**1.介绍：**

**优势：**

Flask-Restful是一个专门用来写restful api的一个插件。

使用它可以快速的集成restful api接口功能。

在系统的纯api的后台中，这个插件可以帮助我们节省很多时间。

**缺点：**

如果在普通的网站中，这个插件就没有优势了，因为在普通的网站开发中，是需要去渲染HTML代码的，

而Flask-Restful在每个请求中都是返回json格式的数据。

**2.安装：**

|  |
| --- |
| pip install flask-restful |

**3.基本使用：**

定义Restful的类视图：

**1.** 从`flask\_restful`中导入`Api`，来创建一个`api`对象。

**2.** 写一个类视图，让他继承自`Resource`类，然后在这个里面，

使用你想要的请求方式来定义相应的方法，比如你想要将这个类视图只能采用`post`请求，那么就定义一个`post`方法。

**3.** 使用`api.add\_resource`来添加类视图与`url`。

|  |
| --- |
| **from** flask **import** Flask,url\_for,render\_template  **from** flask\_restful **import** Api,Resource  app = Flask(\_\_name\_\_)  api = Api(app)  **class** LoginView(Resource):  **def** post(self):  **return** {**"flag"**:**"yes"**}  **def** get(self):  **return** {**"flag"**:**"no"**}  *# api.add\_resource(LoginView,'/login/')*  api.add\_resource(LoginView,**'/login/'**,**'/login2/',**endpoint=**"login"**)  **with** app.test\_request\_context():  *# print(url\_for('loginview'))*  print(url\_for(**'login'**))  @app.route(**'/'**)  **def** hello\_world():  **return** render\_template(**'index.html'**)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

**4.注意：**

如果你想返回json数据，那么就使用flask\_restful，如果你是想渲染模版，那么还是采用之前的方式，就是`app.route`的方式。

url还是跟之前的一样，可以传递参数。也跟之前的不一样，可以指定多个url。

endpoint是用来给url\_for反转url的时候指定的。如果不写endpoint，那么将会使用视图的名字的**小写**来作为endpoint。