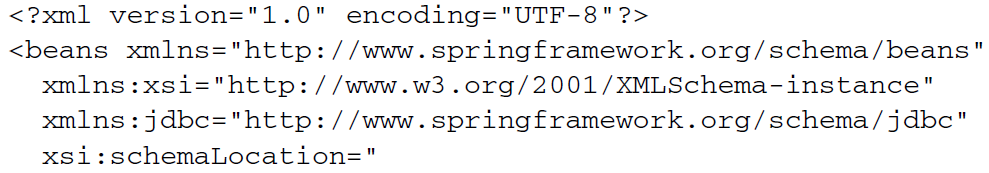
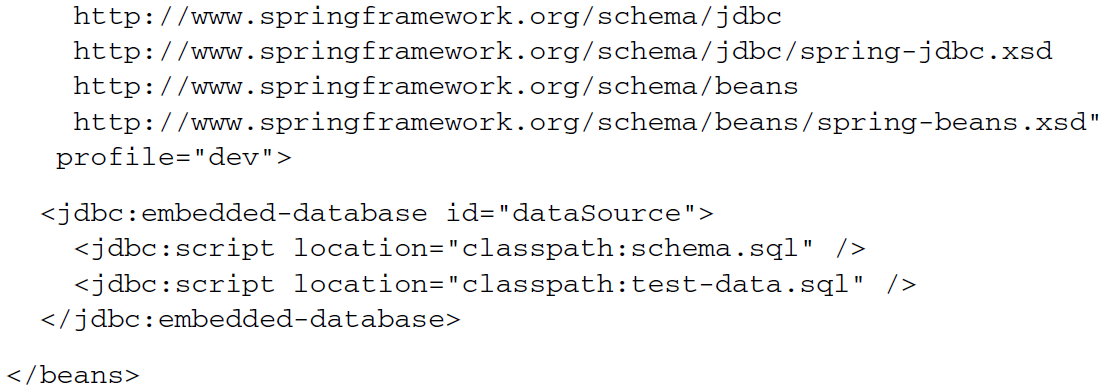
**CONFIGURING PROFILES IN XML**

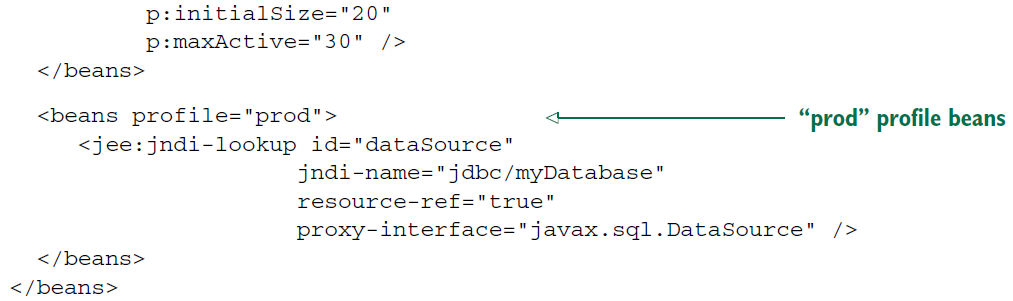
* We can also configure profiled beans in XML by setting the *profile* attribute of the *<bean>* element. For example, to define the embedded database *DataSource* bean for development in XML, you can create a configuration XML file that looks like this:

****

****

* Likewise, you could create another configuration file, with *profile* set to *prod* for the production-ready JNDI-obtained *DataSource* bean by the *qa* profile
* All the configuration XML files are collected into the deployment unit (likely a WAR file), but only those whose *profile* attribute matches the active profile will be used.
* Rather than creating a proliferation of XML files for each environment, you also have the option of defining *<beans>* elements embedded in the root <*beans>* element. This helps to collect all profiled bean definitions into a single XML file, as shown next:

****

****

* Aside from the fact that all these beans are now defined in the same XML file, the effect is the same as if they were defined in separate XML files.
* There are three beans, all of type *javax.sql.DataSource* and all with an ID of *dataSource.* But at runtime, only one bean will be created, depending on which profile is active.