***Configuring a data source***

Regardless of which form of Spring-supported data access you use, you’ll likely need to configure a reference to a data source. Spring offers several options for configuring data-source beans in your Spring application, including these:

* Data sources that are defined by a JDBC driver
* Data sources that are looked up by JNDI
* Data sources that pool connections
* For production-ready applications, I recommend using a data source that draws its connections from a connection pool. When possible, I prefer to retrieve the pooled data source from an application server via JNDI. With that preference in mind, let’s start by looking at how to configure Spring to retrieve a data source from JNDI.

***Using JNDI data sources***

Spring applications are often deployed to run in a Java EE application server such as WebSphere or JBoss, or even a web container like Tomcat. These servers allow you to configure data sources to be retrieved via JNDI. The benefit of configuring data sources in this way is that they can be managed completely external to the application, allowing the application to ask for a data source when it’s ready to access the database.