Spring Web Flow 2 (which we will call Web Flow from now on) is an important piece of the puzzle when it comes down to building scalable, easy-to-develop-and-maintain web applications. In the upcoming three chapters, we will make you familiar with Web Flow and its strengths and weaknesses; we will also cover why, when, and, above all, how you should use it. In this chapter, we start off explaining what Web Flow is and what it can do for you. Later on, we will discuss some basic Web Flow elements that you need to understand before you can start building your first flow.

The common thread throughout the Web Flow chapters is the sample application, which we create as we go through these chapters. We will start in this chapter by illustrating how you can turn a plain MVC sample with a bit of configuration into a Web Flow-enabled application. Next, we will gradually enhance the bookstore sample application with Web Flow functionality, explaining each feature in detail along the way. We took special care to ensure that you don’t need the sample application at hand when reading these chapters. All relevant code will be shown and explained accordingly.

In this chapter, we will limit ourselves to creating a single flow: the create (book) order flow. Chapters 11 and 12 will go more into detail about the application. In Chapter 11, we will expand the functionality of the sample application, and we will cover some more detailed configuration options. In Chapter 12, we will make you familiar with some more advanced aspects such as using AJAX, flow-managed persistence, and so forth.

If you already have some Web Flow experience, you might want to skim this chapter quickly and advance to the next chapter. If you are new to Web Flow, or want a refresher, then this is the right place to start!

Why Web Flow

In the Web Flow chapters, we will start by discussing the Web Flow 2.x family. We will always refer to version 1 of the tool as Web Flow 1. Otherwise, all features discussed relate to Web Flow 2.x (version 2.3.1 at the time of writing).

You may wonder why you should bother reading about Web Flow. You might especially be wondering why you should bother learning Yet Another Framework. Because let’s be honest: the Java landscape is wide, and there is tremendous amount of choice available in web frameworks (you have clearly opted for Spring MVC—good!). However, within those frameworks, there are still extras that you can add to make your life easier.

Easier, we say? Well, yes. We really mean this. By the end of these three chapters, we will have convinced you that using Web Flow can make management of your web applications easier. Let’s start with some theory first, so we can illustrate what Web Flow can do for you. Hopefully, this will get you interested enough to read through all three of the Web Flow chapters!
The Flow Concept

Despite the many features and advantages Web Flow has to offer, it all revolves around “the flow” (no wonder it is also called Web Flow). Before going any further, let’s back up and cover what a flow is and what it is modeled after. This will help a lot in understanding what Web Flow is trying to accomplish and how it will help you.

A flow is something that occurs naturally in most applications (see Figure 10-1). The basic example is a wizard. Imagine a web shop order wizard that requires you to follow several steps to place your order. These steps form a flow. You start with Step 1, where you fill in some identification data, such as your first and last name, and then click the Next button. This takes you to Step 2, which asks you where to ship your order. Step 3, the last step, allows you to enter delivery options: the type of shipment, whether you would like your package to be sent before a certain date (it’s possible there might be multiple shipments if the package wasn’t complete by that date), and so forth. If the wizard is any good, it will allow you to navigate back and forward between steps, remembering the state (data) you entered for that specific step in the flow. Each step is a small piece of the flow. They are linked together by a form of navigation. Sometimes you could also have the option to skip a step. At the end of the process, you can finalize your flow. In our example, this would process your order and make sure your products are delivered as soon as possible.

Web Flow is all about creating these flows, which you accomplish by managing state between the different flow steps, expressing navigation, and so forth. So far we’ve discussed the core aspects of a flow; however, there are also some other nifty features associated with flows. We will discuss these next.

Figure 10-1. A basic flow in an application