

Now that you've completed the section, let's see if you can answer these questions. The answer sheet is available as a PDF in the "Support Files" folder.

Section 2: WebMVC – a First Glance

- 1) Which among these are the advantages of using **@RestController** when you are not returning HTML?
 - a. The controller is now restful
 - b. All method return values are considered to be a value object that is returned
 - c. No effect at all. For the developer, it's similar to using **@Service** or **@Component**
 - d. You no longer need to specify **@RequestBody** in the method declaration
 - e. You now have to provide hyperlinks for each result
- 2) When you have only **@RequestMapping** without further parameters, which request does this match to?
 - a. Any **GET** request
 - b. Any request for the root mapping
 - c. None; it needs to be specified
 - d. Any request that is not handled elsewhere
- 3) You don't want to provide a value for an **int** typed request parameter all the time and you want to have a default of 5, how can you solve this?
 - a. Add **default=5** to the **@RequestParam** annotation
 - b. You always have to provide a value
 - c. Switch to **Integer** and then check if it is null and apply the default in this case
 - d. Add **default="5"** to the **@RequestParam** annotation
- 4) Assume you have a request handler which has a value object as a parameter without any annotation. What happens when a request for it arrives?
 - a. Data from the request body and the parameters are applied
 - b. Data from the request path and the parameters are applied
 - c. Data from the request body and the request path are applied
 - d. Data from the request body, the request path, and the parameters are applied
 - e. Nothing happens
 - f. When a parameter is missing, the request fails
 - g. When a parameter is missing, the value is simply null
- 5) What happens when you write your test class with the MockMVC test helper?
 - a. The Spring context fully starts up
 - b. The embedded servlet container is started on a fixed port
 - c. The embedded servlet container is started on a free random port
 - d. Only the WebMVC controllers are started
 - e. You can import beans from your application's context in the test
 - f. You won't be able to import beans from your application's context in the test
 - g. You cannot use the MockBeans annotation; it would break the application