

Руковање заглављем HTTP захтева

У овој вежби су обрађене следеће теме:

1. Руковање заглављима у HTTP захтеву, што омогућује прилагођавање приказа у Веб претраживачу у односу на параметре у заглављу (на пример, детектовање Веб претраживача одакле долази захтев).

2. Прилагођавање садржаја Веб странице (динамичко генерисање помоћу сервлета) у зависности од тога одакле је корисник дошао до странице. Заглавље `Referer` се користи да би се идентификовала локација странице одакле корисник приступа. Овде се могу уочити следећи случајеви који се јављају у пракси:

- ❑ Креирање садржаја сајта у зависности од линка са којег се дошло до странице на сајту (нпр. сајтови за тражење послова).
- ❑ Промена садржаја у зависности да ли се линк одакле се долази налази у зони коју штити *firewall*, или је изван те зоне (безбедносна политика која се имплементира на Веб сајту).
- ❑ Укључивање одговарајућих линкова за повратак на страницу одакле је корисник дошао.
- ❑ Праћење посета са различитих сајтова (нпр. колико су ефикасне рекламе помоћу банера на појединим сајтовима).

ПРИМЕР (requestheader)

Нову веб апликацију **requestheader** треба пријавити серверу **Apache Tomcat** тако што се у датотеку **server.xml** дода контекст апликације.

```
<Context path="/requestheader" docBase="requestheader"
  debug="0" reloadable="true"/>
```

1. Сервлет који приказује информације о заглављу HTTP захтева

У пројекту креирати сервлет **ShowRequestHeaders** који чита информације о главној линији HTTP захтева, и потом информације из заглавља HTTP захтева. Све информације сервлет приказује у форматираној HTML страници. Сервлет треба пријавити у **web.xml** датотеци Веб апликације, што приказује следећи код.

```
<servlet>
  <servlet-name>ShowRequestHeaders</servlet-name>
  <servlet-class>ShowRequestHeaders</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>ShowRequestHeaders</servlet-name>
  <url-pattern>/ShowRequestHeaders</url-pattern>
</servlet-mapping>
```

Код који приказује основне информације о главној линији HTTP захтева (метода, идентификатор ресурса, страинг са упитом захтева и протокол) је:

```
"<B>Request Method: </B>" + request.getMethod() + "<BR>\n" +
"<B>Request URI: </B>" + request.getRequestURI() + "<BR>\n" +
"<B>Request Query String: </B>" + request.getQueryString() + "<BR>\n" +
"<B>Request Protocol: </B>" + request.getProtocol() + "<BR><BR>\n" +
```

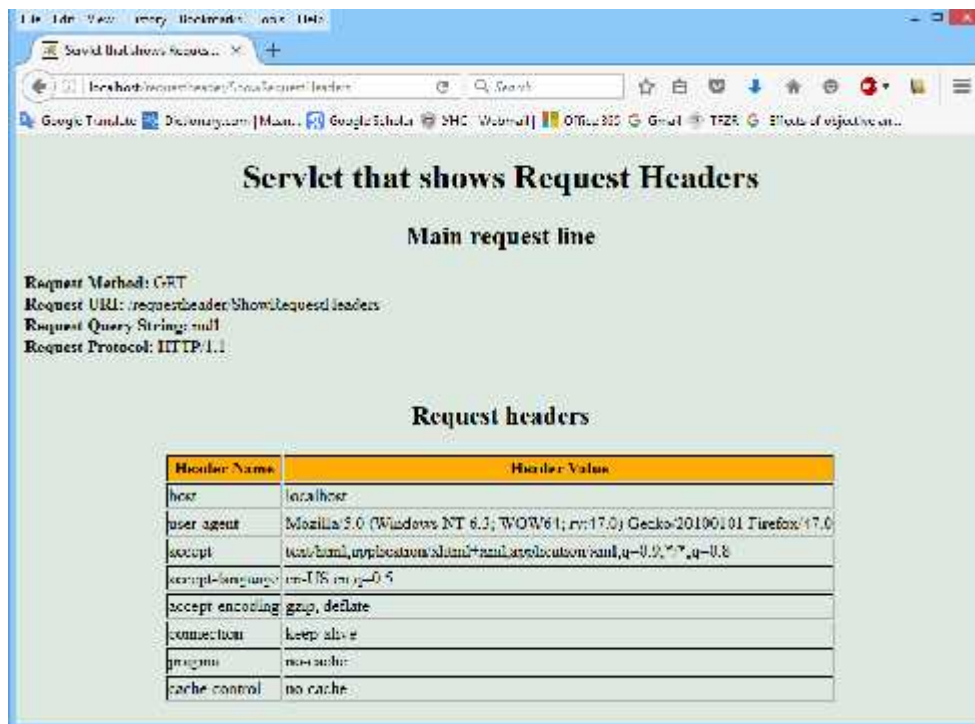
Код који приказује податке из HTTP заглавља у форматираној HTML табели је:

```
"<H2 ALIGN=\"CENTER\">" + "Request headers" + "</H2>\n" +
"<TABLE BORDER=1 ALIGN=\"CENTER\">\n" +
"<TR BGCOLOR=\"#FFAD00\">\n" +
"<TH>Header Name<TH>Header Value");

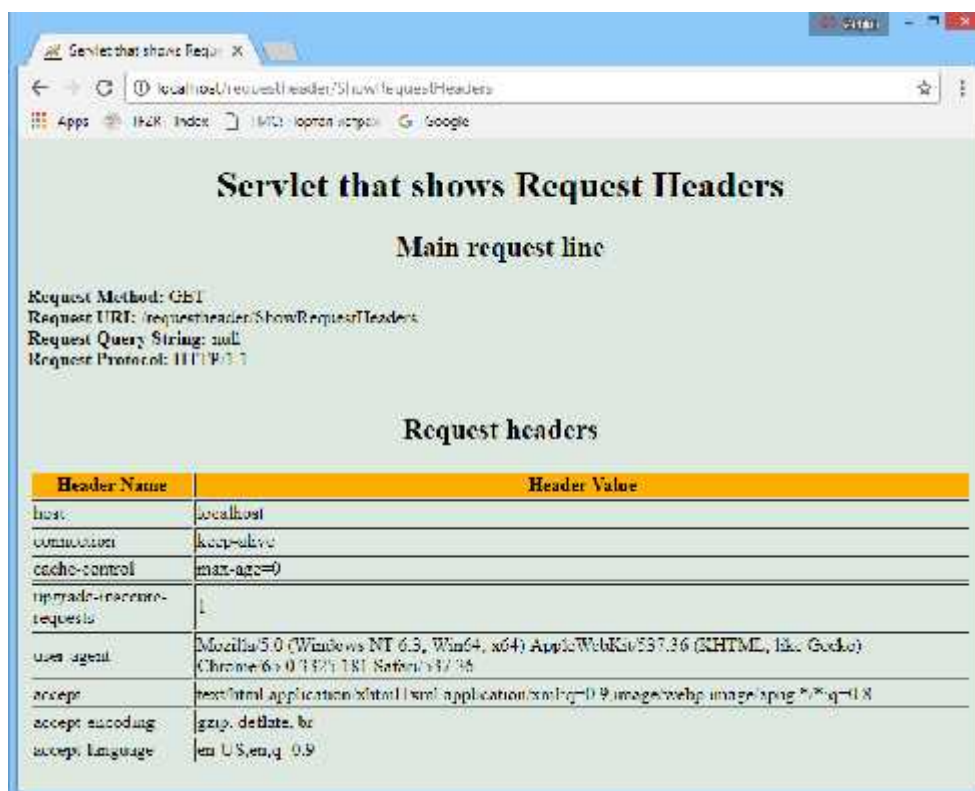
Enumeration headerNames = request.getHeaderNames();
while(headerNames.hasMoreElements()) {
  String headerName = (String)headerNames.nextElement();
  out.println("<TR><TD>" + headerName);
  out.println("    <TD>" + request.getHeader(headerName));
}
out.println("</TABLE>\n</BODY></HTML>");
```

С обзиром да различити Веб претраживачи могу послати различите параметре у HTTP заглављу, потребно је написати код који ће прочитати сва заглавља без обзира на Веб претраживач. Да би се то реализовало све параметре HTTP заглавља је потребно прочитати као колекцију објеката типа **Enumeration** (погледати <https://docs.oracle.com/javase/7/docs/api/java/util/Enumeration.html>), којој се после може приступити и прочитати сваки параметар (петља која пролази кроз све параметре докле год их има).

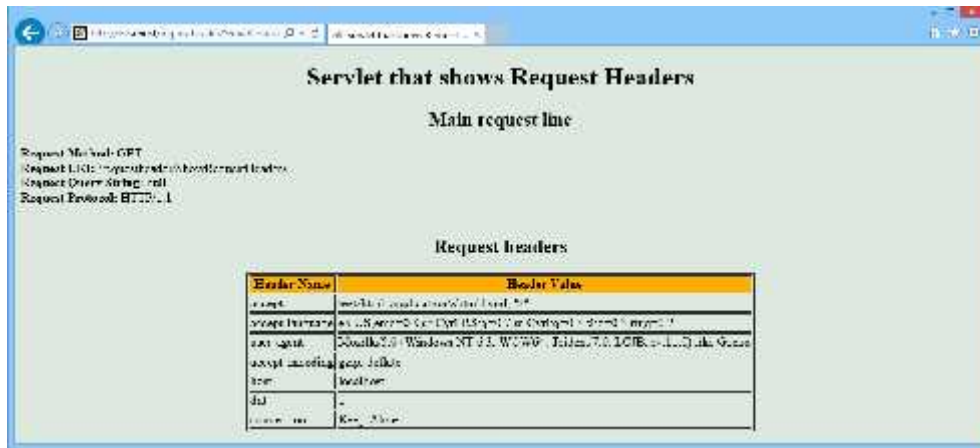
На следећој слици је приказан излаз који штампа сервлет **ShowRequestHeaders** када је покренут из Веб претраживача **Mozilla Firefox 47.0.2**.



На следећој слици је приказан излаз који штампа сервлет `ShowRequestHeaders` када је покренут из Веб претраживача **Google Chrome**.



На следећој слици је приказан излаз који штампа сервлет **ShowRequestHeaders** када је покренут из Веб претраживача **Microsoft Internet Explorer**.



Употреба HTTP заглавља се може искористити за прилагођавање приказа Веб апликација у различитим Веб претраживачима, али и на различитим уређајима.

2. Праћење приступа Веб апликацији (сервлету) са различитих локација

У пројекту креирати сервлет (**GetLinkLocation**) који омогућује проверу локације одакле је стигао захтев, и на основу тога прилагођава садржај странице. Сервлет треба да детектује да ли је позван са главне странице Веб апликације (**index.html**) или неке друге (нпр. **testlink.html**), или директно уносом имена сервлета у адресну линију Веб претраживача. Сервлет треба пријавити у **web.xml** датотеци Веб апликације, што приказује следећи код.

```
<servlet>
    <servlet-name>GetLinkLocation</servlet-name>
    <servlet-class>GetLinkLocation</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>GetLinkLocation</servlet-name>
    <url-pattern>/GetLinkLocation</url-pattern>
</servlet-mapping>
```

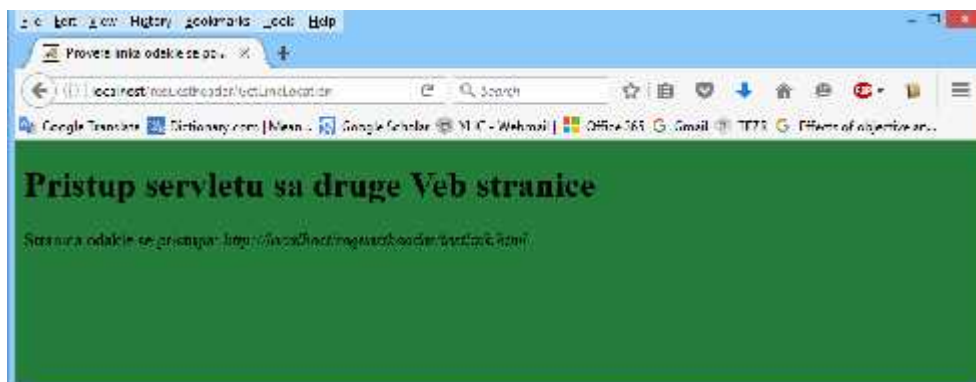
Приликом позива сервлета **GetLinkLocation** помоћу линка са насловне странице Веб апликације (подешена као подразумевана страница у датотеци **web.xml**) приказује само Веб контекст Веб апликације без назива датотеке **index.html**.

http://localhost/requestheader/index.html



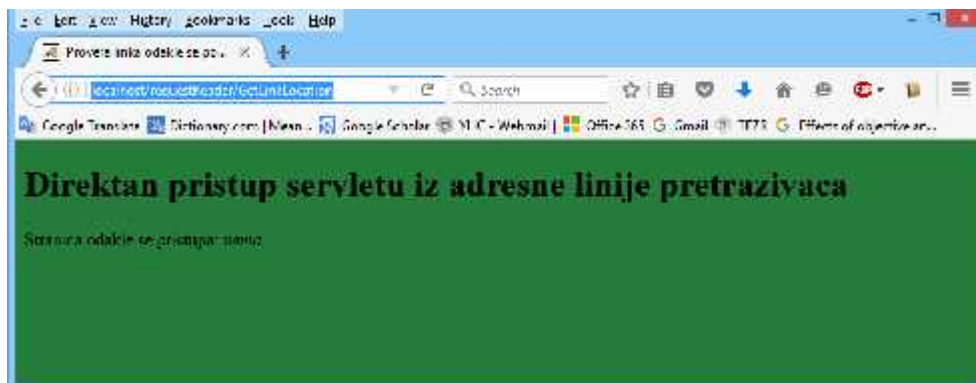
Приликом позива сервлета `GetLinkLocation` помоћу линка са странице `testlink.html` приказује се пуна адреса странице `testlink.html`.

`http://localhost/requestheader/testlink.html`



Када се сервлет `GetLinkLocation` директно позива куцањем адресе у адресној линији Веб претраживача, тада **не постоји страница** са које се прелази (*одакле се долази*) па се приказује информација о томе. Таква ситуација се детектује када је вредност заглавља **Referer** једнака **null**. Позив директним куцање адресе сервлета у адресној линији Веб претраживача даје следећи приказ о позиву:

`http://localhost/requestheader/GetLinkLocation`



Задаци за самостални рад

Задатак 1

Сервлет **ShowRequestHeaders** у решеном примеру 1 из овог упутства модификовати тако да се за детектоване различите Веб претраживаче (нпр. **Mozilla Firefox, Google Chrome, Microsoft Internet Explorer**) помоћу заглавља **User-Agent** обезбеди:

Задатак 1.1. Различито форматирање излазног HTML документа тако да се на пример позадина, текст и наслови различито форматирају за различите претраживаче.

Задатак 1.2. Различито форматирање излазног HTML документа тако да се за сваки детектовани претраживач у HTML документ улинкује различит документ са стиловима (нпр. `mozillastyle.css`, `chromestyle.css`, `explorerstyle.css`) за приказ Веб страница.

Задатак 2

За израду овог задатка се као полазна тачка може користити сервлет **GetLinkLocation** написан у примеру 2.

Задатак 2.1. Написати и тестирати сервлет који на основу локација линка одакле је позван приказује веб странице са информацијама о локацији линка форматиране на различит начин. За тестирање креирати неколико HTML страница са линковима према сервлету.

Задатак 2.1. Написати и тестирати сервлет који пребројава позиве са неколико HTML страница које треба користити приликом тестирања сервлета, и када се са било које странице детектује 5 позива приказати на новој HTML страници табеларни приказ укупног броја позива са сваке од тест страница.

Задатак 3

Написати сервлет који прати позиве са различитих линкова (креирати неколико HTML страница које позивају сервлет) и из различитих Веб претраживача (нпр. **Mozilla Firefox, Google Chrome, Microsoft Internet Explorer**), и када се са било ког претраживача детектује 5 приступа укупно за све линкове, приказује укупне податке о приступу по претраживачима и линковима.

Линкови

- [1] **The Apache Tomcat.** <http://tomcat.apache.org/>
- [2] Marty Hall and Larry Brown. *Core Servlets and JavaServer Pages, Free Online Version of Second Edition.* <http://pdf.coreservlets.com/>. Chapter 5: Handling the Client Request: HTTP Request Headers.
- [3] <https://tomcat.apache.org/tomcat-9.0-doc/servletapi/overview-summary.html>
- [4] <https://tomcat.apache.org/tomcat-9.0-doc/servletapi/index.html>.
- [5] <https://docs.oracle.com/javase/7/docs/api/java/util/Enumeration.html>.