

flask.restful API

restful API规范

- **restful api**是用于再前端与后台进行通信的一套规范。使用这个规范可以让前后端开发变得更加轻松
- **协议:** 采用**http**或者**https**协议
- **数据传输格式:**数据之间传输的格式应该都使用**json**，而不使用**xml**
- **url链接:**
 - **url链接**中，不能有动词，只能有名词。并且对于一些名词，如果出现复数，那么应该再后面加**s**。在**restfulAPI**的设计中，**URL**代表了资源的集合和资源之间的关系。因此，**URL**应该是对资源的描述，而不是对操作的描述。使用名词作为**URL**的一部分，使**API**更加直观地表示资源和资源之间的关系，而不是过多地暴露实现细节和操作的方法
 - 例子，设计一个**restful API**来管理博客文章。根据“**restfulAPI**的名词化”原则，我们可以设计如下的**URL**结构：
 - a. 获取所有文章：GET /articles
 - b. 获取特定文章：GET /articles/{article_id}
 - c. 创建特定文章：POST /articles
 - d. 更新特定文章：PUT /articles/{article_id}
 - e. 删除特定文章：DELETE /articles/{article_id}
- **HTTP请求的方法:**
 - **GET:**从服务器上获取资源
 - **POST:** 在服务器上新创建一个资源
 - **PUT:** 在服务器上更新资源。(客户端提供所有改变后的数据)
 - **PATCH:** 在服务器上更新资源。(客户端只提供需要改变的属性)
 - **DELETE:** 从服务器上删除资源
- **常用状态码:**
 - **200** 服务器成功响应客户端的请求
 - **400** 用户发出的请求有错误，服务器没有进行新建或修改数据的操作
 - **404** 用户发送的请求的**url**不存在
 - **406** 用户请求不被服务器接受(比如服务器期望客户端发送某个字段，但是没有发送)

- 500 服务器内部错误，比如出现了bug

Flask-Restful插件的基本使用

pip3.7 install flask--restful即可安装

定义restful的视图

定义视图函数的时候，要继承自 `flask_restful.Resource`类，然后再根据当前请求的method来定义相应的方法。比如期望客户端是使用get方法发送过来的请求，那么就定义一个get方法。类似于MethodView.示例代码如下

```
from flask import Flask, render_template, url_for
from flask_restful import Api, Resource
app = Flask(__name__)
#用Api来绑定app
api = Api(app)
class IndexView(Resource)
    def get(self, username):
        return {"username": "xiaoxin"}
api.add_resource(IndexView, '/', '/login/<username>', endpoint =
'index')
```

注意：

`add_resource`的第二个参数是访问这个视图函数的url，这个url可以跟之前的route一样，可以传递参数。并且还有一点不同的是，这个方法可以传递多个url来指定这个视图函数。如果想返回json数据，那么就使用`flask_restful`，如果你想渲染模板，那么还是采用之前的方式，就是'`app.route`'的方式

对于一个视图函数，你可以指定好一些字段用于返回。以后可以使用ORM模型或者自定义的模型的时候，他会自动的获取模型中的相应的字段，生成json数据，然后再返回给客户端。这其中需要导入 `flask_restful.marshmallow`装饰器。并且需要写一个字典，来指示需要返回的字段，以及该字段的数据类型。示例代码如下：

```
+ from flask_restful import fields
+ class ProfileView(Resource):
+     定义好返回给前端的参数
+     resource_fields = {
+         'username': fields.String,
+         'age': fields.Integer,
+         'school': fields.String
```

```
+ }  
+ #restful规范中， 要求， 定义好了返回的参数  
+ #即使这个参数没有值， 也应该返回， 返回一个None回去  
+ @marshal_with(resource_fields)  
+ def get(self, use_id):  
+     user = User.query.get(user_id)  
+     return user
```