**서블릿이란?**

**동적 웹페이지 생성의 필요성**

\* 웹 페이지가 사용자가 보낸 정보에 기반하는 경우

\* 웹 페이지가 수시로 바뀌는 정보에 기반하는 경우

\* 웹 페이지가 데이터베이스나 서버의 다른 자원에 기반하는 경우

**서버사이드 자바**

\* 애플릿은 다운로드에 시간이 오래걸리고, 접근권한에 한계가 있다

\* 서버 사이드 자바는 서버측에서 실행되고 결과만 보내기에 다운로드 시간이 짧아지고, 기존의 애플리케이션과 마찬가지로 접근권한을 갖기에 애플릿의 한계를 극복한다

**서블릿의 개념**

\* 서블릿은 자바 프로그램이 서버의 역할을 할 수있도록 하는 확장이다

\* 보안성이 있고, 이식가능하며 사용이 쉬워서 CGI를 대체한다

\* 웹 서버의 요청에 따라 동적으로 모듈을 불러들인다

\* (당연히) 서블릿은 JVM에서 작동된다

\* (당연히) 서버측에서 실행되기 때문에 브라우저를 가리지 않는다

**서블릿이 하는 일**

A. Request Header와 유저가 보낸 데이터를 받는다

B. 결과를 생성한다

C. Response Header와 생성된 결과를 보낸다

**자바 서블릿의 작동 원리**

1. 브라우저가 정보를 요청

2. 웹 서버(Tomcat 등의 Servlet 컨테이너)가 이를 받아들여 서블릿에게 요청한다

3. 서블릿은 요청을 처리하여 응답한다

4. 웹 서버는 응답을 브라우저로 보내준다

5. 브라우저가 결과를 보여준다

**자바 서블릿의 대안**

\* CGI 프로그램: 매 요청마다 프로세스가 새로 생성되어 반응이 느리다

\* NSAPI, ISAPI 등이 있음

**자바 서블릿의 장점**

효율성: 프로세스를 생성하는 CGI에 비해 가벼운 자바 스레드만을 생성하여 처리하기에 효율적

지속성: 메모리에 상주하면서 상태를 관리할 수 있다

이식성

Robustness: 예외 처리, 쓰레기 수집 등이 메모리 누수를 막아주고, 다양한 클래스 라이브러리가 지원되어 개발이 간편하다

확장성: 상속 등을 통해 확장 가능

보안성

강력성: 웹 서버와 직접 통신하고, 데이터베이스, 세션 관리가 가능

편리성: HTML 데이터 해석과 HTTP 헤더 분석, 쿠키 관리 등을 제공

**자바 서블릿의 구성**

javax.servlet : 보편적인 서블릿 인터페이스가 구현되어 있음

javax.servlet.http : HTTP용 서블릿 인터페이스가 구현되어 있음

public void doGet(HttpServletRequest request, HttpServletResponse response)

public void doPost(HttpServletRequest request, HttpServletResponse response)

-> 실질적으로 요청에 따라 적절한 응답을 보내주는 처리를 수행하는 함수. 대게 이 함수를 오버라이딩하여 사용하게 된다. 이 함수 속에 직접 HTML 코드를 넣어 결과를 생성해도 좋지만, 그것보다는 JSP로 forwarding하여 처리하는 것이 깔끔하고 유지보수에도 좋다.

클라이언트의 요청 방식

|  |  |
| --- | --- |
| **전송 방식** | **설멍** |
| Get 방식 | 주소 창을 타고 넘어가기 때문에 서버로 보내는 데이터를 사용자가 그대로 볼 수 있다. 그래서 보안에 취약하다. 255자 이하의 적은 양의 데이터를 전송한다. |
| Post 방식 | Html header를 타고 넘어가기 때문에 보안에 강하다. 255자 이상의 대용량의 데이터를 전송한다. |

<form>태그에 사용된 주요 속성은 method와 action이다.

|  |  |
| --- | --- |
| 속성 | 설명 |
| Method | 어떤 방식으로 데이터를 넘겨 줄 것인지를 결정한다.  Get과 post 방식 중에서 하나를 선택한다.  Post는 사용자가 입력한 내용을 발송할 때 그 내용을 공개하지 않는다.  Get은 사용자가 입력한 결과를 알 수 있도록 내용을 공개한다. |
| Action | 전송(submit) 버튼을 누르면 action 속성 다음에 기술한 URL에 지정된 파일로 이동한다. |

**<form>태그를 이용한 get 방식의 요청 예**

<form method=”get” action=”CallServlet”>

<input type=”submit” value=”전송”>

</form>

**<form>태그를 이용한 post 방식의 요청의 예**

<form method=”post” action=”CallServlet”>

<input type”submit” value=”전송”>

</form>

<form>태그 외에도 html의 <a>태그를 사용하여 링크를 걸어 주면 서블릿은 get방식으로 요청한 것으로 인식한다.

**<a>태그를 이용한 get방식 요청의 예**

<a href=”CallServlet”> get방식의 요청</a>

[**JSP이란?**](http://javahwan.tistory.com/53)

JSP는 **JavaServer Pages**의 약자이며 스크립트 언어이다.

JSP의 특징

* 자바를 기반으로 하는 스크립트 언어로 자바가 제공하는 기능을 **그대로** 사용할 수 있다.
* HTTP와 같은 프로토콜에 따라 **클라이언트**의 **요청**을 **처리**하고 **응답**한다.
* HTML, XML 등 클라이언트가 요청한 **문서**를 **생성**하는데 주로 사용된다.
* 서블릿/EJB 등의 **엔터프라이즈 기술**들과 잘 **융합**된다.
* 표현 언어, 표현식, 스크립트릿 등 다양한 스크립트 요소와 액션 태그 등을 제공함으로써 보다 쉽게 웹 어플리케이션을 프로그래밍 할 수 있도록 도와준다.

JSP는 HTTP 프로토콜을 알맞게 처리할 수 있도록 설계되었다. 예를 들어, JSP는 웹 브라우저가 전송한 데이터를 읽어오고 생성한 데이터를 웹 브라우저에 전송하는 것처럼 웹 어플리케이션을 개발하는 데 필요한 기능등을 스크립트 차원에서 제공하고 있다. 또한, 사용자의 세션을 쉽게 관리 할 수 있도록 자체적으로 **세션 기능**을 **제공**하고 있다.

JSP는 주로 웹 브라우저의 요청에 따라 알맞은 HTML 결과 화면을 생성하기 위해 사용된다. 하지만 HTML뿐만 아니라 XML문서, PDF 문서 등 다양한 문서를 생성하는 데에도 JSP를 사용할 수 있다.

**JSP 기본 태그**

|  |  |  |
| --- | --- | --- |
| **종류** | **사용용도** | **형식** |
| 스크립트릿 | 자바 코드를 기술함 | <% %> |
| 선언 | 변수와 메소드를 선언함 | <%! %> |
| 표현식 | 계산식이나 함수를 호출한 결과를 문자열 형태로 출력함 | <%= %> |
| 주석 | JSP페이지에 설명을 넣음 | <%-- --%> |
| 지시자 | JSP페이지의 속성을 지정함 | <%@ %> |

**request 내부 객체의 요청 파라미터 관련 메소드**

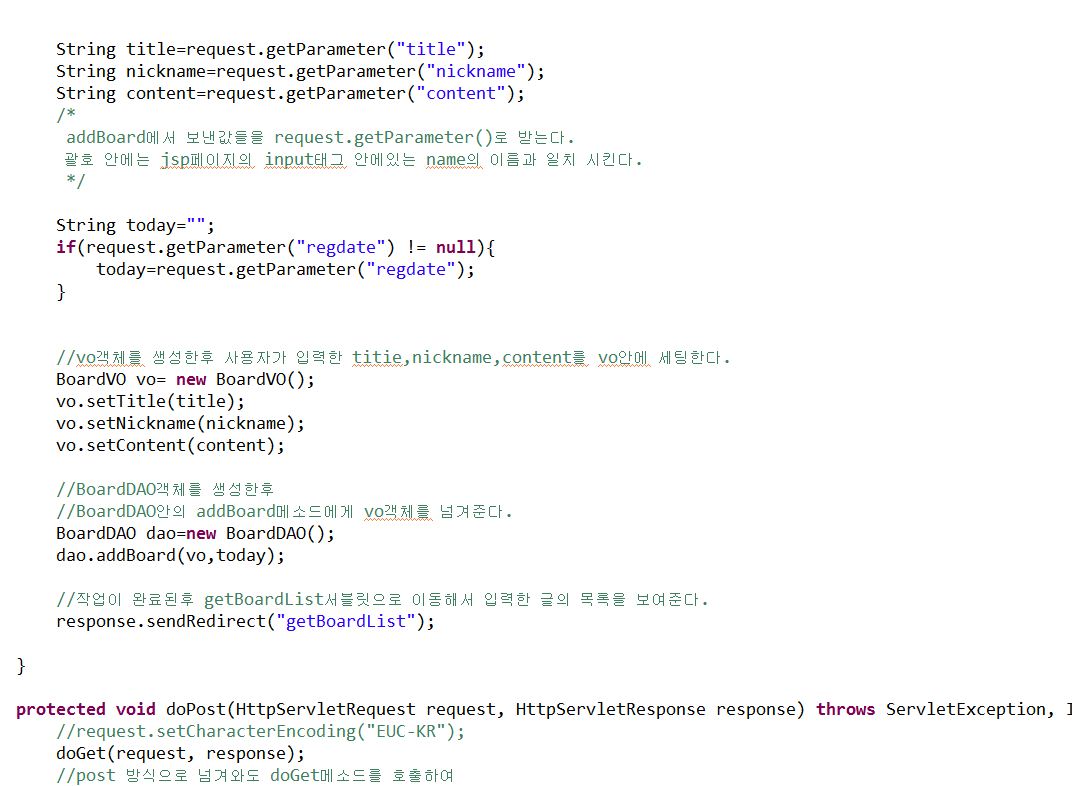
|  |  |
| --- | --- |
| 메소드 | 설명 |
| String getParameter(name) | 파라미터 변수 name에 저장된 변수를 얻어내는 메소드로, 이때 변수의 값은 String으로 리턴된다. |
| String[] getParameterValues(name) | 파라미터 변수 name에 저장된 모든 변수값을 얻어내는 메소드로, 이때 변수의 값은 String 배열로 리턴된다. checkbox에서 주로 사용된다. |
| Enumeration getParameterNames() | 요청에 의해 넘어오는 모든 파라미터 변수를 java.util.Enumeration 타입으로 리턴한다. |

**- request 내장 객체의 웹 브라우저, 웹 서버 및 요청 헤더의 정보 관련 메소드**

|  |  |
| --- | --- |
| 메소드 | 설명 |
| String getProtocol() | 웹 서버로 요청 시, 사용 중인 프로토콜을 리턴한다. |
| String getServerName() | 웹 서버로 요청 시, 서버의 도메인 이름을 리턴한다. |
| String getMethod() | 웹 서버로 요청 시, 요청에 사용된 요청 방식(GET, POST, PUT 등)을 리턴한다. |
| String getQueryString() | 웹 서버로 요청 시, 요청에 사용된 QueryString을 리턴한다. |
| String getRequestURI() | 웹 서버로 요청 시, 요청에 사용된 URL 로부터 URI 값을 리턴한다. |
| String getRemoteAddr() | 웹 서버로 정보를 요청한 웹 브라우저의 IP주소를 리턴한다. |
| int getServerPort() | 웹 서버로 요청 시, 서버의 Port번호를 리턴한다. |
| String getContextPath() | 해당 JSP 페이지가 속한 웹 어플리케이션의 콘텍스트 경로를 리턴한다. |
| String getHeader(name) | 웹 서버로 요청 시, HTTP 요청 헤더(header)의 헤더 이름인 name에 해당하는 속성값을 리턴한다. |
| Enumeration getHeaderNames() | 웹 서버로 요청 시, HTTP 요청 헤더(header)에 있는 모든 헤더 이름을 리턴한다. |

**게시판에 글 입력하는 예제**

****

****