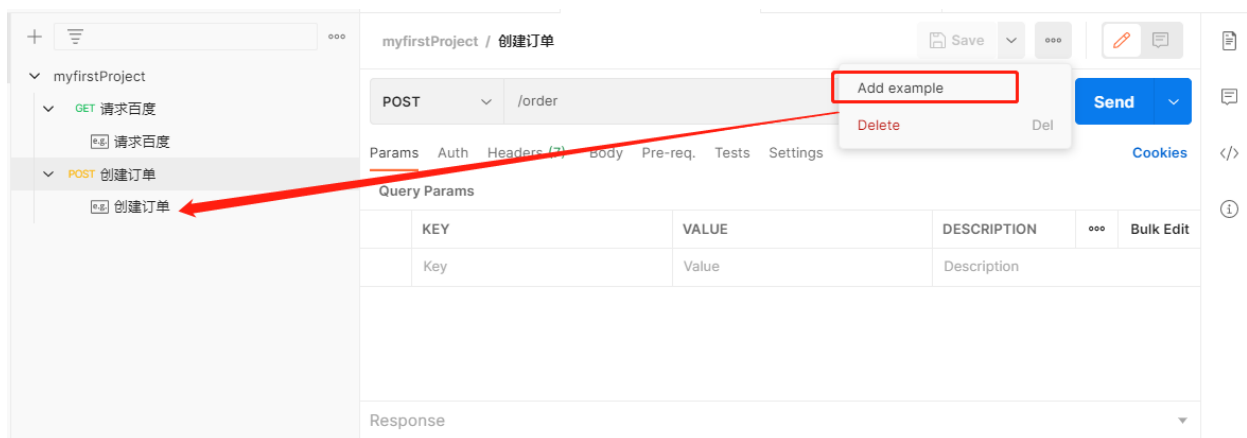
- 在集合内创建相关api



- 把创建的api生成文档



- 每个api添加例子



Overview 创建订单 + ... No Environment

myfirstProject / 创建订单 / 创建订单 Save 🗑 </>

POST ▼ /order ... i

Params Headers Body

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Headers

Status Code 201 Created

Pretty Raw Preview JSON ≡

```
1 {  
2   "success": 1  
3 }
```

- 创建MOCK服务

space new import Create mock server × T ... No Environment

myfirstProject ☆ ...

- > GET 请
- ▼ POST 创
 - Share collection
 - Run collection
 - Edit
 - Add request
 - Add folder
 - Monitor collection
 - Mock collection**
 - Create a fork
 - Create Pull Request
 - Merge changes
 - View documentation
 - Rename Ctrl+E
 - Duplicate Ctrl+D
 - Export
 - Manage roles

Create a mock server

☒ Select collection to mock 2. Configuration

Mock server name

Collection myfirstProject

Version CURRENT ▼ Choose a collection version for the mock server. If the collection is not linked to an environment, it can only be created on current version.

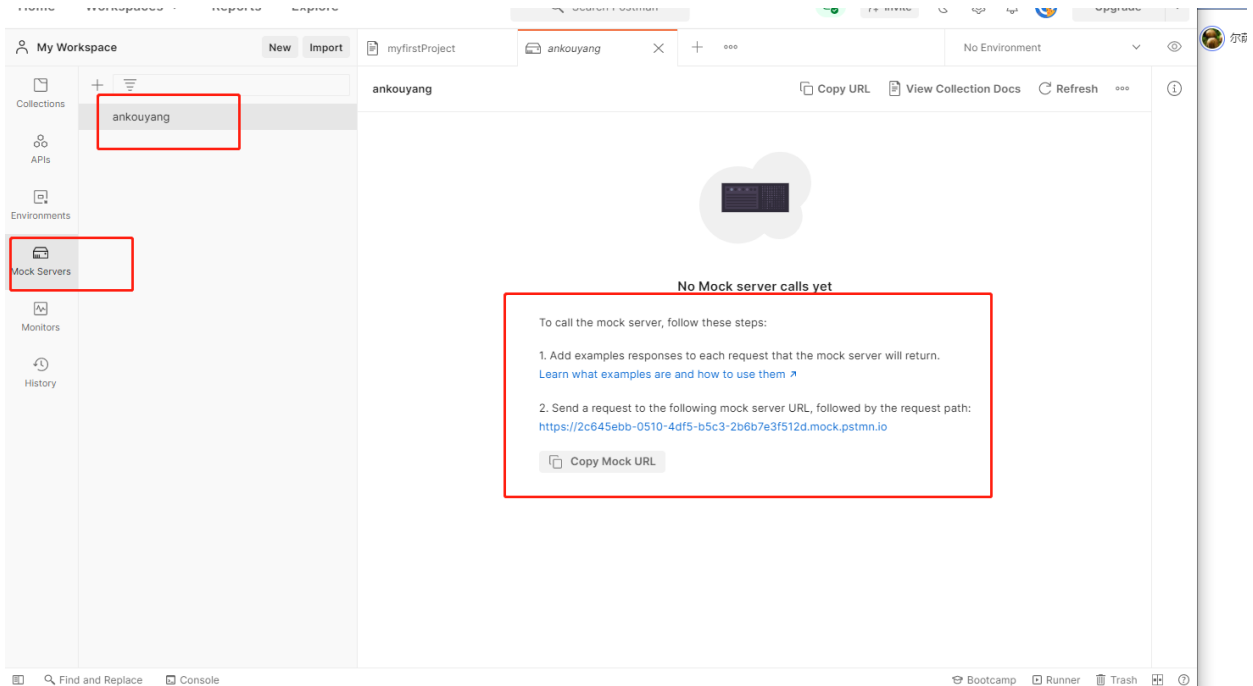
Environment No Environment ▼ An environment is a group of variables for storing and reusing values. [Learn more](#)

☐ Save the mock server URL as an environment variable Note: This will create a new environment containing the URL.

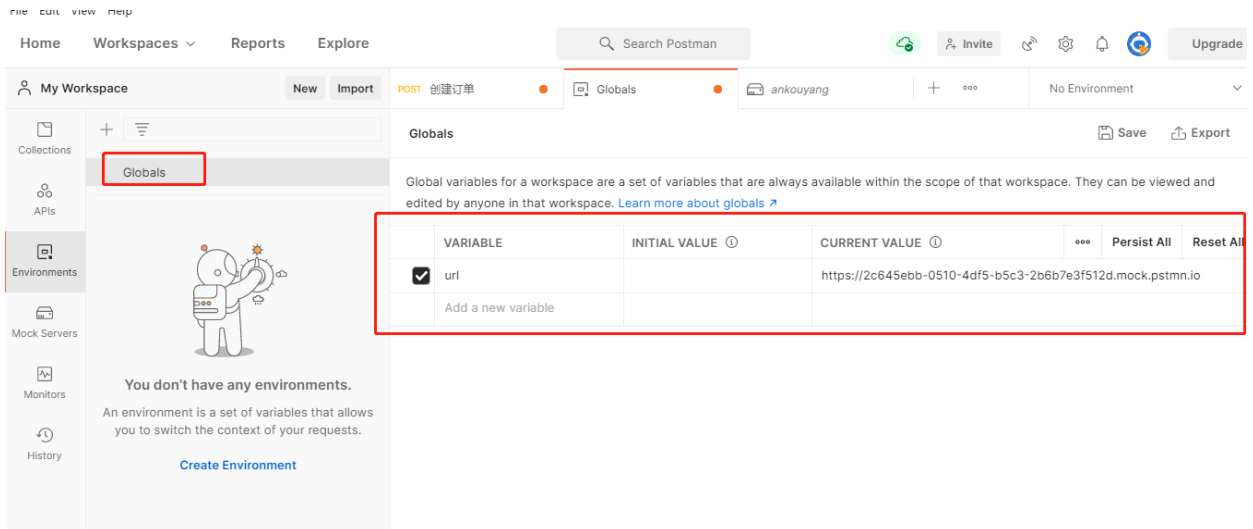
Close Create Mock S

Find and Replace Console Bootcamp Ru

- 查看已创建的MOCK服务

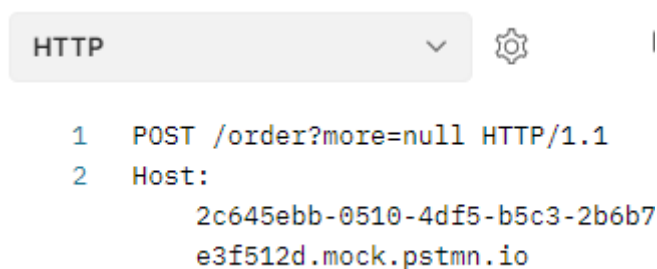


- 全局设置url并映射成MOCK的服务地址



- body传参不同的格式

1. none



2. form-data

```
1 POST /order?more=null HTTP/1.1
2 Host:
    2c645ebb-0510-4df5-b5c3-2b6b7
    e3f512d.mock.pstmn.io
3 Content-Length: 215
4 Content-Type: multipart/
    form-data;
    boundary=----WebKitFormBounda
    ry7MA4YWxkTrZu0gW
5
6 ----WebKitFormBoundary7MA4YWxkTrZ
    u0gW
7 Content-Disposition: form-data;
    name=""; filename="/D:/
    project/web知识体系/js进阶/函
    数/4、underscore源码分析.pdf"
8 Content-Type: application/pdf
9
.0 (data)
.1 ----WebKitFormBoundary7MA4YWxkTrZ
    u0gW
.2
```

3. x-www-form-urlencoded

```
1 POST /order?more=null HTTP/1.1
2 Host:
    2c645ebb-0510-4df5-b5c3-2b6b7
    e3f512d.mock.pstmn.io
```

4. raw(JSON)

```
1 POST /order?more=null HTTP/1.1
2 Host:
    2c645ebb-0510-4df5-b5c3-2b6b7
    e3f512d.mock.pstmn.io
3 Content-Type: application/json
4 Content-Length: 27
5
6 {
7
8   "$schema": 2222
9 }
```

5. binary

```
1 POST /order?more=null HTTP/1.1
2 Host:
    2c645ebb-0510-4df5-b5c3-2b6b7
    e3f512d.mock.pstmn.io
3 Content-Type: application/pdf
4 Content-Length: 22
5
6 "<file contents here>"
```

- 返回数据的body的MOCK数据,可以随机生成相关数据 如id productName img....

The screenshot shows a REST client interface with the following details:

- URL:** `POST /order ...`
- Params:** A table with columns KEY, VALUE, DESCRIPTION, and Bulk Edit. It contains one row with KEY: Key, VALUE: Value, and DESCRIPTION: Description.
- Body:** The response body is shown in JSON format (Pretty view). The status code is 201 Created. The JSON structure is:

```
1 {
2   "code": 200,
3   "success": 1,
4   "data": {
5     "id": "{{randomInt}}",
6     "name": "{{randomProductName}}",
7     "image": "{{randomImageUrl}}"
8   }
9 }
```

The "data" object is highlighted with a red box.
- Footer:** Includes icons for Bootcamp, Runner, Trash, and a settings icon.

- 通过Monitors查看的接口的请求情况
- 通过History查看相关的请求历史

课程小结

- 从RESTFUL角度：前端不是无状态，而是无业务状态
- Mock数据和接口测试的其他方法
 - Mock.js
 - Mocha/jest……