8253jang@daum.net
장희정

Listener 란?

Listener 는 웹 애플리케이션에서 발생하는 특정 이벤트(예: 객체 생성, 소멸, 속성 변경 등)를 감지하고, 이에 대응하는 동작을 수행할 수 있도록 해주는 컴포넌트이다. 주로 애플리케이션의 생명주기나 세션의 상태를 관리하거나, 특정 이벤트에 반응하는 로직을 구현하는 데 사용된다.

Listener 는 이벤트 기반 프로그래밍을 가능하게 하며, **특정 이벤트가 발생했을 때 자동으로 호출되는 메서드를 정의하여 기능을 작성한다.**

☞ Listener 의 주요 역할:

1. **애플리케이션 생명주기 관리** - **ServletContextListener**
   - 애플리케이션이 시작되거나 종료될 때 실행되는 코드를 정의할 수 있다.

2. **세션 생명주기 관리** - **HttpSessionListener**
   - 사용자의 세션이 생성되거나 소멸될 때 특정 작업을 수행할 수 있다.
   예를 들어, 세션 만료 시 사용자의 데이터를 정리하는 작업을 할 수 있다

3. **요청 및 응답 속성 관리** - **ServletRequestListener**
   - 요청, 세션, 애플리케이션 scope 에서 속성이 추가, 수정, 제거될 때 동작할 로직을 추가할 수 있다.

☞ 주요 Listener 인터페이스

Listener 는 다양한 인터페이스를 제공하며, 각각의 인터페이스는 특정한 이벤트를 감지하고 처리하는 데 사용된다

.

8253jang@daum.net
장희정

▷ ServletContextListener:

```
public class MyAppListener implements ServletContextListener {

    @Override

    public void contextInitialized(ServletContextEvent sce) {

        // 애플리케이션이 시작될 때 실행할 코드

    }

    @Override

    public void contextDestroyed(ServletContextEvent sce) {

        // 애플리케이션이 종료될 때 실행할 코드

    }

}
```

▷ HttpSessionListener:

```
public class MySessionListener implements HttpSessionListener {

    @Override

    public void sessionCreated(HttpSessionEvent se) {

        // 세션이 생성될 때 실행할 코드

    }

    @Override

    public void sessionDestroyed(HttpSessionEvent se) {
```

8253jang@daum.net
장희정

```
        // 세션이 소멸될 때 실행할 코드

    }

}
```

▷ServletRequestListener:

```
public class MyRequestListener implements ServletRequestListener {

    @Override

    public void requestInitialized(ServletRequestEvent sre) {

        // 요청이 초기화될 때 실행할 코드

    }

    @Override

    public void requestDestroyed(ServletRequestEvent sre) {

        // 요청이 완료되었을 때 실행할 코드

    }

}
```

☞ Listener 의 등록 방법

1) Web.xml 을 통한 등록

```
<listener>
    <listener-class>com.example.MyAppListener</listener-class>
</listener>
```

8253jang@daum.net
장희정

2) 어노테이션을 통한 등록

```
@WebListener
public class MyAppListener implements ServletContextListener {
        // 이벤트 처리 코드
}
```

☞ Listener 를 사용하는 이유

- **애플리케이션 초기화 작업** : 애플리케이션이 시작될 때 필요한 초기 설정을 수행하거나 리소스를 로드할 수 있다.
- **세션 관리** : 사용자의 세션이 생성되거나 소멸될 때 상태를 관리하거나 필요한 리소스를 정리할 수 있다.
- **이벤트 감지** : 애플리케이션 전반에서 발생하는 이벤트를 감지하고, 필요한 작업을 자동으로 수행할 수 있다.

Listener 는 이벤트에 반응하여 웹 애플리케이션의 생명주기나 상태를 관리하고, 요청 및 응답의 처리 흐름을 제어하는 데 중요한 역할을 한다.