syntx.co

Difference between loading context via DispatcherServlet and ContextLoaderListener

Share

I am writing this post to resolve any confusions about ways to set up Spring Application Contexts.

In Spring Web Applications, there are two types of container, each of which is configured and initialized differently. One is the "Application Context" and the other is the "Web Application Context". Lets first talk about the "Application Context".

Application Context is the container initialized by a ContextLoaderListener or ContextLoaderServlet defined in the web.xml and the configuration would look something like this:

```
<listener>
     <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<context-param>
     <param-name>contextConfigLocation</param-name>
     <param-value>classpath:*-context.xml</param-value>
</context-param>
```

In the above configuration, I am asking spring to load all files from the classpath that match *-context.xml and create an Application Context from it. This context might, for instance, contain components such as middle-tier transactional services, data access objects, or other objects that you might want to use (and re-use) across the application. There will be one application context per application.

The other context is the "WebApplicationContext" which is the *child* context of the application context. Each DispatcherServlet defined in a Spring web application will have an associated WebApplicationContext. The initialization of the WebApplicationContext happens like this:

```
<servlet>
      <servlet-name>platform-services</servlet-name>
      <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
      <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:platform-services-servlet.xml</param-value>
      </init-param>
      <load-on-startup>1</load-on-startup>
</servlet>
```

You provide the name of the spring configuration file as a servlet initialization parameter. What is important to remember here is that the name of the XML must be of the form <servlet name>-servlet. xml. In our example, the name of the servlet is **platform-services** therefore the name of our XML must be **platform-service-servlet.xml.**

Whatever beans are available in the ApplicationContext can be referred to from each WebApplicationContext. It is a best practice to keep a clear separation between middle-tier services such as business logic components and data access classes (that are typically defined in the ApplicationContext) and web- related components such as controllers and view resolvers (that are defined in the WebApplicationContext per Dispatcher Servlet).

java spring