

Day12

视图函数

- FBV
 - function base view
- CBV
 - class base view

I

类视图

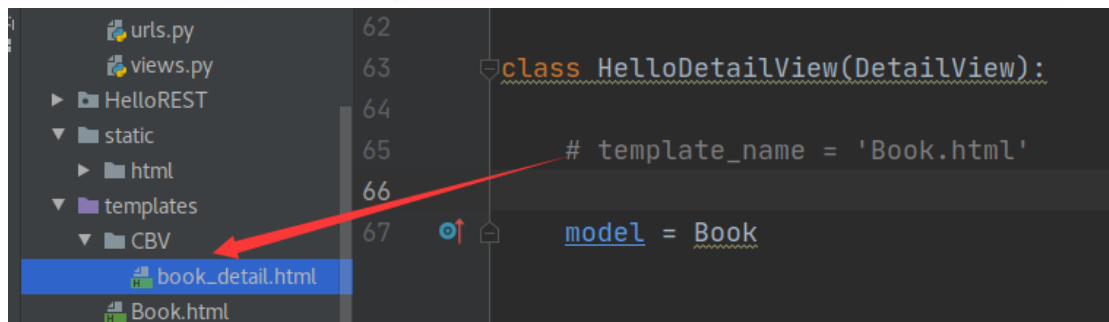
- CBV
- 继承自View
- 注册的时候使用的as_view()
- 入口
 - 不能使用请求方法的名字作为参数的名字
 - 只能接受已经存在的属性对应的参数
 - 定义了一个view
 - 创建了一个类视图对象
 - 保留，拷贝传递进来的属性和参数
 - 调用dispatch方法
 - 分发
 - 如果请求方法在我们的允许的列表中
 - 从自己这个对象中获取请求方法名字小写对应的属性，如果没有找到，会给一个默认http_method_not_allowed
 - 如果请求方法不在我们允许的列表中，直接就是http_method_not_allowed
 - 从自己这个对象中获取请求方法名字小写对应的属性，如果没有找到，会给一个默认http_method_not_allowed
 - 如果请求方法不在我们允许的列表中，直接就是http_method_not_allowed
 - 之后将参数传递，调用函数
- 默认实现了options
 - 获取接口信息，可以获取接口都允许什么请求
- 简化版流程
 - as_view
 - dispatch
 - 调用实现请求方法对应的函数名



类视图

- View
 - 核心
 - dispatch
- TemplateView
 - 多继承的子类
 - View
 - 分发
 - dispatch()
 - ContextMixin
 - 接收上下文
 - 从视图函数传递到模板的内容
 - 函数 `get_context_data`
 - TemplateResponseMixin
 - 将内容渲染到模板中
 - `template_name`
 - `template_engine`
 - `response_class`
 - `content_type`
 - 函数 `render_to_response`

I



```
62
63 class HelloDetailView(DetailView):
64
65     # template_name = 'Book.html'
66
67     model = Book
```

The file explorer on the left shows a project structure with files like `urls.py`, `views.py`, `static/html`, `templates/CBV/book_detail.html`, and `Book.html`. A red arrow points from `book_detail.html` to the `model = Book` line in the code.