

# Postman断言

## 常用断言

### 最基本的返回验证

```
//1. 验证返回状态码是否是200
pm.test("Status code is 200", function () {
    pm.response.to.have.status(200);
});

tests["判断返回的状态为200"] = responseCode.code === 200;

//2. 验证返回body内是否含有某个值
pm.test("Body matches string", function () {
    pm.expect(pm.response.text()).to.include("string_you_want_to_search");
});

//3. 验证某个返回值是否是100
pm.test("Your test name", function () {
    var jsonData = pm.response.json();
    pm.expect(jsonData.value).to.eql(100);
});

//验证返回body中是否含有某个字符串
pm.test("Body is correct", function () {
```

## 断言小案例

### 接口返回的数据

```
{
  "status": 1,
  "res": [
    {
      "id": 39,
      "from": "东方",
      "to": "南方"
    },
    {
      "id": 38,
      "from": "西北",
      "to": "东南"
    }
  ]
}
```

```
//(1) 先获取到返回的json数据:
var responBody = JSON.parse(responseBody);

//2) 断言status返回的值为1
tests["判断返回的status返回为1"] = responseBody.status === 1;

//3) 断言res下第一个元素中from的值为东方
tests["res中第一个元素中from的值正确"] = responseBody.res[0].from === "东方";

//4. 判断数据返回类型是什么。我大概整理一下几种类型的: number 、 string 、 object 、 array 、 boolean 、 undefind。
tests["判断res下第一个元素中id的返回元素为number"] = typeof(responseBody.res[0].id) === "number";
//如果需要判断其他的类型就可以用同样的方法更换最后的类型就可以了。

//5. 判断返回数据中是否存在某个元素。这个虽然到现在一直没用得上, 但是也是一个基础的断言语句了还是上面的返回数据为例子, 判断返回元素中是否有status
tests["判断返回的元素中带有status"] = responseBody.has("status");
```

## 其他的操作

```
//1.将获取到的值设置为环境变量-->key变量名 value 环境变量值
pm.environment.set("key",value)
```

```
//2.设置流程控制跳转的下一条需要执行的接口-->requestname 为需要跳转的接口名
postman.setNextRequest(requestname)
```

```
//3.判断一个字段返回的值是否在一个列表中出现过;
//例子: 接口A中返回的列表中的name字段的值, 需要在B个列表中也出现且相等
//Blist是B接口返回的数组, Alist是A接口返回的数组, 通过遍历B数组查看是否有A数组中的name值, -1为没有的情况, 所以使用 >-1判断是否存在
for (var k=0; k<Blist.length; k++){
    tests["判断name是否相同"] = Alist.indexOf(Blist[k].name) >-1;
}
```

## 常见断言语法

---

```
//1.清除一个全局变量
postman.clearGlobalVariable("variable_key");
```

```
//2.清除一个环境变量
postman.clearEnvironmentVariable("variable_key");
```

```
//3.response包含内容
tests["Body matches string"] =responseBody.has("string_you_want_to_search");
```

```
//4.将xml格式的response转换成son格式
var jsonObject = xml2Json(responseBody);
```

```
//5.response等于预期内容
tests["Body is correct"] = responseBody === "response_body_string";
```

```
//6.json解析key的值进行校验
tests["Args key contains argument passed as url parameter"] = 'test' in responseJSON.args
```

```
//7.检查response的header信息是否有被测字段
tests["Content-Type is present"] = postman.getResponseHeader("Content-Type");
```

```
//8.响应时间判断
tests["Response time is less than 200ms"] = responseTime < 200;
```