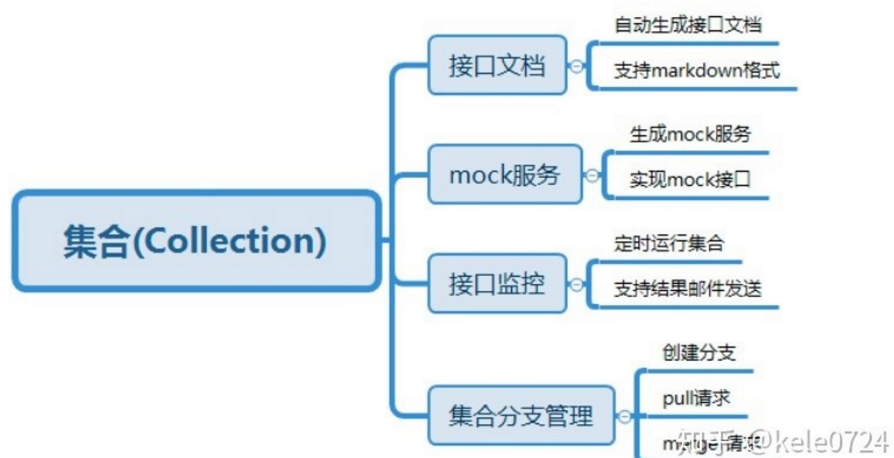


接口测试

2023年8月10日

16:07



接口测试:

请求的运行过程 (OK)

常见的接口关联 (OK)

集合管理-Collections (OK)

变量-环境变量、全局变量、集合变量 (OK)

断言-Tests (OK)

接口关联 (OK)

请求前置脚本-Pre-request Script (OK)

认证-Authorization(OK)

导入与导出(OK)

快速查找与替换(OK)

pm对象解析(OK)

Postman断言:

(1)、检验接口响应的状态码

```
pm.test("Status code is 200", function () {  
    pm.response.to.have.status(200)  
});
```

(2)、检查响应信息中是否包含某些指定的字符串

```
pm.test("Body matches string", function () {  
    pm.expect(pm.response.text()).to.include("水瓶");  
});
```

(3)、检查从JSON响应中获取到某个字段，判断其是否与预期字段一致

```
pm.test("断言type是否为今日运势:", function () {  
    var jsonData = pm.response.json();  
    pm.expect(jsonData.data.type).to.eql("今日运势");  
});
```

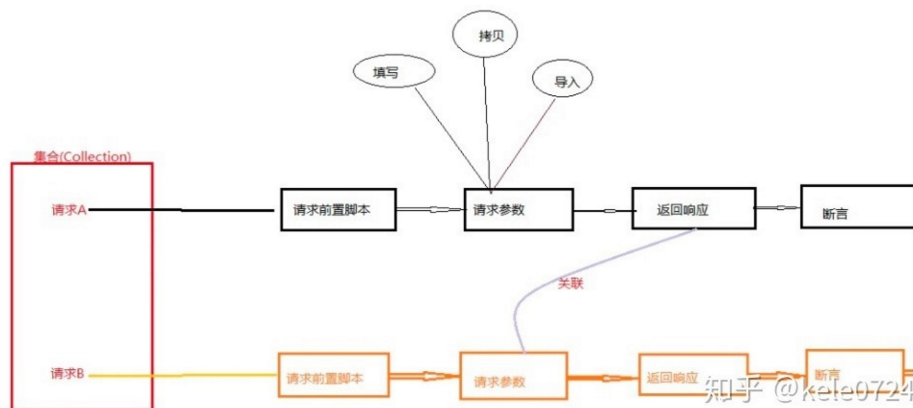
(4)、检查实际获取的响应体与预期结果的响应体是否一致

```
pm.test("Body is correct",function () {
```

```

    pm.response.to.have.body("response_body_string");
  })
  检查获取的头部与预期结果的头部是否一致
  pm.test("Body is correct",function () {
    pm.response.to.have.header("response_body_string");
  })
  (5)、响应时间
  pm.test("Response time is less than 200ms", function () {
    pm.expect(pm.response.responseTime).to.be.below(200);
  });

```



接口关联

```

// (1) 、获取返回的json数据
var jsonData = pm.response.json();
console.log(jsonData)
// (2) 、获取url的值
var type = jsonData.data.type;
// (3) 、将url保存为全局变量
pm.globals.set("type",type)

```

请求前置脚本

在pre-request Script里面编写对应的请求前置脚本



```

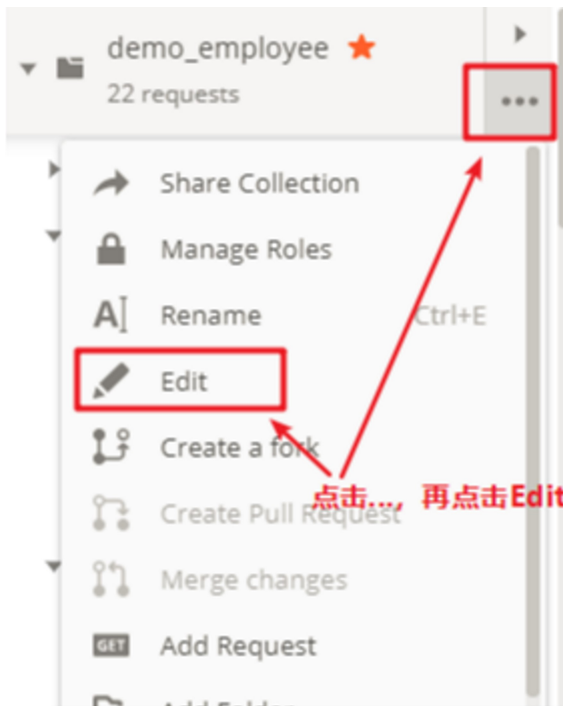
//1、获取到随机数
var random = Math.random();
console.log(random);
//2、将随机数保存到环境变量中
pm.environment.set("random_number",random);

```

认证

在Authorization中选择对应的信息

1、Inherit auth from parent从父级继承身份验证，在对一个集合进行测试时，在集合的根目录下就把token获取到的话，集合下的所有请求都会自动获取到token,无需任何处理



点击..., 再点击Edit

Name
demo_employee

Description **Authorization** Pre-request Scripts Tests Variables

This authorization method will be used for every request in this collection. You can override this by specifying one in the request.

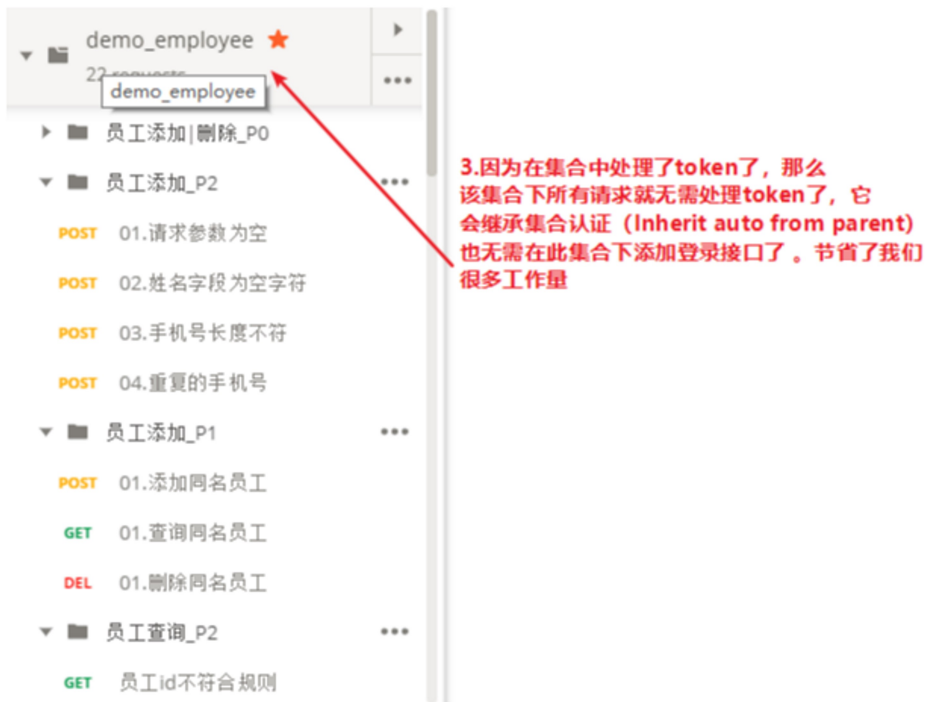
TYPE
Bearer Token

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. [Learn more about variables](#)

Token
{{token}}

2.根据项目情况而定, 我们所测的项目正好是这种情况。选择类型为Bearer Token。在右侧填写{{token}}, 获取全局变量中的token

Cancel Update



2、No Auth

无需身份认证

3、API Key

在请求头里面添加model:data xxx-xxx-xxx-xxx

4、通过Bearer Token认证，相当于在请求头中添加Authorization: Bearer Token

快速导入与导出

1、导出集合脚本

选择其中一个集合，点击右边的三个点 (...) ,选择'export'.

在弹出的对话框Collection v2.1(recommended),点击'Export' .

选择本地路径保存即可，默认导入的文件格式是json且文件名中含有Collection,这样能区分出是集合文件。

2、导出环境/全局变量

进入右上角的Environment quick look，选择所需的环境变量或全局变量，进入点击export即可导出，默认导出的文件格式是json文件名找那个含有environemnt或globals，可以区分出这是环境变量文件或全局变量文件。

快速查找与替换

在界面的左下角找到find and replace按钮，可以进行Regex正则匹配和Ignore Case忽略大小写，WHERE字段里面可以选择查询所有，或者从环境变量、全局变量、集合以及打开的请求里面查找,REPLACE WITH字段可以输入替换后的值，点击提交后即可替换

pm对象之sendRequest解析

通过pm.sendRequest在沙箱中发送请求

// 发送Get请求

```
pm.sendRequest("http://cx.shouji.360.cn/phonearea.php?number=13012345678", function(err,res) {
```

```

    if(err) {
        console.log(err);
    } else {
        pm.environment.set("xab","123");
    }
});
// 发送Post请求
const PostRequest = {
    url: 'https://petstore3.swagger.io/api/v3/pet',
    method: 'POST',
    header: 'Content-Type:application/json',
    body: {
        mode: 'raw',
        raw: JSON.stringify({
            "id": 10,
            "name": "doggie",
            "category": {
                "id": 1,
                "name": "Dogs"
            },
            "photoUrls": [
                "string"
            ],
            "tags": [
                {
                    "id": 0,
                    "name": "string"
                }
            ],
            "status": "available"
        })
    }
};
pm.sendRequest(PostRequest, function(err,res) {
    console.log(err ? err : res.json());
});

```

案例说明:项目token的获取通过sendRequest来实现，并在其它请求中使用

在Pre-request Script中编写发送登录接口的脚本，获得token后保存到集合变量中，在Authorization中设置获取到的token，使token在集合中有效，运行该集合，该集合下的其他接口不用再设置token了。

常用变量

pm.variables 局部变量，在请求中是使用

pm.environment 环境变量，在环境中都可以使用

pm.collectionVariables 集合变量，集合中的每个请求都可以使用该变量

pm.globals 全局变量，所有请求都可以使用

常见的方法:

pm.variables.has("var_name") 检查当前作用域内是否存在变量var_name，返回boolean类型

pm.variables.set("var_name","value"), 设置局部变量值

pm.variables.get("var_name"), 获取局部变量值

pm.variables.replaceIn("var_name"), 将动态变量{{var_name}}语法替换为其实际的解析值

pm.variables.toObject(), 返回本地范围内所有变量的对象

pm.response对象包含与收到的响应有关的所有信息

pm.response.code: 返回响应状态码

pm.response.status: 返回响应状态信息

pm.response.headers: 返回响应头

pm.response.responseTime: 返回响应时间

pm.response.responseSize(): 返回响应大小

pm.response.text(): 返回响应体的文本数据

pm.response.json(): 返回响应体的json数据