

### Steps required for handling life cycle events of **ServletContext**

=====

1. Create a class which inherits **ServletContextListener** interface.
2. Override both the **abstract methods** in your class as per the requirement
3. Compile the class and place its bytecode in the **classes folder** of the application.
4. In **web.xml** create a tag called **<listener>** mentioning the name of the child class you have created.
5. Start the **Server** and automatically **Tomcat** will call **contextInitialized()** method.

### Passing attributes to ServletContext

=====

1. Many times we may want to pass some data from Listener class to Servlet.
2. For example: suppose our Listener class is opening DB connection and storing it in a `java.sql.Connection` object.
3. Now, every Servlet which wants to interact with the DB will require that Connection object (For creating Statement or PreparedStatement object).
4. In this case we have to set these objects as attributes inside the ServletContext object in our Listener class and in our Servlets we can get back these objects which are stored inside ServletContext.

5. This is done in 3 steps:

a) In the Listener class obtain the ServletContext object by calling the method `getServletContext()` of `ServletContextEvent` object available as argument in `contextInitialized()` method.

b) The ServletContext object has a method called `setAttribute()` having the following prototype:

```
public void setAttribute(String, Object);
```

The first argument represent the key name and the second argument is the object we want to share or pass from Listener to Servlets. For example: suppose we want to send current date from Listener to Servlet then our code will be:

```
java.util.Date d=new java.util.Date();  
ServletContext ctxt = e.getServletContext();  
ctxt.setAttribute("today", d);
```

c) Now, within Servlet's code we can call the method **`getAttribute()`** of **`ServletContext`** object to get back the **`Date`** object set by the **`Listener`** class. The method `getAttribute()` has following prototype:

```
public Object getAttribute(String);
```

The argument passed is the key name and the return value is the Object belonging to that key. In our case the code in Servlet will be:

```
ServletContext ctxt=super.getServletContext();  
java.util.Date d=(java.util.Date)ctxt.getAttribute("today");
```