**Flask\_RESTful功能之参数验证：**

**1.参数验证：**也叫参数解析

Flask-Restful插件提供了类似WTForms来验证提交的数据是否合法的包，叫做reqparse。

**2.基本用法：(**借助于测试工程师 常用的接口测试工具postman来检验**)**

|  |
| --- |
| **from** flask **import** Flask,url\_for,render\_template  **from** flask\_restful **import** Api,Resource,reqparse,inputs  app = Flask(\_\_name\_\_)  api = Api(app)  **class** RegisterView(Resource):  **def** post(self):  #验证用户名  #1.创建解析器对象  parser = reqparse.RequestParser()  #2.利用解析器对象添加 需要验证的参数  parser.add\_argument(**'uname'**,type=str,help=**'用户名验证错误！'**,required=**True**,trim=**True**)  #3.利用解析器对象进行验证，若正确，直接返回验证后合格的参数值，若错误，抛异常信息给客户端  args = parser.parse\_args()  print(args)  **return** {**"tips"**:**"注册成功"**}  api.add\_resource(RegisterView,**'/register/'**)  @app.route(**'/'**)  **def** hello\_world():  **return** render\_template(**'index.html'**)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |

**3.其中add\_argument方法使用详解**

**add\_argument方法**可以指定这个字段的名字，这个字段的数据类型等，验证错误提示信息等，具体如下：

**default：**默认值，如果这个参数没有值，那么将使用这个参数指定的默认值。

**required：**是否必须。默认为False，如果设置为True，那么这个参数就必须提交上来。

**type：**这个参数的数据类型，如果指定，那么将使用指定的数据类型来强制转换提交上来的值。

**choices：**固定选项。提交上来的值只有满足这个选项中的值才符合验证通过，否则验证不通过。

**help：**错误信息。如果验证失败后，将会使用这个参数指定的值作为错误信息。

**trim：**是否要去掉前后的空格。

**其中的type，**

可以使用python自带的一些数据类型(如str或者int)，

也可以使用flask\_restful.inputs下的一些特定的数据类型来强制转换。

比如一些常用的：

url：会判断这个参数的值是否是一个url，如果不是，那么就会抛出异常。

regex：正则表达式。

date：将这个字符串转换为datetime.date数据类型。如果转换不成功，则会抛出一个异常。

**4.代码案例：**

|  |
| --- |
| **from** flask **import** Flask,url\_for,render\_template  **from** flask\_restful **import** Api,Resource,reqparse,inputs  app = Flask(\_\_name\_\_)  api = Api(app)  #Flask\_RESTful功能之参数验证  **class** RegisterView(Resource):  **def** post(self):  #用户名 密码 年龄 性别 出生日期 号码 个人主页  # 1.创建解析器对象  parser = reqparse.RequestParser()  #2.利用解析器对象添加 需要验证的参数  parser.add\_argument(**'uname'**,type=str,help=**'用户名验证错误！'**,required=**True**,trim=**True**)  parser.add\_argument(**'pwd'**, type=str, help=**'密码验证错误！'**,default=**"123456"**)  parser.add\_argument(**'age'**,type=int,help=**'年龄验证错误！'**)  parser.add\_argument(**'gender'**,type=str,choices=[**'男'**,**'女'**,**'双性'**])  parser.add\_argument(**'birthday'**,type=inputs.date,help=**'生日字段验证错误！'**)  parser.add\_argument(**'phone'**,type=inputs.regex(**r'1[3578]\d{9}'**))  parser.add\_argument(**'phomepage'**,type=inputs.url,help=**'个人中心链接验证错误！'**)  #3.利用解析器对象进行验证，若正确，直接返回验证后合格的参数值，若错误，抛异常信息给客户端  args = parser.parse\_args()  print(args)  **return** {**"tips"**:**"注册成功"**}  api.add\_resource(RegisterView,**'/register/'**)  @app.route(**'/'**)  **def** hello\_world():  **return** render\_template(**'index.html'**)  **if** \_\_name\_\_ == **'\_\_main\_\_'**:  app.run(debug=**True**) |