



一、MVC 是什么？通俗版本

MVC 是一种软件架构模式，把一个网站分成三部分：

```
M = Model (模型 → 处理数据和逻辑)  
V = View (视图 → 页面、界面)  
C = Controller (控制器 → 负责调度，处理用户请求)
```

一句话解释：

Controller 负责接收请求 → 调用 Model 处理数据 → 再把结果交给 View 展示

这样代码清晰、分工明确。



二、为什么需要 MVC？

如果不使用 MVC，你的网站可能是这样的：

```
login.php 既写 HTML 页面，又写 SQL，又写逻辑  
register.php 也包含 HTML + SQL + 逻辑  
user_edit.php 同样...
```

所有代码挤在一起：

- 页面与业务混在一起
- 不好维护
- 代码越写越乱
- 改一个地方可能导致全站崩溃

MVC 就是为了解决这种“屎山”式代码。



三、MVC 的三部分分别做什么？

1 M (Model 模型)：处理数据和业务逻辑

Model 是所有 **数据、数据库操作、业务规则** 的地方。

它的职责：

- 查询数据库
- 插入、修改、删除数据
- 定义数据结构
- 验证数据格式
- 实现某些“业务逻辑”（例如积分计算、密码加密等）

Model 一般不写 HTML，也不显示给用户看。

❖ 在 Yii2 中 Model 的例子：

```
class User extends \yii\db\ActiveRecord
{
}
```

你可以像这样查询用户：

```
$user = User::findOne(1);
```

不用写 SQL!

2 V (View 视图) : 负责界面展示 (HTML/CSS/JS)

View 只负责一件事：

把数据变成页面展示给用户。

它只应该包含：

- HTML
- CSS
- JS
- 从 Controller 传来的变量

不应该包含：

- 复杂逻辑
- 数据库操作
- 业务计算

❖ Yii2 中 View 示例：

```
views/user/view.php
<h1>User: <?= $user->name ?></h1>
<p>Email: <?= $user->email ?></p>
```

很简单，只显示数据。

3 C (Controller 控制器) : 负责调度和流程

Controller 像“大脑”或“交通指挥员”，做这些事：

- 接收用户的 HTTP 请求
- 决定调用哪个 Model
- 将 Model 返回的数据传给 View
- 返回最终页面给浏览器

Controller 不显示页面，也不处理数据，只协调各部分。

◆ Yii2 中 Controller 示例：

```
class UserController extends Controller
{
    public function actionView($id)
    {
        $user = User::findOne($id);
        return $this->render('view', ['user' => $user]);
    }
}
```

这里：

- `$user = User::findOne($id);` → 调用 Model
- `render('view', [...])` → 调用 View 并传递数据

四、一个完整的 MVC 工作流程（举例说明）

用户访问：

```
http://example.com/user/view?id=1
```

MVC 的工作过程如下：

① Controller 接收请求

```
UserController 调用 actionView()
```

② Controller 调用 Model 获取数据

```
$user = User::findOne(1)
```

③ Model 去数据库查询，返回数据对象

```
User 对象 {id:1, name:"Tom", email:"xxx"}
```

④ Controller 把数据交给 View

```
render('view', ['user' => $user])
```

⑤ View 显示 HTML 页面

浏览器看到：

```
User: Tom  
Email: xxx
```



五、为什么 MVC 有用？（优势）

✓ 1. 前后端分离，易维护

- View 专注页面
- Model 专注逻辑
- Controller 负责协调

修改页面 → 改 View

修改数据库 → 改 Model

互不影响。

✓ 2. 多人协作更高效

- 后端开发写 Model + Controller
- 前端写 View

互不冲突。

✓ 3. 代码结构清晰，适合大项目

项目越大，结构越重要，MVC 在企业端广泛使用。

✓ 4. 安全性更高

Model 层统一处理数据库，避免重复写 SQL，减少漏洞。

✓ 5. 框架能自动生成大量代码（CRUD）

例如 Yii2 使用 Gii 可以自动生成：

- 控制器
- Model
- View (增删改查界面)

几分钟创建一个后台管理系统。

六、在 Yii2 框架中的 MVC 目录结构

一个 Yii2 项目的典型 MVC 结构：

```
controllers/  
    SiteController.php  
models/  
    User.php  
views/  
    site/  
        index.php
```

非常清晰。

七、总结：一句话记住 MVC

Controller 接收请求 → Model 处理数据 → View 显示页面

MVC 是 Web 开发最核心的思想。