

5.1 Reading Request Headers

Reading headers is straightforward; just call the `getHeader` method of `HttpServletRequest` with the name of the header. This call returns a `String` if the specified header was supplied in the current request, `null` otherwise. In HTTP 1.0, all request headers are optional; in HTTP 1.1, only `Host` is required. So, always check for `null` before using a request header.

Core Approach



Always check that the result of `request.getHeader` is non-null before using it.

Header names are not case sensitive. So, for example, `request.getHeader("Connection")` is interchangeable with `request.getHeader("connection")`.

Although `getHeader` is the general-purpose way to read incoming headers, a few headers are so commonly used that they have special access methods in `HttpServletRequest`. Following is a summary.

- **`getCookies`**

The `getCookies` method returns the contents of the `Cookie` header, parsed and stored in an array of `Cookie` objects. This method is discussed in more detail in [Chapter 8](#) (Handling Cookies).

- **`getAuthType` and `getRemoteUser`**

The `getAuthType` and `getRemoteUser` methods break the `Authorization` header into its component pieces.

- **`getContentLength`**

The `getContentLength` method returns the value of the `Content-Length` header (as an `int`).

- **`getContentType`**

The `getContentType` method returns the value of the `Content-Type` header (as a `String`).

- **`getDateHeader` and `getIntHeader`**

The `getDateHeader` and `getIntHeader` methods read the specified headers and then convert them to `Date` and `int` values, respectively.

- **`getHeaderNames`**

Rather than looking up one particular header, you can use the `getHeaderNames` method to

get an `Enumeration` of all header names received on this particular request. This capability is illustrated in [Section 5.2](#) (Making a Table of All Request Headers).

- **`getHeaders`**

In most cases, each header name appears only once in the request. Occasionally, however, a header can appear multiple times, with each occurrence listing a separate value. `Accept-Language` is one such example. You can use `getHeaders` to obtain an `Enumeration` of the values of all occurrences of the header.

Finally, in addition to looking up the request headers, you can get information on the main request line itself (i.e., the first line in the example request just shown), also by means of methods in `HttpServletRequest`. Here is a summary of the four main methods.

- **`getMethod`**

The `getMethod` method returns the main request method (normally, `GET` or `POST`, but methods like `HEAD`, `PUT`, and `DELETE` are possible).

- **`getRequestURI`**

The `getRequestURI` method returns the part of the URL that comes after the host and port but before the form data. For example, for a URL of `http://randomhost.com/servlet/search.BookSearch?subject=jsp`, `getRequestURI` would return `"/servlet/search.BookSearch"`.

- **`getQueryString`**

The `getQueryString` method returns the form data. For example, with `http://randomhost.com/servlet/search.BookSearch?subject=jsp`, `getQueryString` would return `"subject=jsp"`.

- **`getProtocol`**

The `getProtocol` method returns the third part of the request line, which is generally `HTTP/1.0` or `HTTP/1.1`. Servlets should usually check `getProtocol` before specifying *response* headers ([Chapter 7](#)) that are specific to HTTP 1.1.