The `@WebListener` annotation is used to register a class as a listener of a web application. The annotated class must implement one or more of the following interfaces:

-
  - `javax.servlet.ServletContextListener`
  - `javax.servlet.ServletContextAttributeListener`
  - `javax.servlet.ServletRequestListener`
  - `javax.servlet.ServletRequestAttributeListener`
  - `javax.servlet.http.HttpSessionListener`
  - `javax.servlet.http.HttpSessionAttributeListener`

Syntax

```
@WebListener([optional description])
```

Attributes

Name	Type	Required	Description
value	*String*	Optional	Description of the listener.

Examples

- The following example code uses the `@WebListener` annotation to register a class as a listener for the `ServletContextListener`'s events:

```
import javax.servlet.ServletContextEvent;
import javax.servlet.ServletContextListener;
import javax.servlet.annotation.WebListener;

@WebListener
public class ContextListener implements
ServletContextListener {

    @Override
    public void contextInitialized(ServletContextEvent
event) {
        System.out.println("The application started");
    }

    @Override
    public void contextDestroyed(ServletContextEvent
event) {
        System.out.println("The application stopped");
    }
}
```

-

- The following example code registers a listener which implements two interfaces with description:

```
import javax.servlet.http.HttpSessionAttributeListener;
import javax.servlet.http.HttpSessionListener;

@WebListener("Session listener for the application")
public class MySessionListener implements
HttpSessionListener,
        HttpSessionAttributeListener {
```

```
7
8       // overrides required methods here...
9   }
```