

HttpServletRequest 살펴보기

```
package javax.servlet;

...

public interface ServletRequest {
    Object getAttribute(String name);
    // Request가 갖는 속성 name에 대한 값을 반환한다

    Enumeration<String> getAttributeNames();
    // Request가 갖는 속성들의 name들에 대한 Enumeration 객체를 반환한다

    String getCharacterEncoding();
    // Request Body에 사용된 인코딩 정보를 얻는다

    void setCharacterEncoding(String var1) throws UnsupportedEncodingException;
    // Request Body에 대해 인코딩한 정보를 정의한다
    // UnsupportedEncodingException: 이미 인코딩 정보가 지정되어 있을 때 발생하는 예외

    int getContentLength();
    // Request Body의 byte 길이를 얻는다. inputStream으로 body를 읽어올텐데 stream의 길이를 찾는데 사용된다.
    // 길이를 모르면 -1 을 반환하고, Integer.MAX_VALUE 보다 길다면 Integer.MAX_VALUE 를 반환한다.

    long getContentLengthLong();
    // getContentLength가 int형으로 반환한다면 long 형으로 더 긴 body length를 표현가능하다
    // 역시 길이를 모를 경우 -1 을 반환한다

    String getContentType();
    // Request Body의 MIME type을 반환한다
    // 모를경우 null을 반환한다

    ServletInputStream getInputStream() throws IOException;
    // binary data로 Request Body 정보를 담은 ServletInputStream(inputstream)을 반환한다.
    // body를 읽기위해 이 메서드 또는 getReader() 메서드도 이용가능하다.
    // IllegalStateException: getReader() 메서드가 이미 호출되었을 때의 예외(Stream은 한번만 읽을 수 있다)
    // IOException: inputstream에서 발생하는 예외

    String getParameter(String var1);
    // name에 일치하는 request parameter를 반환한다
    // name에 일치하는 parameter가 존재하지 않을 시 null 반환
    // name에 일치하는 parameter가 두개 이상일 경우 첫번째 값을 반환한다
    // name에 일치하는 모든 parameter를 가져오고 싶을 땐 getParameterValues()를 사용해야 한다

    // HTTP Post와 같이 parameter가 request body와 함께 전송된다면 getInputStream() 과 getReader() 메서드를 호출한 경우 이 메서드의 실행을 방해할 수 있다
```

```

Enumeration<String> getParameterNames();
// Request가 갖는 parameter name들에 대한 Enumeration 객체를 반환한다

String[] getParameterValues(String var1);
// name에 일치하는 모든 parameter를 배열로 반환한다
// name에 일치하는 parameter가 존재하지 않을 시 null 반환

Map<String, String[]> getParameterMap();
// Request에 대한 parameter 정보를 java.util.Map 형태로 반환한다
// HttpServlet에서는 query string과 posted form data도 함께 포함되어 반환된다

String getProtocol();
// Request 에 사용된 프로토콜 정보를 반환한다
// form: protocol/majorVersion.minorVersion

String getScheme();
// Request를 만들기 위해 사용된 방법의 이름을 반환합니다.
// ex> http, https, ftp

String getServerName();
// 서버의 이름을 반환합니다

int getServerPort();
// 서버에서 사용중인 포트 번호를 반환합니다

BufferedReader getReader() throws IOException;
// Character Data로 Request Body 정보를 담은 BufferedReader를 반환한다.
// characterEncoding 방법에 맞게 해석하여 반환해준다
// UnsupportedEncodingException: 지원되지 않는 인코딩이거나 해석되지 못할 때
// IllegalStateException: getInputStream() 메서드가 이미 호출되었을 때의 예외(Stream
은 한번만 읽을 수 있다)
// IOException: inputstream에서 발생하는 예외

String getRemoteAddr();
// 요청 클라이언트의 IP 주소를 반환한다

String getRemoteHost();
// 요청 클라이언트의 호스트 이름을 반환한다

void setAttribute(String key, Object o);
// 주어진 이름의 속성을 설정한다

void removeAttribute(String key);
// 주어진 이름의 속성을 제거한다

Locale getLocale();
// 클라이언트에게 제공할 기본 Local을 반환한다
// Accept-Language Header 정보를 통해 설정된다
// Accept-Language Header 정보가 존재하지 않다면 서버의 기본 local로 지정된다

Enumeration<Locale> getLocales();

```

```

// 클라이언트에게 제공할 기본 Local들에 대한 Enumeration을 반환한다

boolean isSecure();
// 안전한 채널을 통해 전송되었는지 여부를 반환한다

RequestDispatcher getRequestDispatcher(String path);
// 해당 path의 RequestDispatcher를 반환한다
// 서블릿 컨테이너가 RequestDispatcher를 반환할 수 없다면 null을 반환한다

/** @deprecated */
String getRealPath(String path);

int getRemotePort();
// Request를 보내 클라이언트의 IP 포트 번호, 혹은 가장 마지막 프록시의 포트 번호를 반환한다

String getLocalName();
// Request를 수신한 Host 이름을 반환한다

String getLocalAddr();
// Request를 수신한 IP 주소를 반환한다

int getLocalPort();
// Request를 수신한 포트 번호를 반환한다

ServletContext getServletContext();
// 가장 최근 해당 ServletRequest를 dispatch한 ServletContext를 반환한다

AsyncContext startAsync() throws IllegalStateException;
// 해당 request를 비동기모드로 설정한다

AsyncContext startAsync(ServletRequest request, ServletResponse response) throws
IllegalStateException;

boolean isAsyncStarted();

boolean isAsyncSupported();

AsyncContext getAsyncContext();

DispatcherType getDispatcherType();
}

```