

* 예외 페이지

- 예외 상황이 발생했을 경우 웹 컨테이너(톰캣)에서 제공되는 기본적인 예외페이지가 보여집니다.
- 개발 과정에서는 이러한 예외 페이지를 보고 어떤 에러가 발생했는지 알 수 있기 때문에 오류를 수정하는 데 도움이 됩니다.
- 그러나 사용자에게 상용 서비스를 제공하고 있는데 이러한 딱딱한 페이지가 보여진다면 사용자로 하여금 불쾌감을 일으키고, 해당 사이트에 대한 신뢰도가 하락하게 됩니다.
- 또한 코드의 일부가 노출되어 보안 측면에도 좋지 않습니다.
- 그래서 개발자가 따로 만들어 둔 에러 페이지로 유도하여 사용자에게 친숙한 페이지를 보여줍니다.

* HTTP 주요 응답 상태 코드

1. 404: 요청한 URL을 찾을 수 없는 경우.
2. 500: 서버측 내부 오류로 인해 페이지가 나타나지 않는 경우(java, JSP 페이지 내의 코드오류)
3. 200: 요청을 성공적으로 처리함.
4. 307: 임시로 페이지를 리다이렉트함.
5. 400: 클라이언트의 요청이 잘못된 구문으로 작성됨.
6. 405: 요청 방식을 허용하지 않음(GET, POST 등)
7. 503: 서버가 일시적으로 서비스를 제공할 수 없음(일시적 서버과부하, 서버 임시 보수 등)

* JSP에서 에러를 처리하는 방법

1. 직접 예외를 처리하기

- 자바의 키워드인 try ~ catch를 사용하여 직접 개발자가 예외를 처리합니다.

2. 에러를 처리할 페이지를 따로 지정하기

- JSP는 실행 도중 예외가 발생할 때 톰캣 기본 에러화면 대신 개발자가 지정한 JSP페이지를 보여줄 수 있는 기능을 제공하고 있습니다.
- 에러가 발생하면 보여줄 JSP페이지는 페이지 지시자(directive)의 `errorPage` 속성을 사용하여 지정합니다.
- 예외가 발생할 것으로 예상되는 페이지에 예외가 발생했을 때 보여줄 페이지를 지정합니다.

ex) `<%@ page errorPage="예외가 발생했을 시 보여줄 페이지" %>`

- 에러 발생시 유도된 페이지에는 페이지 지시자태그로 `isErrorPage` 속성을 사용하여 `true`로 값을 설정합니다.

ex) `<%@ page isErrorPage="true" %>`

- 에러 페이지로 지정된 JSP파일 내에서는 예외를 처리해주는 `exception` 내장객체를 사용할 수 있습니다.
- 인터넷 익스플로러 브라우저의 경우 전체 응답 결과의 데이터가 512바이트보다 작을 경우 마이크로소프트에서 자체 제공하는 에러페이지를 출력합니다.

3. 응답 상태 코드별로 에러 페이지 지정하기

- JSP는 에러 코드별로 사용할 에러 페이지를 web.xml파일 수정을 통해 지정할 수 있습니다.
- 이렇게 지정한 에러페이지는 일반 JSP파일과 동일하게 작성하면 됩니다.

4. 예외 타입별로 에러 페이지 지정하기

- 발생하는 예외의 종류별로도 에러 페이지를 지정할 수 있습니다. web.xml파일을 수정합니다.

* 에러 페이지의 우선순위

- 에러 페이지를 여러 방법으로 지정한 경우 다음의 우선순위에 따라 사용할 에러 페이지를 선택합니다.

1. 페이지 지시자 태그의 `errorPage`속성에 지정한 페이지.
2. web.xml에 지정한 에러 타입에 따른 페이지.
3. web.xml에 지정한 응답 상태 코드에 따른 페이지.
4. 위 3항목에 해당하지 않을 경우 톰캣이 제공하는 에러 페이지.

* JSP Action Tag

- JSP 페이지 내에서 어떤 동작을 하도록 지시하는 태그입니다.
- 종류에는 페이지 이동을 강제하는 `forward`, 페이지를 삽입하는 `include`, `forward`나 `include`를 할 때 값을 지정하는 `param`, 자바의 클래스와 연동하는 `useBean` 등이 있습니다.
- 1. `forward`: 현재의 페이지에서 다른 특정 페이지로 전환할 때 사용합니다.
- 2. `include`: 현재 페이지에 다른 페이지를 삽입할 때 사용합니다.
- 3. `param`: `forward` 및 `include` 태그에 데이터를 전달할 목적으로 사용되는 태그입니다. `name`과 `value` 속성으로 이루어져 있습니다.

* `forward` vs `sendRedirect`

1. `forward`

- 요청 받은 요청객체(request)를 위임하는 컴포넌트에 요청 객체값을 동일하게 전달할 수 있습니다.
- 요청을 처리함과 동시에 해당 결과를 바로 보여줘야 하는 경우
- ex) 게시판 목록에서 글 제목을 클릭했을 때 바로 내용을 보여줘야 하는 경우.

2. `sendRedirect`

- 요청 받은 요청객체를(request) 위임하는 컴포넌트에 전달하는 것이 아닌, 새로운 요청객체를 생성합니다.
- 요청을 처리한 다음 새로운 요청으로 작업을 해야할 경우에 사용합니다.
- ex) 회원가입 요청 처리 후 단순히 메인페이지로 안내해야 할 경우.