day15 问题管理

- 任伟博
- 毕云峰

今日概要

- 问题更新 + 操作记录
- 问题列表筛选
- 邀请成员

今日详细

1.知识点

1.1 反射

```
print( xxx_object.name )
print( getattr(xxx_object,"name") )

示例1:
    request.POST
    getattr(request,'POST')

示例2:
    row = models.User.objects.filter(id=1).first()
    row.name
    row.email
    getattr(row,'name')
```

```
xxx_object.name = "武沛齐"
setattr(xxx_object,"name","武沛齐")

示例1:
    row = models.User.objects.filter(id=1).first()
    row.email = "xxx@live.com"
    setattr(row,'email',"xxx@live.com")
    row.save()
```

需求：我通过ajax给你发送了一个数据 `{'v1':"email",'v2':"wupeiqi@qq.com"}` 或 `{'v1':"name",'v2':"alex"}` 或 `{'v1':"age",'v2':18}`，请你获取到这个字典后，对数据库中的用户表进行一次更新操作。

```python
def index(request):
    data_dict = json.loads(request.body.decode('utf-8'))

    user_object = models.User.objects.filter(id=1).first()
    setattr(user_object,data_dict["v1"],data_dict['v2'])
    user_object.save()

    return JsonResponse({"status":"成功"})
```

1.2 orm字段

需求：前端发送json `{'key':"email"}` 或 `{'key':"password"}`，后端接收到数据之后，去ORM类User中校验是否允许为空。

```
class UserInfo(models.Model):
    username = models.CharField(verbose_name='用户名', max_length=32,
db_index=True)
    email = models.EmailField(verbose_name='邮箱', max_length=32,null=True)
    mobile_phone = models.CharField(verbose_name='手机号', max_length=32)
    password = models.CharField(verbose_name='密码', max_length=32)

def index(request):
    data_dict = json.loads(request.body.decode('utf-8'))

    field_object = models.UserInfo._meta.get_field(data_dict["key"])
    print( field_object.verbose_name ) # 邮箱 ; 密码
    print( field_object.null ) # True ; False
```

1.3 可迭代对象

类中定义了 `__iter__`方法，且他返回一个迭代器。那么我们成根据类创建的对象，为可迭代对象。

是可迭代对象支持for循环。

```python
class Foo:
    pass

obj1 = Foo()
obj2 = Foo()

# 报错
for item in obj1:
    print(item)
```

```python
class Bar:
    def __iter__(self):
        yield 1
        yield 2
        yield 3

obj3 = Bar()
obj4 = Bar()

for item in obj3:
    print(item) # 1、2、3
```

示例:

```python
class Bar:
    def __iter__(self):
        yield 1
        yield 2
        yield 3

def index(request):
    obj = Bar()
    return render(request,'index.html',{'data_list':obj})
```

```html
<html>
    ...
    <ul>
        {% for data in data_list %}
            <li>{{data}}</li>
        {% endfor %}
    </ul>
</html>
```

2.问题的更新

2.1 给前端的标签绑定事件

2.2 触发事件发送ajax

```
{"name":"issues_type","value":"功能"}
{"name":"subject","value":"好饿呀阿123123"}
{"name":"status","value":6}
```

2.3 后台接收数据&更新

2.4 生成更新记录并返回

9：40 上课