

4장. JSP페이지의 내장 객체(IMPLIED OBJECT)와 영역(SCOPE)

학습목표 : 웹 컨테이너는 JSP 페이지에서 사용되는 9개의 객체를, 객체의 생성 없이 바로 사용할 수 있도록 제공 한다. 이들 객체들을 JSP의 내장객체(Implicit Object)라고 부르는데, 이번 장에서는 이들 내장 객체가 무엇이며 어떻게 쓰이는지 그리고 이들 중 영역을 가지는 4개의 객체에 대해 학습한다.

목차(1/2)

- 1. 내장객체(Implicit Object)의 개요
- 2. 내장객체(Implicit Object)의 종류
 - (1) request 객체
 - (2) response 객체
 - (3) out 객체
 - (4) pageContext 객체
 - (5) session 객체

목차(2/2)

- (6) application 객체
- (7) config 객체
- (8) page 객체
- (9) exception 객체
- 3. 내장객체(Implicit Object)의 영역

1. 내장객체(Implicit Object)의 개요

- JSP에서 제공하는 객체
 - ▣ request, response, out, session, application, pageContext, page, config, exception
 - ▣ request, session, application, pageContext 내장 객체는 속성(attribute)값을 저장하고 읽을 수 있는 메소드인 setAttribute()메소드와 getAttribute()메소드를 제공.

2. 내장객체(Implicit Object)의 종류

□ request 객체

- ▣ 웹 브라우저의 요청 정보를 저장하고 있는 객체
- ▣ 입력폼에 입력한 사용자의 요구사항을 얻어낼 수 있도록 요청 메소드를 제공

2. 내장객체 (Implicit Object)의 종류

□ request 객체

▣ 사용자의 요구사항을 얻어내는 요청 메소드

- String **getParameter(name)** : 파라미터 변수 name에 저장된 변수 값을 얻어내는 메소드
- String[] **getParameterValues(name)** : 파라미터 변수 name에 저장된 모든 변수 값을 얻어내는 메소드
- Enumeration **getParameterNames()** : 요청에 의해 넘어오는 모든 파라미터 변수를 java.util.Enumeration 타입으로 리턴

2. 내장객체(Implicit Object)의 종류

□ request 객체

- 요청된 파라미터의 값 외에도 웹 브라우저와 웹 서버의 정보도 가져올 수 있음
- 웹 브라우저, 웹 서버 및 요청 헤더의 정보를 가져올 때 사용되는 메소드
 - String **getProtocol()** : 웹 서버로 요청 시, 사용 중인 프로토콜 리턴.
 - String **getServerName()** : 웹 서버로 요청 시, 서버의 도메인 이름을 리턴

2. 내장객체(Implicit Object)의 종류

□ request 객체

▣ 사용자의 요구사항을 얻어내는 요청 메소드

- String **getMethod()** : 웹 서버로 요청 시, 요청에 사용된 요청 방식(GET, POST, PUT 등)을 리턴
- String **getQueryString()** : 웹 서버로 요청 시, 요청에 사용된 QueryString을 리턴
- String **getRequestURI()** : 웹 서버로 요청 시, 요청에 사용된 URL로부터 URI값을 리턴
- String **getRemoteHost()** : 웹 서버로 정보를 요청한 웹 브라우저의 호스트 이름을 리턴

2. 내장객체(Implicit Object)의 종류

□ request 객체

▣ 사용자의 요구사항을 얻어내는 요청 메소드

- String **getRemoteAddr()** : 웹 서버로 정보를 요청한 웹 브라우저의 IP주소를 리턴
- String **getServerPort()** : 서버의 Port번호를 리턴
- String **getContextPath()** : 해당 JSP페이지가 속한 웹 어플리케이션의 컨텍스트 경로를 리턴
- String **getHeader(name)** : HTTP 요청 헤더(header) 헤더이름 name에 해당하는 속성 값을 리턴.
- Enumeration **getHeaderNames()** : 웹 서버로 요청 시, HTTP 요청 헤더(header)에 있는 모든 헤더이름을 리턴.

2. 내장객체(Implicit Object)의 종류

□ response 객체

- 웹 브라우저의 요청에 대한 응답 정보를 저장하고 있는 객체
- 응답 정보와 관련하여 주로 헤더 정보 입력, 리다이렉트 등의 기능을 제공

2. 내장객체 (Implicit Object)의 종류

□ response 객체

▣ response 객체의 메소드

- **void setHeader(name, value)** : 헤더정보 값을 수정.
- **void setContentType(type)** : 웹 브라우저의 요청 결과로 보일 페이지의 contentType을 설정
- **void sendRedirect(url)** : 페이지를 이동.

2. 내장객체(Implicit Object)의 종류

□ out 객체

- ▣ JSP페이지의 출력할 내용을 가지도 있는 출력 스트림 객체.
- ▣ 표현식(<%=문장%>) 과 같음
- ▣ out 객체의 메소드
 - **boolean isAutoFlush()** : 출력 버퍼가 다 찼을 때 처리여부를 결정.
 - **int getBufferSize()** : 전체 출력 버퍼의 크기를 리턴.

2. 내장객체(Implicit Object)의 종류

□ out 객체

▣ out 객체의 메소드

- **int getRemaining()** : 현재 남아 있는 출력 버퍼의 크기 리턴.
- **void clearBuffer()** : 출력 버퍼에 저장되어 있는 내용을 비움.
- **String println(str)** : 주어진 내용을 출력. 이때 줄 바꿈은 적용되지 않음.
- **void flush()** : 출력 버퍼의 내용을 웹 브라우저에 전송하고 비움.
- **void close()** : 출력 버퍼의 내용을 웹 브라우저에 전송하고 출력 스트림을 닫음.

2. 내장객체(Implicit Object)의 종류

□ pageContext 객체

- ▣ JSP페이지 대한 정보를 저장하고 있는 객체.
- ▣ 다른 내장객체를 구하거나, 페이지의 흐름제어 그리고 에러데이터를 얻어낼 때 사용.
- ▣ pageContext 내장 객체의 메소드
 - ServletRequest **getRequest()** : request객체를 얻어냄.
 - ServletResponse **getResopnse()** : response객체를 얻어냄

2. 내장객체(Implicit Object)의 종류

□ pageContext 객체

▣ pageContext 내장 객체의 메소드

- JspWriter **getOut()** : out객체를 얻어냄
- HttpSession **getSession()** : session객체를 얻어냄
- ServletContext **getServletContext()** : application객체를 얻어냄
- Object **getPage()** : page 객체를 얻어냄
- ServletConfig **getServletConfig()** : config객체를 얻어냄
- Exception **getException()** : exception객체를 얻어냄

2. 내장객체(Implicit Object)의 종류

□ session 객체

- 하나의 웹 브라우저 내에서 정보를 유지하기 위한 세션 정보를 저장하고 있는 객체
- 웹 브라우저(클라이언트)당 1개가 할당.
 - 주로 회원관리 시스템에서 사용자 인증에 관련된 작업을 수행할 때 사용.

2. 내장객체(Implicit Object)의 종류

□ session 객체

▣ session 내장 객체의 메소드

- String **getId()** : 해당 웹 브라우저에 대한 고유한 세션 ID를 리턴.
- long **getCreationTime()** : 해당 세션이 생성된 시간을 리턴.
- long **getLastAccessedTime()** : 마지막 접근시간을 리턴.
- void **setMaxInactiveInterval(time)** : 세션 유지시간을 지정
- int **getMaxInactiveInterval()** : 기본 값은 30분으로 세션 유지시간을 리턴 .
- boolean **isNew()** : 새로 생성된 세션의 경우 true 값을 리턴.
- void **invalidate()** : 세션을 무효화.

2. 내장객체(Implicit Object)의 종류

□ application 객체

- ▣ 웹 어플리케이션 Context의 정보를 저장하고 있는 객체.
- ▣ 서버의 설정 정보, 자원에 대한 정보, 어플리케이션이 실행되는 동안에 발생할 수 있는 이벤트 로그 정보등을 제공.
- ▣ 웹 어플리케이션 당 1개의 객체가 생성.
 - 주로 방문자 카운트와 같은 하나의 웹 어플리케이션에서 공유하는 변수에 사용

2. 내장객체(Implicit Object)의 종류

□ application 객체

▣ application 객체 메소드

- String getServerInfo() : 웹 컨테이너의 이름과 버전을 리턴.
- String getMimeType(fileName) : 지정한 파일의 MIME타입 리턴.
- String getRealPath(path) : 지정한 경로를 웹 어플리케이션 시스템상의 경로로 변경하여 리턴.
- void log(message) : 로그 파일에 message를 기록.

2. 내장객체(Implicit Object)의 종류

□ config 객체

- ▣ JSP페이지 대한 설정정보를 저장하고 있는 객체.
- ▣ 서블릿이 초기화되는 동안 참조해야 할 정보를 전달.
- ▣ 컨테이너당 1개의 객체가 생성.

2. 내장객체(Implicit Object)의 종류

□ config 객체

▣ config 내장 객체 메소드

- Enumeration getInitParameterNames() : 모든 초기화 파라미터 이름을 리턴.
- String getInitParameter(name) : 이름이 name인 초기화 파라미터의 값을 리턴.
- String getServletName() : 서블릿의 이름을 리턴.
- ServletContext getServletContext() : 서블릿 ServletContext 객체를 리턴.

2. 내장객체(Implicit Object)의 종류

- page 객체
 - ▣ JSP 페이지를 구현한 자바 클래스 객체
 - ▣ 웹 컨테이너는 자바만을 스크립트 언어로 지원하지기 때문에 page 객체는 현재 거의 사용되지 않음.

2. 내장객체(Implicit Object)의 종류

□ exception 객체

- ▣ JSP 페이지에서 예외가 발생한 경우에 사용되는 객체

- ▣ exception 객체의 메소드

- String getMessage() : 발생한 예외의 메시지를 리턴.
- String toString() : 발생한 예외 클래스명과 메시지 리턴.
- String printStackTrace() : 예외발생시 예외가 발생한 곳을 추적함.

3. 내장객체(Implicit Object)의 영역

- 웹 어플리케이션은 page, request, session, application 이라는 4개의 영역을 가짐
- 내장 객체의 영역 : 객체의 유효기간
 - ▣ 객체를 누구와 공유할 것인가를 나타냄
- **page영역**
 - ▣ 한 번의 웹 브라우저(클라이언트)의 요청에 대해 하나의 JSP페이지가 호출.

3. 내장객체(Implicit Object)의 영역

□ request영역

- 한 번의 웹 브라우저(클라이언트)의 요청에 대해 같은 요청을 공유하는 페이지가 대응.
- 같은 request영역
 - include 액션 태그, forward 액션 태그를 사용하면 request 객체를 공유하게 됨

3. 내장객체(Implicit Object)의 영역

□ session영역 :

- ▣ 하나의 웹 브라우저 당 1개의 session객체가 생성.
- ▣ 같은 session영역
 - 같은 웹 브라우저 내에서는 요청되는 페이지들
 - 주로 회원관리에서 회원인증에 사용.

3. 내장객체 (Implicit Object)의 영역

□ application영역

- 하나의 웹 어플리케이션 당 1개의 application 객체가 생성.
- 같은 application영역
 - 같은 웹 어플리케이션에 요청되는 페이지들
 - /eud 웹 어플리케이션에서는 같은 application객체를 공유