

Model

In Spring MVC, the `Model` interface is typically used to pass data from the controller to the view. It acts as a container for model attributes that are then rendered in the view.

You can use the `Model` interface to add attributes that you want to make available to the view. For example:

```
```java
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MyController {

 @GetMapping("/hello")
 public String hello(Model model) {
 model.addAttribute("message", "Hello, World!");
 return "hello"; // This would render the "hello.jsp" view
 }
}
```
```

In this example, the `hello` method adds an attribute named "message" to the `Model` object. This attribute can then be accessed in the view (e.g., a JSP or Thymeleaf template) to display dynamic content.

However, the `Model` interface itself does not provide methods to retrieve data. Its main purpose is to provide a way for controllers to pass data to the view layer. If you need to retrieve data in a controller method, you typically use other mechanisms such as `@RequestParam` or `@RequestBody` to extract data from the request.