In this chapter, you will dive into the internals of Spring MVC, taking a close look at the
org.springframework.web.servlet.DispatcherServlet. You will begin by learning how an incoming
request is handled by the servlet, as well as how to identify which components play a role in the request
handling. After these components have been identified, you will go deeper into the roles and functions
of the different components and the different implementations of those components. You will also learn
how to configure the org.springframework.web.servlet.DispatcherServlet, in part by examining the
default configuration.

DispatcherServlet Request Processing Workflow

In the previous chapter, you learned about the important role the front controller plays in a Model 2
MVC pattern. The front controller takes care of dispatching incoming requests to the correct handler
and prepares the response, so that it can be rendered into something that the user would like to see. The
role of front controller in Spring MVC is played by the org.springframework.web.servlet
.DispatcherServlet. This servlet uses several components to fulfill its role. All these components are
expressed as interfaces, for which one or more implementations are available. The next section will
explore the general role these components play in the request processing workflow. Another upcoming
section will cover the different implementations of the interfaces.

Note We purposely used the term handler. The DispatcherServlet is very flexible and customizable, and it
can handle more types of handlers than just org.springframework.web.servlet.mvc.Controller
implementations or org.springframework.stereotype.Controller annotated classes.
The Workflow

A high-level overview of the request processing workflow is illustrated in Figure 4-1.

Figure 4-1. The request processing workflow

In the previous chapters, you learned about the importance of separation of concerns. Within the Spring Framework, the same rules have been applied. A lot of supporting components have been designed as interfaces with extensibility as well as separation of concerns in mind. Although the high level overview in Figure 4-1 is correct, there is more happening behind the scenes. Figure 4-2 shows a more complete view of the request processing workflow.