**对于MVC模式的理解与阐述**

**一、MVC模式概述**

MVC模式（Model-View-Controller）是一种将软件系统分为三个基本部分的架构模式：

**模型层（Model）**：这个部分主要负责业务逻辑和数据处理，一般是从数据库拿取数据的部分。模型可以直接管理数据和它们的规则逻辑，并提供对数据的访问和存储方式，我认为可以算作是拿取数据的API集合。

**视图层（View）**：这个部分主要负责呈现界面给用户。视图可以将上一部分模型中的数据可视化展现出来，这一部分一般应该由模板语言所定义，这可以被看作是对数据展示方式的逻辑定义。

**控制层（Controller）**：这个部分主要负责处理用户事件，它主要是一系列的逻辑定义，当检测到某些用户事件后可以调用模型和视图完成用户请求的处理。此外我了解到从MVC的一般理解上看，模型层和视图层是直接联系的，所以控制层对模型层进行改变，这种改变将会在视图层被体现。但在实际应用中，似乎控制层更加被常用于连接模型层和视图层，而模型层和视图层之间并没有直接的联系。

**二、MVC工作原理**

**1. 用户交互：**首先，当用户与视图层进行交互的时候，比如说在廖雪峰MVC高级开发的实例里用户可能在Web界面上填写一个表单来登录（这是一个JSP页面），或者点击一个链接来请求特定的页面，这些都属于用户交互。

**2. 控制器处理：**然后，当控制器接收到用户输入后，它就负责解析用户的请求，并使用对应的方法来处理请求。比如点击按钮或提交表单，这个操作会发送一个HTTP请求到服务器。然后这个请求被交给对应的Servlet，这个Servlet就扮演了控制器的角色。例如，在实例里的UserController中，方法（signin，doSignin，signout）负责处理不同的HTTP请求。然后这些方法通过注解（@GetMapping或@PostMapping）与特定的URL路径关联起来。

**3. 模型处理与响应：**在这一个阶段模型会开始处理请求，比如数据库操作，然后返回结果给控制器。模型会查询数据库，然后对数据进行逻辑处理，并生成一个结果进行返回。比如说用户正在尝试登录，控制器将调用模型层的方法来验证用户的信息，然后返回状态（成功或失败）。

**4. 更新视图：**这一步控制器接收模型的数据，然后它将会选择一个相应的视图把模型处理的结果传递给它。视图层会使用这些数据结合那些相应的模板文件来生成HTML页面，然后发送给用户的浏览器。比如在例子中如果用户登录成功，控制器可能会选择一个显示用户信息的视图。如果登录失败，控制器可能会选择一个显示错误消息的视图展示给用户。