

# 前端如何检查自己的请求是否写对

## 一、环境配置

- 1、安装 Python，版本越高越好。
- 2、打开 cmd，输入：pip install -i https://mirrors.aliyun.com/pypi/simple/ fastapi。
- 3、继续输入：pip install -i https://mirrors.aliyun.com/pypi/simple/ uvicorn。

## 二、打开 Uvicorn 服务器

- 1、把 test.py 放到某文件夹。
- 2、打开 cmd，cd 到这个文件夹。
- 3、输入：uvicorn test:app --reload。
- 4、出现 **INFO: Application startup complete.** 则开启成功。
- 5、打开浏览器，进入 <http://127.0.0.1:8000/docs>，能成功看到 FastAPI 文档。
- 6、不要关闭终端，可按 Ctrl+C 关闭 Uvicorn 服务器。

## 三、验证无参数 GET

向 <http://127.0.0.1:8000/hello> 进行 GET，会得到返回的包含一条 message 的 json，内容为"Hello"。

**Curl**

```
curl -X 'GET' \
'http://127.0.0.1:8000/hello' \
-H 'accept: application/json'
```

**Request URL**

```
http://127.0.0.1:8000/hello
```

**Server response**

Code	Details
200	<div><b>Response body</b></div> <pre>{   "message": "Hello" }</pre>

## 三、验证带参数 GET

向 <http://127.0.0.1:8000/a+b> 进行 GET，参数有 str 类型的 name，int 类型的 a 和 b，会得到返回的包含 name 和 result (a+b 的结果) 的 json。

### Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/a+b?name=Coming&a=2&b=3' \
  -H 'accept: application/json'
```

### Request URL

```
http://127.0.0.1:8000/a+b?name=Coming&a=2&b=3
```

### Server response

#### Code

#### Details

200

#### Response body

```
{
  "name": "Coming",
  "result": 5
}
```

### 四、验证带路径参数和参数的 POST

向 [http://127.0.0.1:8000/create\\_message/{message}](http://127.0.0.1:8000/create_message/{message}) 进行 POST，其中{message}为路径参数，是自己传的一个 str，参数有 str 类型的 creator 和 int 类型的 id。注意观察此处，路径参数和参数在 URL 上的体现方式有什么不同。返回值仍然是 json，包括了 message、creator 和 id。

### Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/create_message/Good%20bye%21?creator=Coming&id=666' \
  -H 'accept: application/json' \
  -d ''
```

### Request URL

```
http://127.0.0.1:8000/create_message/Good%20bye%21?creator=Coming&id=666
```

### Server response

#### Code

#### Details

200

#### Response body

```
{
  "message": "Good bye!",
  "creator": "Coming",
  "id": 666
}
```

### 五、验证带请求体的 POST

向 [http://127.0.0.1:8000/create\\_people](http://127.0.0.1:8000/create_people) 进行 POST，没有参数，只有一个请求体，请求体内包含了 name、sex、old 和 introduction。其中 sex 有默认值 Female，即使你没有 sex 这一项也可以。其中 introduction 是可选项，即使你没有 introduction 这一项也可以，会返回 null。我们会发现，请求体是不会体现在 URL 里的。

返回值是所有这些信息，以及一条 note。

#### Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/create_people' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "Coming",
    "sex": "Male",
    "old": 20,
    "introduction": "I\''m Coming."
  }'
```

#### Request URL

```
http://127.0.0.1:8000/create_people
```

#### Server response

##### Code

##### Details

200

##### Response body

```
{
  "name": "Coming",
  "sex": "Male",
  "old": 20,
  "introduction": "I'm Coming.",
  "note": "You send the packet successfully!"
}
```

## 六、写在最后

其实进入 <http://127.0.0.1:8000/docs>，就是一个自动生成的 FastAPI 文档，里面很详细地说明了有什么参数、有什么路径参数、有什么请求体，也可以自行点击“Try it out”来模拟发送请求。我们后端返回的都是一个响应体，请求体和响应体都是 json。如果你们看懂了，各种 GET 和 POST 都会用了，就会发现我前面那些都是废话，只要看文档就能完全看明白一个接口怎么用。