



JEE Course Servlets – Requests

Márcio Fuckner

HvA – Hogeschool van Amsterdam



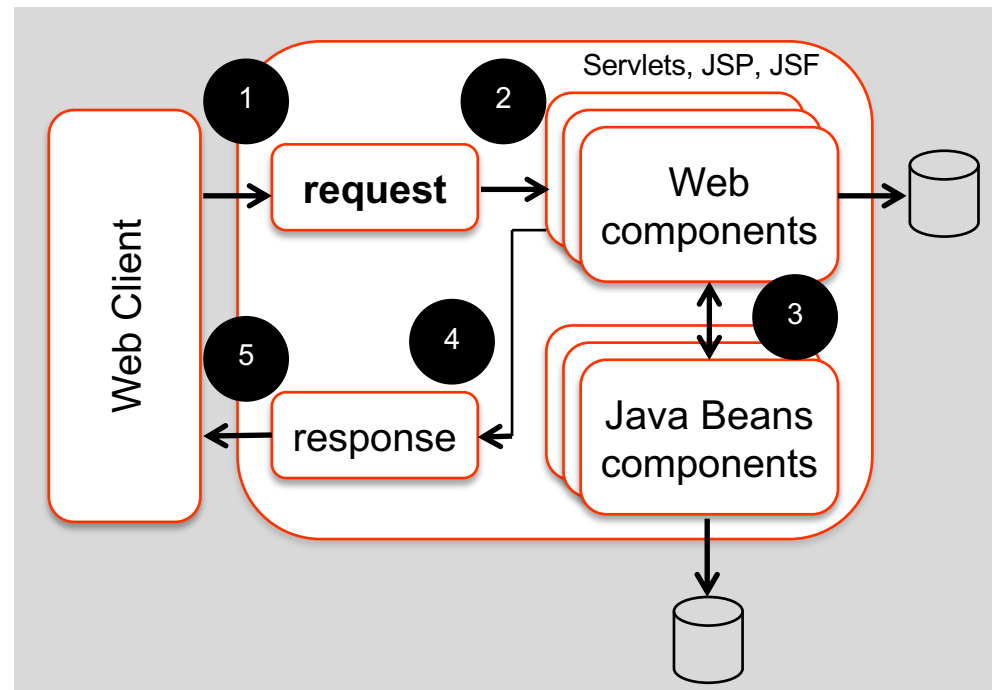
Content

- ▶ What are request objects?
- ▶ How to retrieve query parameters?
- ▶ How to retrieve headers?
- ▶ How to retrieve request information?



Request objects

- ▶ When a servlet receives a request, the container executes the service method passing two references: a **request** and a **response**
- ▶ The request object implements the ***HttpServletRequest*** interface and is used to retrieve client information





Requests methods

- ▶ A request object contains data passed between a client and the servlet. This interface defines methods for accessing the following information:
 - ▶ **Parameters**: explored in this module
 - ▶ **Headers** and request **content**
 - ▶ **Attributes**: will be explored in module 6
- ▶ Some methods of the *HttpServletRequest* interface:

```
String getParameter (String)
```

```
String [] getParameterValues (String)
```

```
Enumeration<String> getParameterNames ()
```

```
String getHeader (String)
```

```
Enumeration<String> getHeaders (String)
```

```
Enumeration<String> getHeaderNames ()
```



Getting a request parameter

- ▶ Query strings are organized in a map of parameters and values.
- ▶ To get a specific parameter we use the *getParameter* method by passing the corresponding parameter name:

```
String email = request.getParameter("email");
```
- ▶ If the parameter has more than one value, we use the *getParameterValues* method, which returns an array of String
- ▶ If the request parameter does not exist, both methods will return null.
- ▶ Syntax:

```
String getParameter(String name);
```

```
String[] getParameterValues(String name);
```



Getting all request parameters

- ▶ To retrieve all request parameters we use the *getParameterNames* method
- ▶ This method returns an *Enumeration<String>* instance, which we can use to iterate over its values
- ▶ If there are no parameters, the method returns an empty *Enumeration*
- ▶ To get the corresponding value during the iteration, one could use the *getParameter* or the *getParameterValues* method
- ▶ Syntax:

```
Enumeration<String> getParameterNames();
```



Getting a header

- ▶ The information exchanged between the client and the server using HTTP is encoded in MIME formats
 - ▶ See: <http://www.iana.org/assignments/media-types/media-types.xhtml>
- ▶ The protocol specification also defines a list of recommended headers (preferred language, content type, content length, cookies, etc.)
- ▶ We can retrieve those headers using a set of methods:

```
String getHeader(String name);  
int getDateHeader(String name);  
int getIntHeader(String name);  
Enumeration<String> getHeaders(String name)
```



Getting all headers

- ▶ To retrieve all available headers, one could use the *getHeaderNames* method
- ▶ This method returns an object of type *Enumeration<String>*
- ▶ If there are no parameters, the method returns an empty *Enumeration*
- ▶ To get the value of each enumeration element one could use the *getHeader* method
- ▶ Syntax:

```
Enumeration getHeaderNames();
```




Other useful request methods

String getAuthType()

String getQueryString()

String getRemoteUser()

String getRemoteAddr()

String getRemoteHost()

String getServerName()

int getServerPort()

String getProtocol()

String getRequestURI()

String getMethod()

String getAuthType()

String getQueryString()

String getRemoteUser()

String getRemoteAddr()

String getRemoteHost()

String getServerName()

int getServerPort()

String getProtocol()

String getRequestURI()

String getMethod()



Reading the HTTP request body

- ▶ For binary content:
 - ▶ You can read the body of HTTP requests using the `getInputStream()` method of `HttpServlet`
- ▶ For text content:
 - ▶ You can read the body of HTTP requests using the `getReader()` method of `HttpServlet`
- ▶ The HTTP protocol also accept multipart data (e.g. sending files)
 - ▶ See: `getPart` and `GetParts` of `HttpServlet`



Further reading

HttpServletRequest interface:

<http://tomcat.apache.org/tomcat-8.0-doc/servletapi/javax/servlet/http/HttpServletRequest.html>