

VEEBIRAKENDUSTE

LOOMINE

MTAT.03.230 (6 EAP)

4. Loeng

Helle Hein

Teema: Serveripoolne programmeerimine – Java servletid

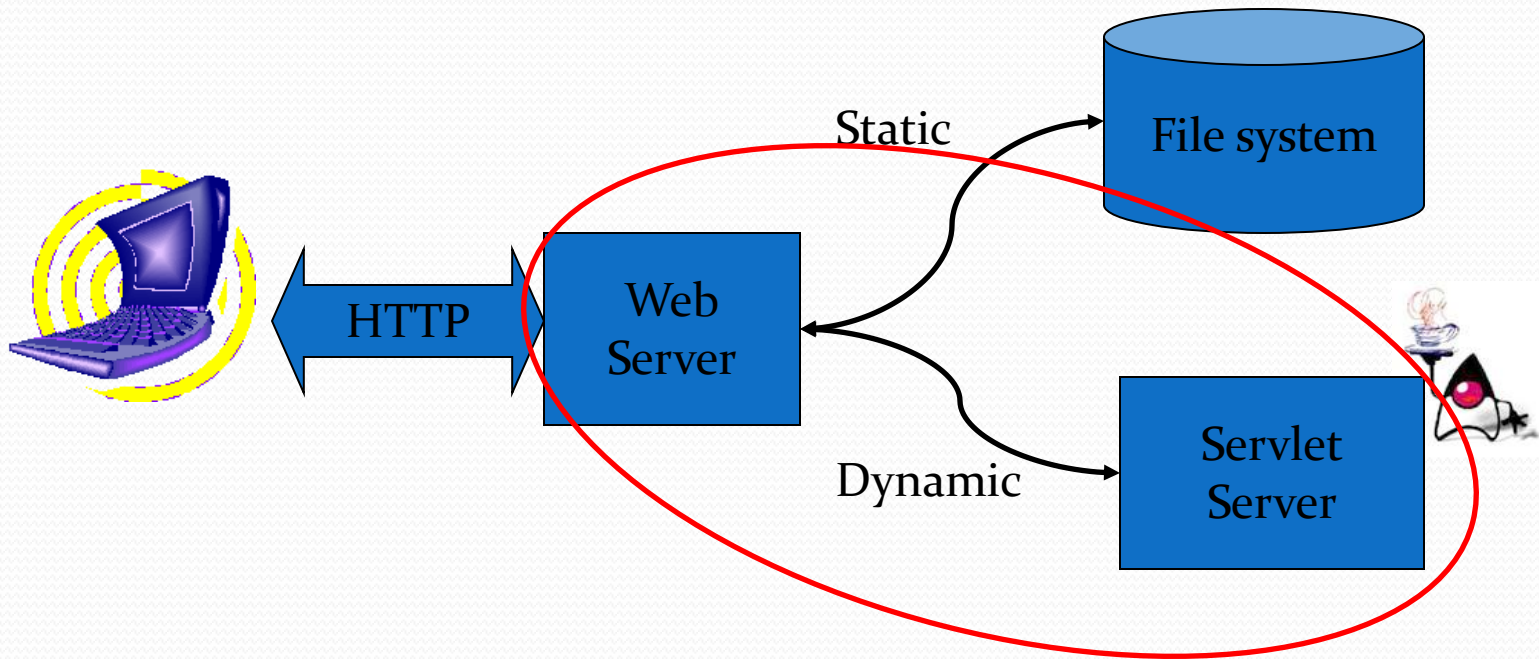
Veebiserveri vastused:

- Staatilised
- Dünaamilised

Mis on Java servletid?

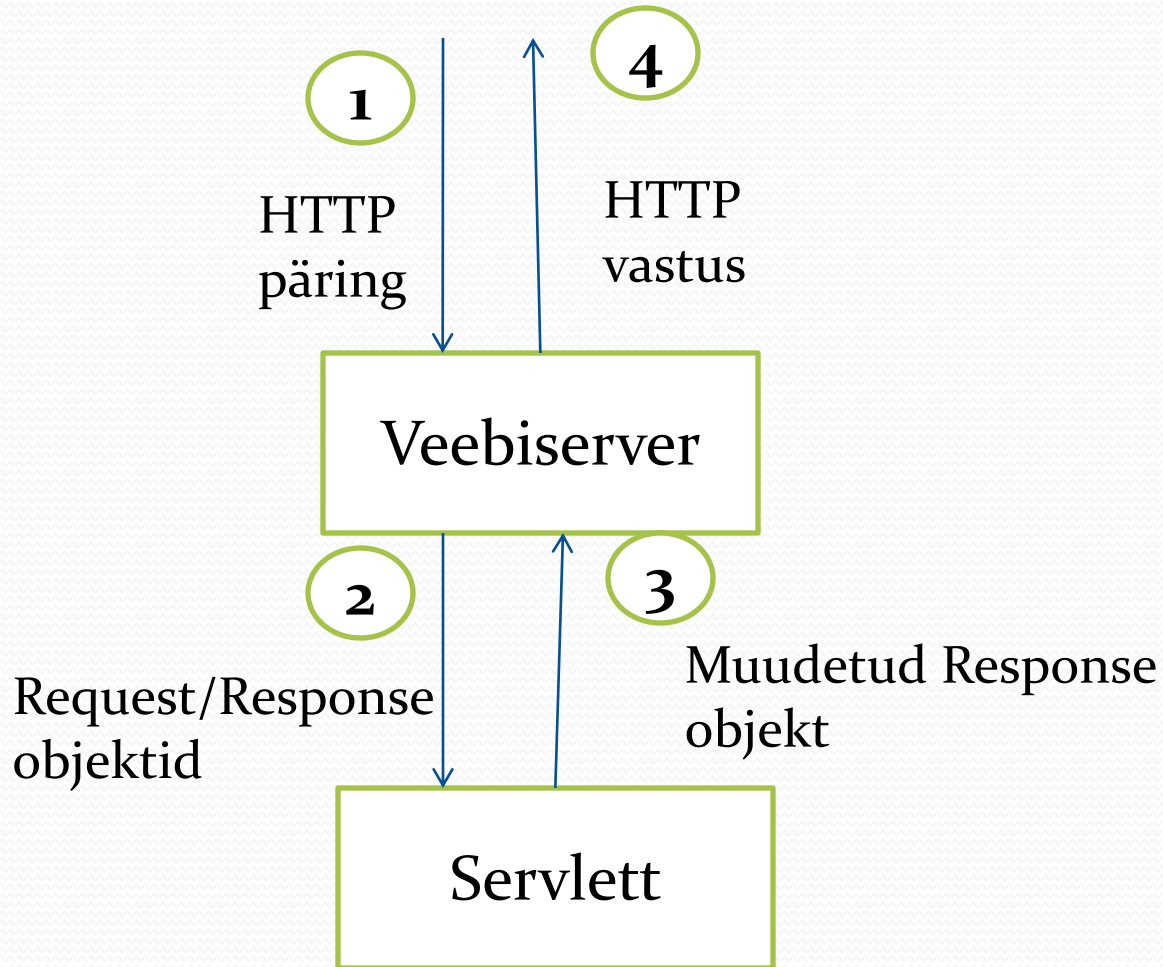
- Java kooditükid, mis töötavad serveris.
- Käivitatakse *konteineris*
- Toetavad klient-server suhtlust
 - Tavaliselt üle HTTP

Kus asuvad servletid?

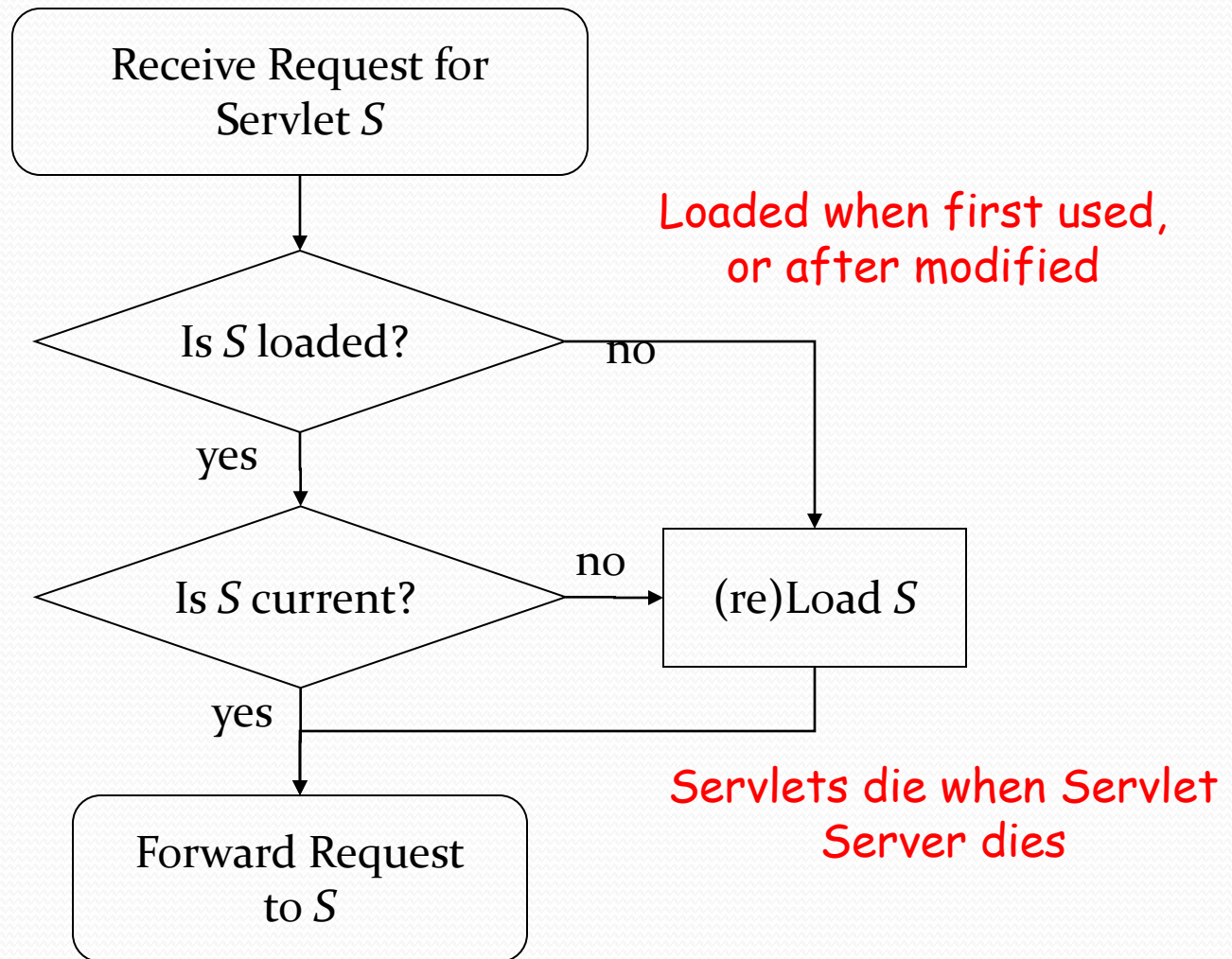


Tomcat = Web Server + Servlet Server

Java servletid



Servleti kasutamine



Servlet ja JSP mootorid (vabavara)

- Apache Tomcat

- <http://jakarta.apache.org/tomcat/>

- Allaire/Macromedia JRun

- <http://www.macromedia.com/software/jrun/>

- New Atlanta ServletExec

- <http://www.servletexec.com/>

- Gefion Software LiteWebServer

- <http://www.gefionsoftware.com/LiteWebServer/>

- Caucho's Resin

- <http://www.caucho.com/>

.....

<http://www.coreservlets.com/>

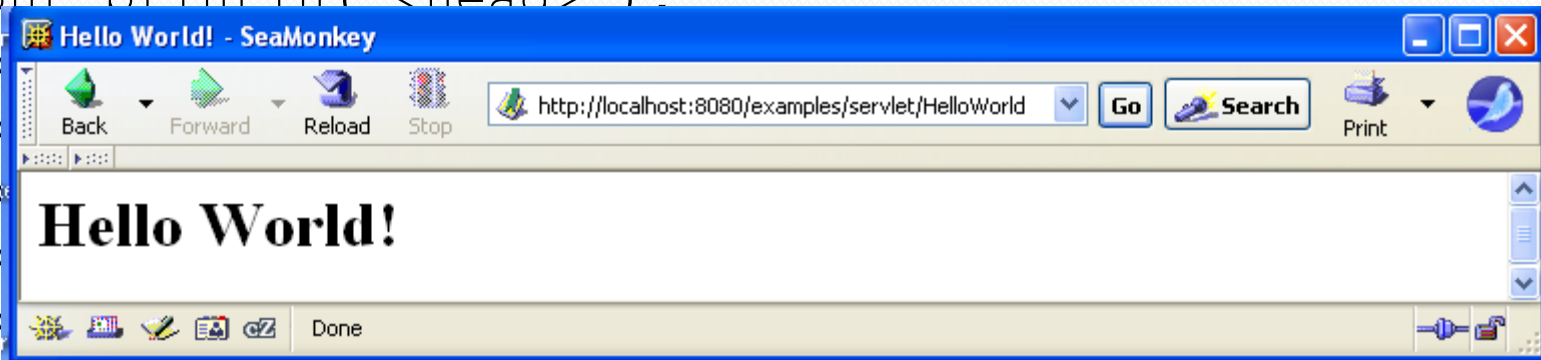
Näide HelloWorld

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws
        IOException, ServletException {
response.setContentType("text/html");
PrintWriter out = response.getWriter();
out.println("<html>");
out.println("<head>");
out.println("<title>Hello World!</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Hello World!</h1>");
out.println("</body>");
out.println("</html>");
    } }

```


Näide HelloWorld

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws
        IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("</html>");
    }
}
```



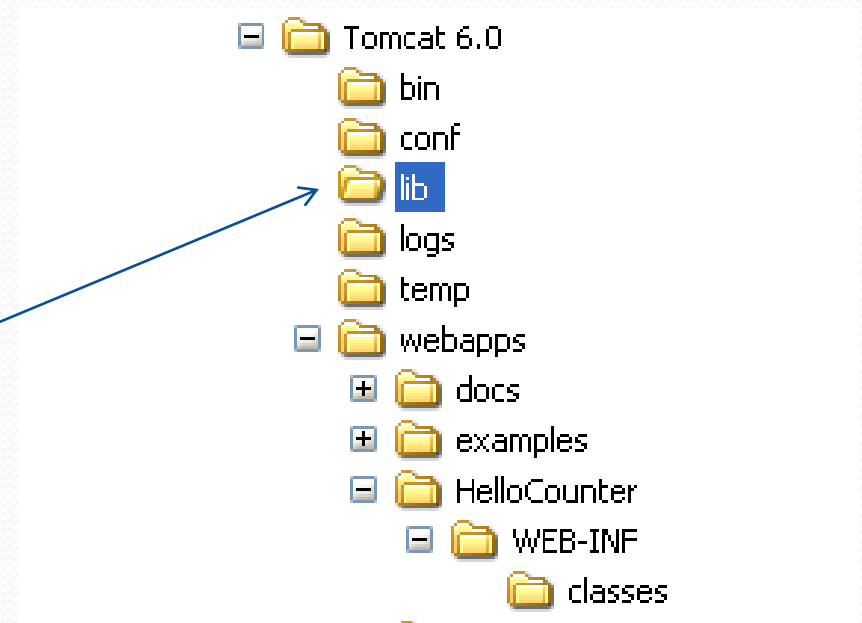
Tomcat 6

Windows keskkond

Servleti kompileerimine:

Lisada CLASSPATH-i
install/lib

javax.servlet-api.jar



Tomcat 6 kataloogipuu

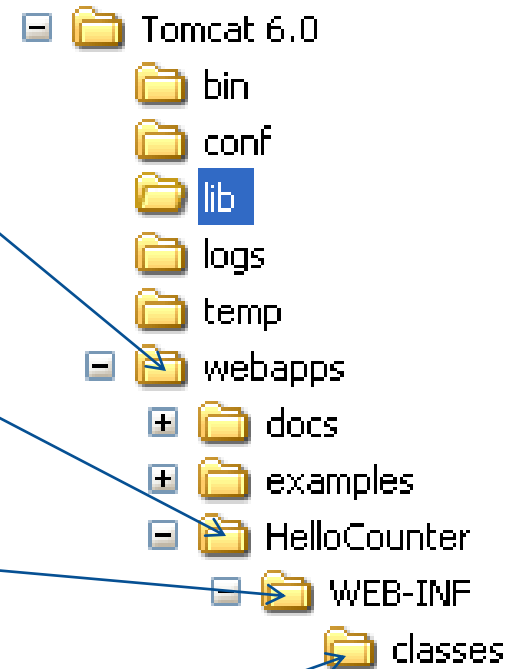
Siia alla oma rakendused

Lua oma rakenduse kataloog

Lua alamkataloog WEB-INF

Lua alamkataloog classes

Windows keskkond



Tomcat 6 kataloogipuu (jätk)

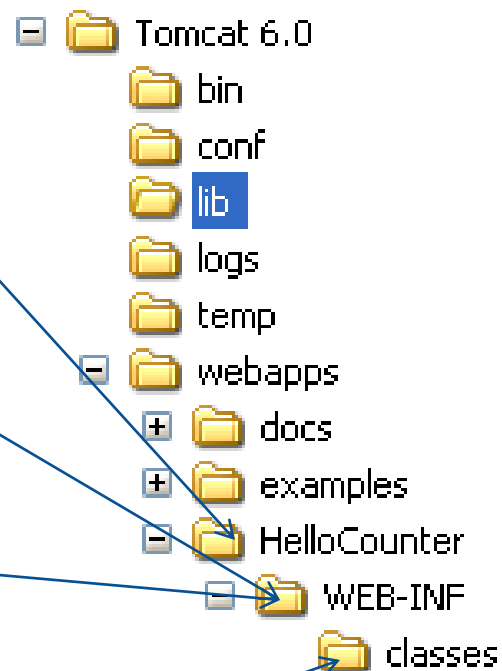
Windows keskkond

Rakenduse staatiline sisu

Dünaamiline sisu on WEB-INF-is

web.xml on WEB-INF-is

Servlettide class failid on siin



```
public void doGet (HttpServletRequest request,  
                  HttpServletResponse response)
```

Tavaliselt meetodis `doGet ()` tuleb teha järgmised töid:

1. Seada HTTP Content-Type. Tavaliselt on see `text/html`, võib anda ka kodeeringu, nt `ISO-8859-1`.
2. Luua objekt `PrintWriter` kasutades objekti `HttpServletResponse` meetodit `getWriter`.
3. Väljastada kooskõlaline HTML dokument kasutades `PrintWriter` objekti.
4. Sulgeda objekt `PrintWriter`.

Servlet API

<http://tomcat.apache.org/tomcat-5.5-doc/servletapi/index.html>

Servleti võrdlus Java programmiga:

- Servlett ei sisalda `main` meetodit, selle lisab (Tomcat) server;
- Servletis on meetod `doGet ()` või `doPut ()` või mõni teine `doXXX ()` ;
- Servlett ei suhtle vahetult kasutajaga, kogu sisend on objektis `HttpServletRequest` ja kogu väljund kirjutatakse objekti `HttpServletResponse`;
- Servleti põhiväljundiks on `HTML`.

Põhilised sammud servleti käivitamiseks:

- Kompileerida servlett.
- Kopeerida tulemus *.class* fail serverisse vastavasse kausta.
- Käivitada server.
- Registreerida servlett failis `web.xml`
- Navigeerida servletile vastava URL-ga, nt

`http://localhost:8080/examples/servlet/Hello`

URL ja servleti vastavus seatakse failis `web.xml`

web.xml

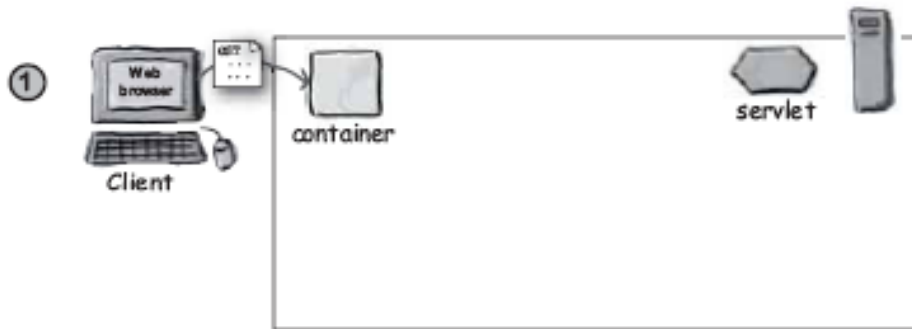
... .

```
<servlet>
  <servlet-name>HW</servlet-name>
  <servlet-class>
    HelloWorld
  </servlet-class>
</servlet>
```

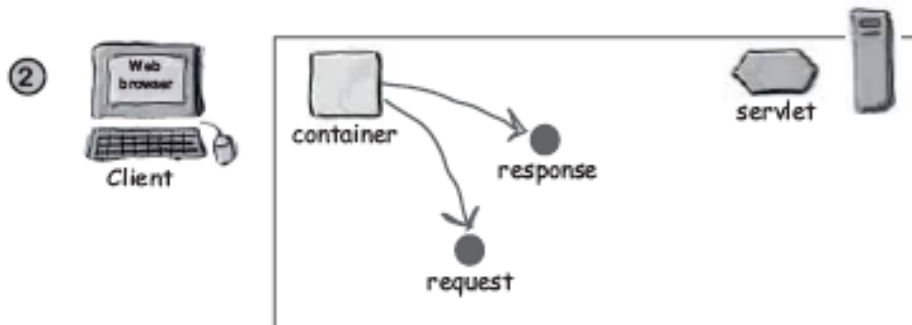
... .

```
<servlet-mapping>
  <servlet-name> HW</servlet-name>
  <url-pattern>/servlet/Hello</url-pattern>
</servlet-mapping>
```

DEMO:
Näide dünaamilisest servletist
HelloCounter.java

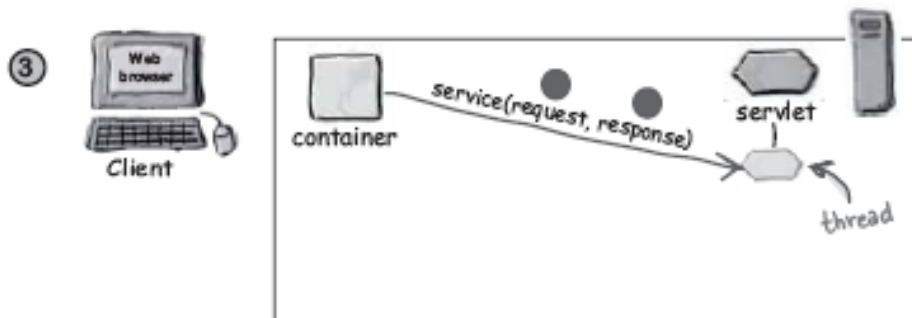


User clicks a link that has a URL to a servlet.

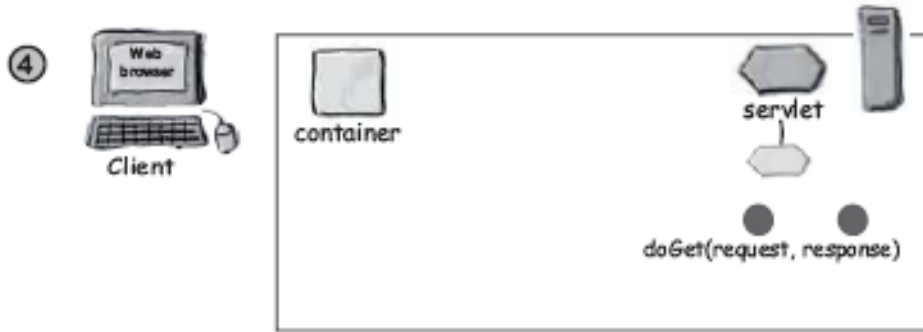


The Container "sees" that the request is for a servlet, so the container creates two objects:

- 1) `HttpServletResponse`
- 2) `HttpServletRequest`

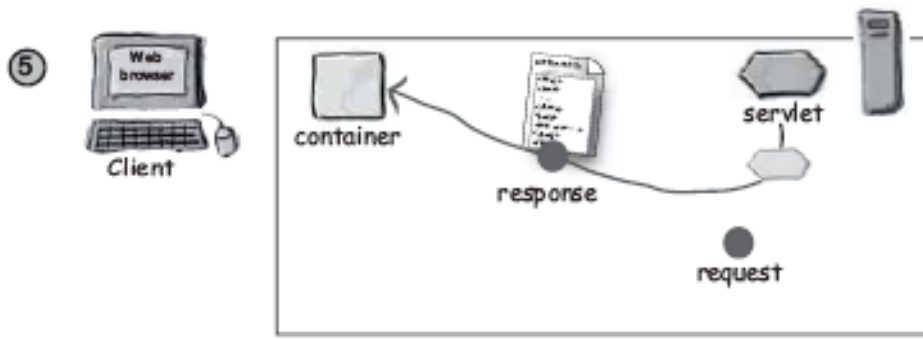


The Container finds the correct servlet based on the URL in the request, creates or allocates a thread for that request, and calls the servlet's `service()` method, passing the request and response objects as arguments.

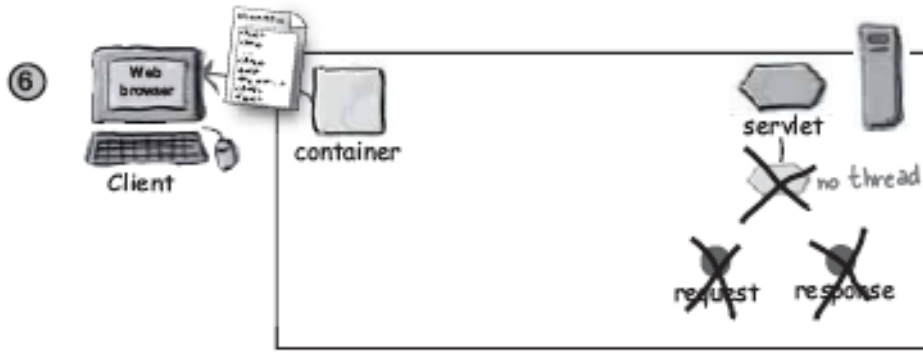


The service() method figures out which servlet method to call based on the HTTP Method (GET, POST, etc.) sent by the client.

The client sent an HTTP GET request, so the service() method calls the servlet's doGet() method, passing the request and response objects as arguments.



The servlet uses the response object to write out the response to the client. The response goes back through the Container.



The service() method completes, so the thread either dies or returns to a Container-managed thread pool. The request and response object references fall out of scope, so these objects are toast (ready for garbage collection).

The client gets the response.

Servleti elutsükkel

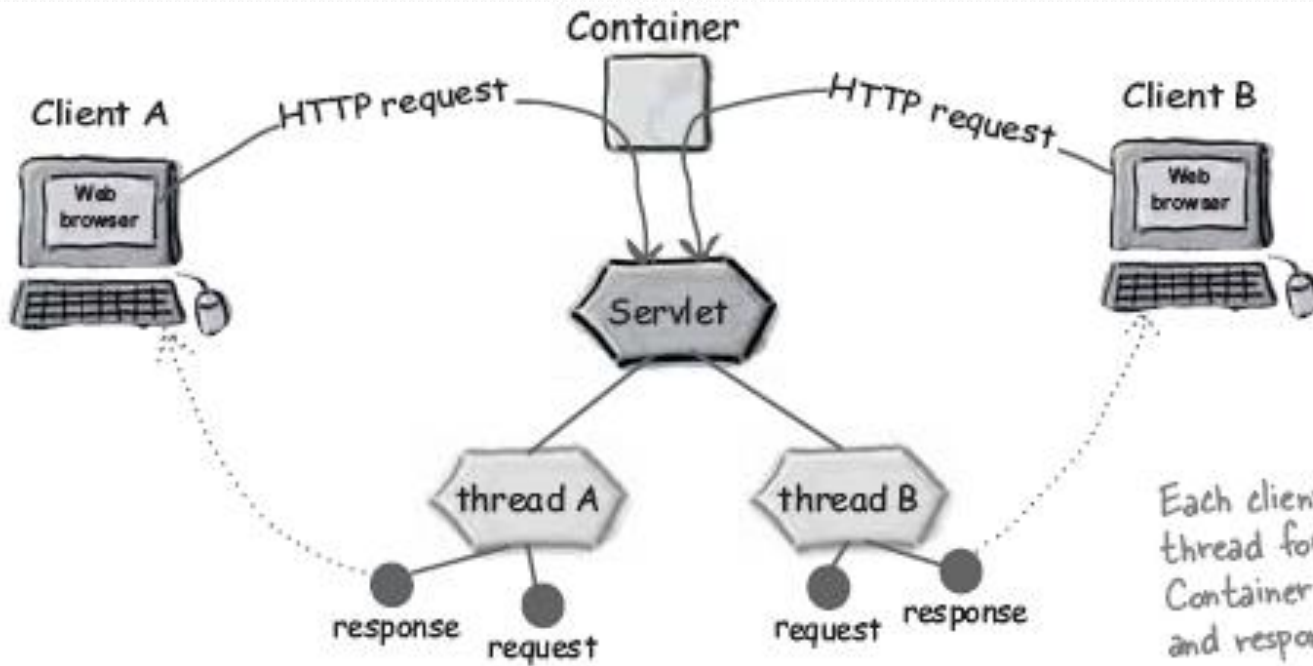
- `init`
 - Täidetakse servleti esimesel laadimisel.
- `service`
 - Kutsutakse välja uue lõime poolt iga päringu korral. Pöördumine `doGet`, `doPost`, jt meetodite poole. Seda meetodit ei tohi üle katta!
- `doGet`, `doPost`, `doXxx`
 - Töödeldakse päringuid `GET`, `POST` jt.
 - Need meetodid tuleb üle katta.
- `destroy`
 - Kutsutakse välja siis kui server kustutab servleti isendi.

Miks ei tohiks üle katta meetodit `service`

- Võib hiljem lisada teisi teenuseid: `doPut`, `doTrace`, jne.
- `service` meetod annab automaatse toetuse järgmistele päringutele:
 - HEAD
 - OPTIONS
 - TRACE
 - ..
- Alternatiiv: `doPost` kutsub välja `doGet`

Servleti initsialiseerimine

- Serveri seadistamine
 - Nt andmebaasi ühenduspuhvrite seadistamine
- `ServletConfig.getInitParameter` - parameetrite lugemiseks
- Seada `web.xml`
 - `.../WEB-INF/web.xml`
 - Paljudel serveritel on liidesed `web.xml` loomiseks



Each client gets a separate thread for each request, and the Container allocates new request and response objects.

Andmete edastamine servletile

- **GET** päring edastab andmed URL-is

`?nimi=väärtus&nimi=väärtus&nimi=väärtus`

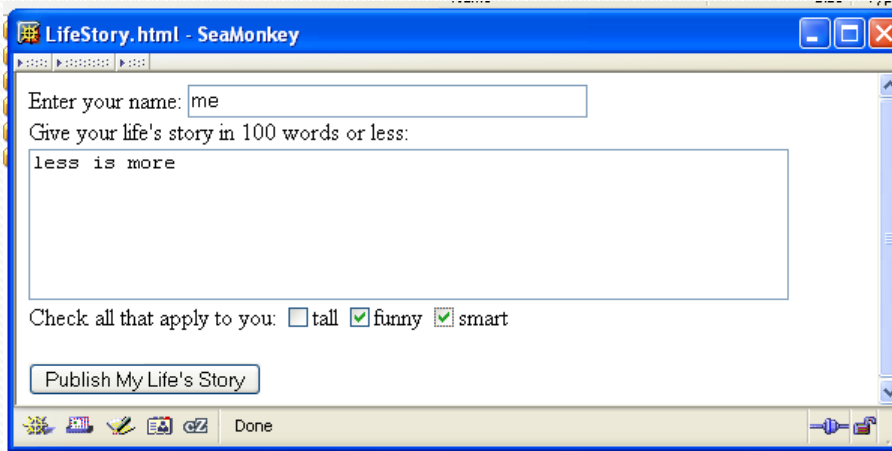
- Tühikud asendatakse **+** märkidega
- Teised erimärgid (<, >, &, ' , „, ..) kodeeritakse;
- nt jutumärk “ -> **%22**

`?arg="a <script & >String"&color=red`

`arg=%22a%20%3Cscript%20&%20%20%3EString%22&color=red`

- Parameetrite nimed võivad korduda eri väärtustega
- **POST** päring edastab andmed sama süntaksi abil, need asuvad päringu kehas
- Andmete kättesaamiseks kasutatakse `HttpServletRequest` meetodeid

Andmete edastamine servatile



SeaMonkey 1.1.14:

```
name=me&lifestory=less+is+more&boxgroup1=funny&boxgroup1=smart&do  
it=Publish+My+Life%27s+Story
```

Firefox 3.06

```
name=me&lifestory=less+is+more&boxgroup1=funny&boxgroup1=s  
mart&doit=Publish+My+Life's+Story
```

IE7

```
name=me&lifestory=more+is+less&boxgroup1=funny&boxgroup1=smart&do  
it=Publish+My+Life%27s+Story
```


Andmete edastamine servletile

Parameetri väärtus sõltub märgisest, nt:

Märgis

input/text, input/password, textarea
input/checkbox, input/radio, input/submit,

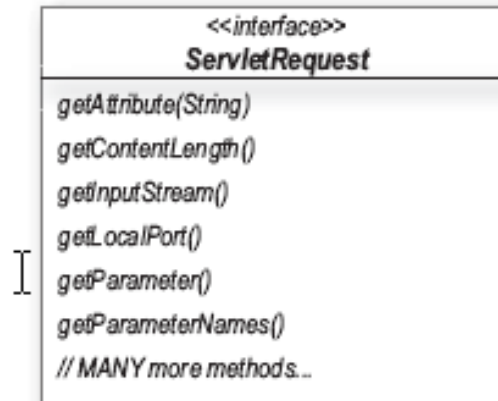
input/image, button/submit
input/hidden

Väärtus

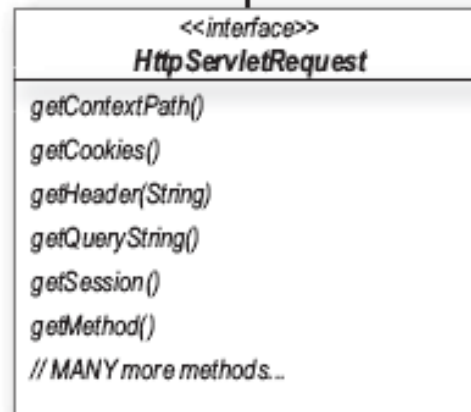
Elemendis olev tekst.
String, mis on seotud *value*
atribuudiga.

Element peab olema valitud.
String, mis seotud *value*
atribuudiga.

ServletRequest interface
(javax.servlet.ServletRequest)



HttpServletRequest interface
(javax.servlet.http.HttpServletRequest)



HttpServletRequest meetodeid:

```
String getQueryString()
```

Tagastab päringustringi kodeeritud kujul.

```
Enumeration getParameterNames()
```

Tagastab String Enumeration-i parameetrite nimedest, mis on URL-is, dekodeeritud kujul.

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

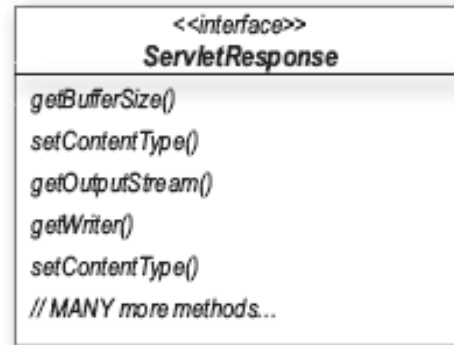
Tagastab parameetri väärtuse dekodeeritud kujul.

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

Tagastab String massiivi antud parameetri väärtuste jaoks dekodeeritud kujul (parameeter võib päringustringis olla mitu korda).

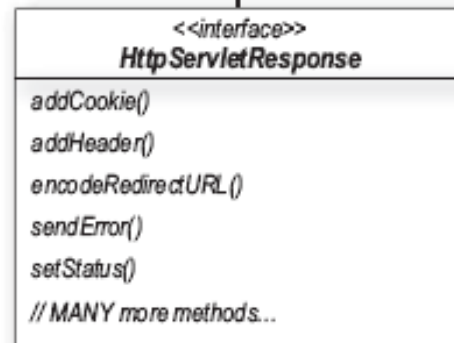
ServletResponse interface

(javax.servlet.ServletResponse)



HttpServletResponse interface

(javax.servlet.http.HttpServletResponse)



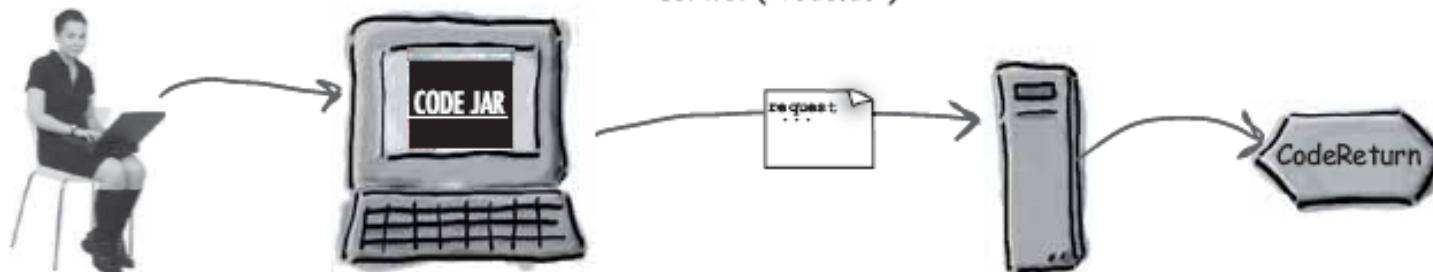
HTML genereerimine

- Seada päis Content-Type
 - Kasutada `response.setContentType()`
- HTML väljastamine
 - Kindlasti lisada DOCTYPE
- Kasutada HTML valideerimisteenust
 - <http://validator.w3.org/>
 - <http://www.htmlhelp.com/tools/validator/>

- ① Diane is desperate to download the JAR of code for the book she's using to learn servlets and JSPs. She navigates to the book's website and clicks the "code jar" link, which refers to a servlet named "Code.do".

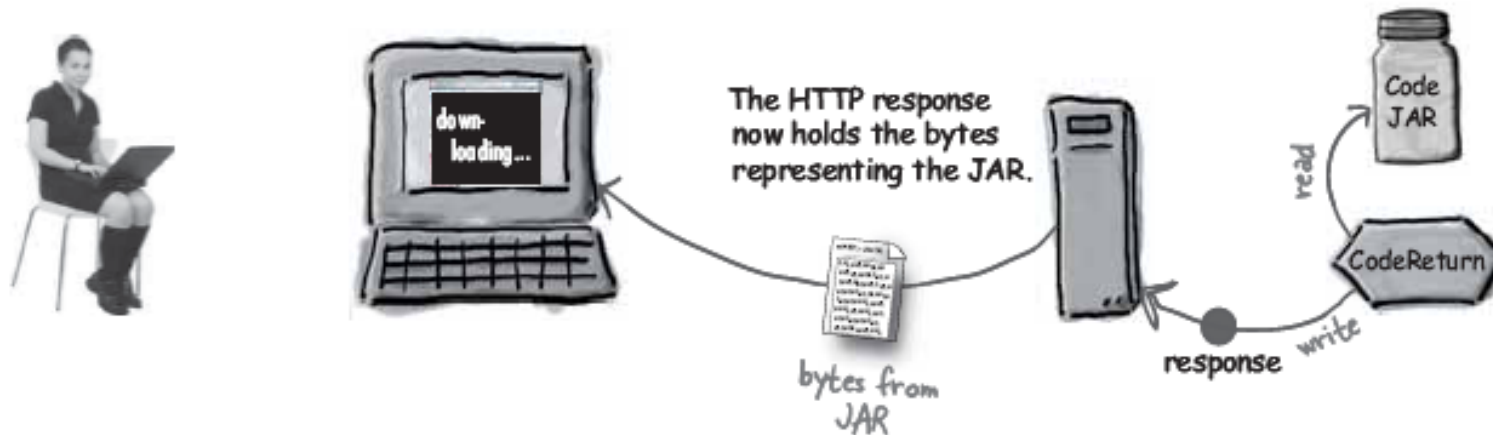
Browser sends an HTTP request to the server with the name of the requested servlet ("Code.do")

The Container sends the request to the CodeReturn servlet (mapped to the name "Code.do" in the DD) for processing.



- ② The JAR starts downloading onto the client's machine.
Diane is pleased.

The CodeReturn servlet gets the bytes for the JAR, then gets an output stream from the response, and writes out the bytes representing the JAR.



```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CodeReturn extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response) throws
        IOException, ServletException {
response.setContentType("application/jar");
    ServletContext ctx = getServletContext();
    InputStream is =
    ctx.getResourceAsStream("/bookCode.jar");
    int read = 0;
    byte[] bytes = new byte[1024];
    OutputStream os = response.getOutputStream();
    while((read =is.read(bytes)) != -1){
        os.write(bytes, 0, read);
    }
    os.flush();
    os.close();
} }
```

Turvalisusest

- servlett peab analüüsima kasutajate poolt sisestatud infot
- Väljastatavas dokumendis tekstilise info sees erisümbolid (< > &) asendada vastavate olemitega.
- Atribuutide väärtuste sees asendada < > & ' „ asendada olemitega.
- cross-site-scripting attack

<http://www.cert.org/advisories/CA-2000-02.html>

Lihtsamast keerulisemaks

- Siiani vaatlesime lihtsaid servlette, kuid see pole piisav, et kirjutada *kasulikke* servlette
 - Vajalikud oskused:
 - Kasutada konfiguratsiooni infot
 - Autentida kasutajaid
 - Jälgida kasutajate seansse
 - Säilitada infot seansside vahel
 - Infovahetus servlettide vahel
 - But remember: The most difficult program in any language is *Hello World!*

Seansid

Session tracking – on selle jälgimine, mis on juhtunud enne seda “vestlust”

- HTTP on olekuta, ta ei garanteeri seda
- Seda tuleb servlettides ise garanteerida

Lahendused seansi jälgimiseks

- **Cookies** on väikesed failid, mida servlett saab paigutada kliendi masinasse ja hiljem tagasi lugeda
- **URL ülekirjutamine**: Võib lisada unikaalse ID URL lõppu kasutaja identifitseerimiseks
- **Peidetud <form> väljad**: võib kasutada unikaalse ID hoidmiseks
- Java **Session Tracking API**

Seansid - Session Tracking API

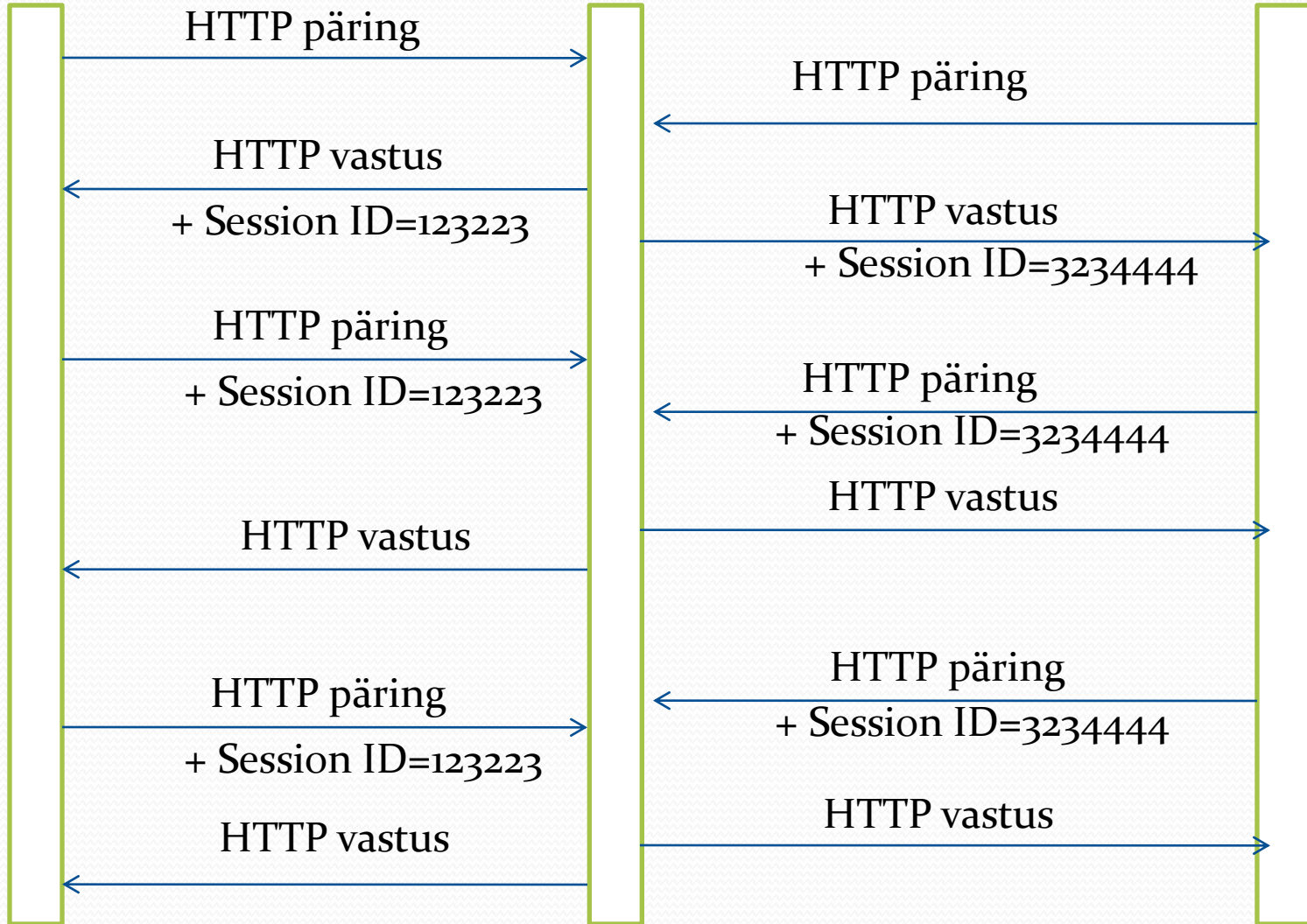
- Seansi ID
- Kõikide päringute hulka, mis toimuvad ühe ja sama seansi IDga, nimetatakse kasutaja seansiks.
- *HttpServletRequest* meetod *getSession()* loob objekti *HttpSession* kui HTTP päring ei sisalda seansi ID
- *HttpSession* meetod *isNew()* annab tõeväärtuse, kas seanss on just äsja algatatud.

Klient 1

Server

Klient 2

Aeg



Seansid

```
HttpSession session =  
request.getSession();  
    if (session.isNew()) {  
        visits++;  
    }
```

DEMO VisitorCounter.java – eri brauseritega

Seansid

HttpSession meetodeid:

void **setAttribute** (String name, Object value)

Object **getAttribute** (String name)

java.util.Enumeration **getAttributeNames** ()

String **getId** ()

DEMO Greeting.java

doGet:

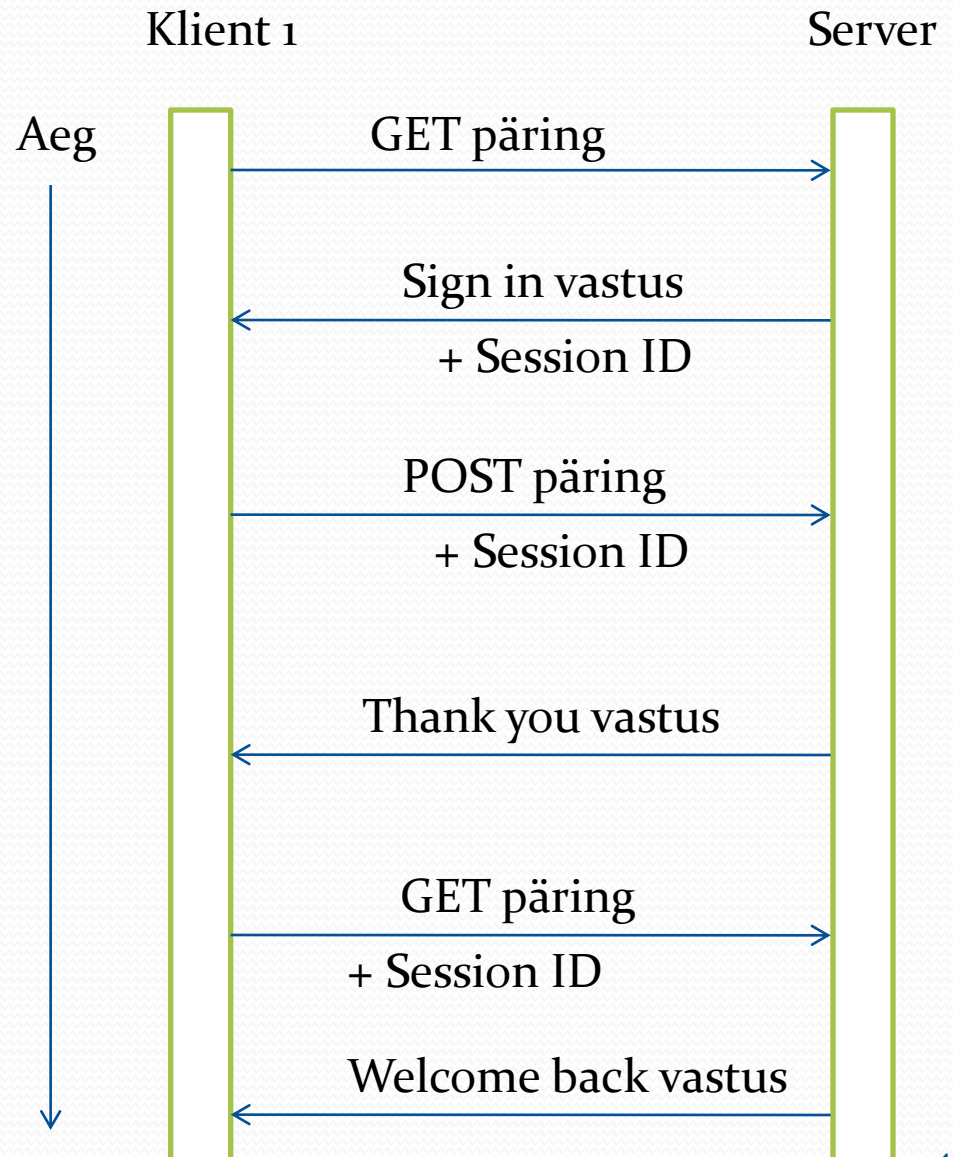
```
HttpSession session = request.getSession();
String signIn = (String)session.getAttribute("signIn");
if (session.isNew() || (signIn == null)) {
    printSignInForm(servletOut, "Greeting");
}
else {
    printWelcomeBack(servletOut, signIn);
}
```


Seansid

doPost:

```
String signIn = request.getParameter("signIn");
HttpSession session = request.getSession();
if (signIn != null) {
    printThanks(servletOut, signIn, "Greeting");
    session.setAttribute("signIn", signIn);
}
else {
    printSignInForm(servletOut, "Greeting");
}
```

Seansid



Seansi katkestamine

HttpSession liidesel on meetodid

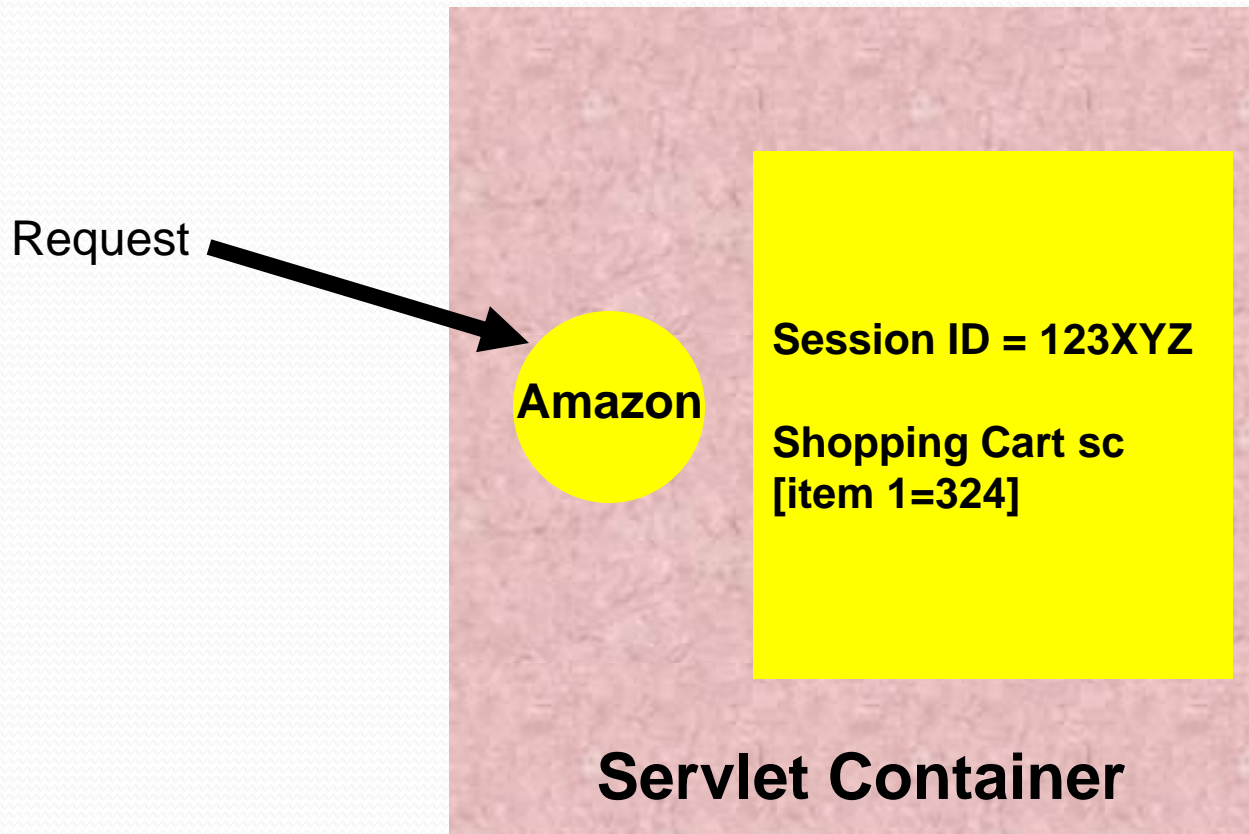
```
void setMaxInactiveInterval(int interval)
```

Määrab aja sekundites kasutaja päringute vahel, pärast mida seanss loetakse lõpetatuks (kutsutakse välja `invalidate()` meetod).

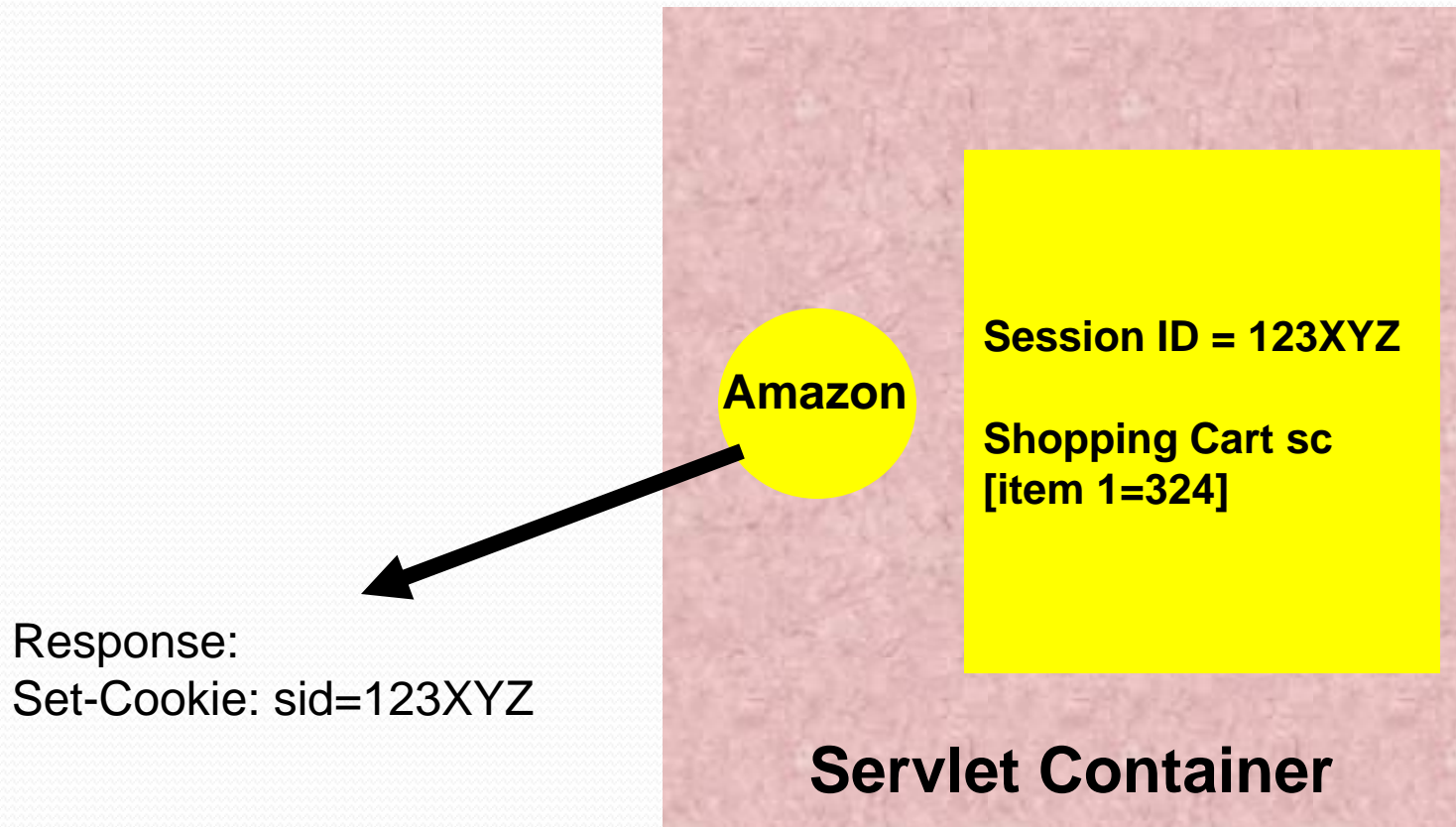
```
void invalidate()
```

Muudab seansiobjekti kehtetuks.

Seansi jälgimine



Seansi jälgimine



Seansi jälgimine

Request:
Set-Cookie: sid=123XYZ



Amazon

Session ID = 123XYZ

**Shopping Cart sc
[item 1=324]**

Servlet Container

Seansi jälgimine

Request:
Set-Cookie: sid=123XYZ



Amazon

Session ID = 123XYZ

**Shopping Cart sc
[item 1=324
item 2=115]**

Servlet Container

Cookies

- **Cookie** on väike tekstiline info, mida server saadab kliendile HTTP vastuse osana
- Cookies ei ole oht turvalisusele
- Cookies *võivad olla* oht privaatsusele
 - Cookiesid võib kasutada reklaamide kohandamiseks
 - Servlett oskab cookiesid lugeda

Cookies

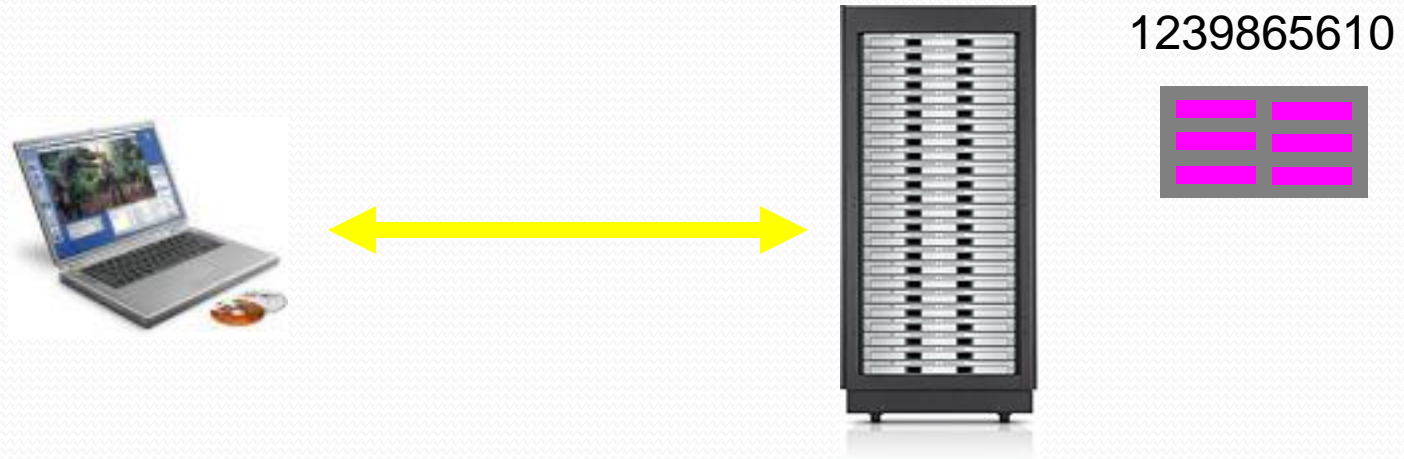


1239865610

```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

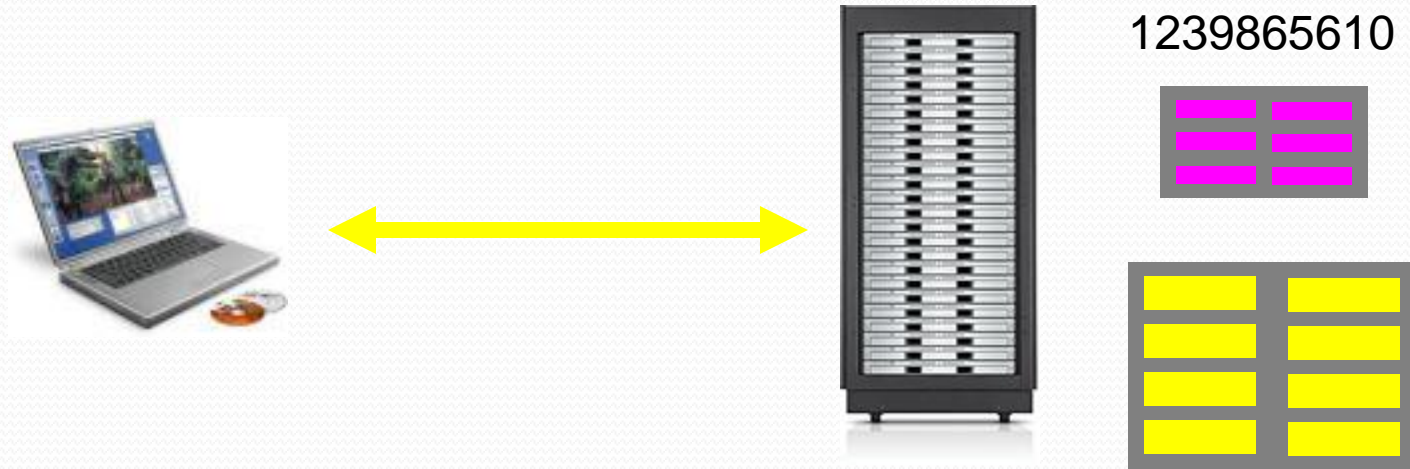
Cookies



```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

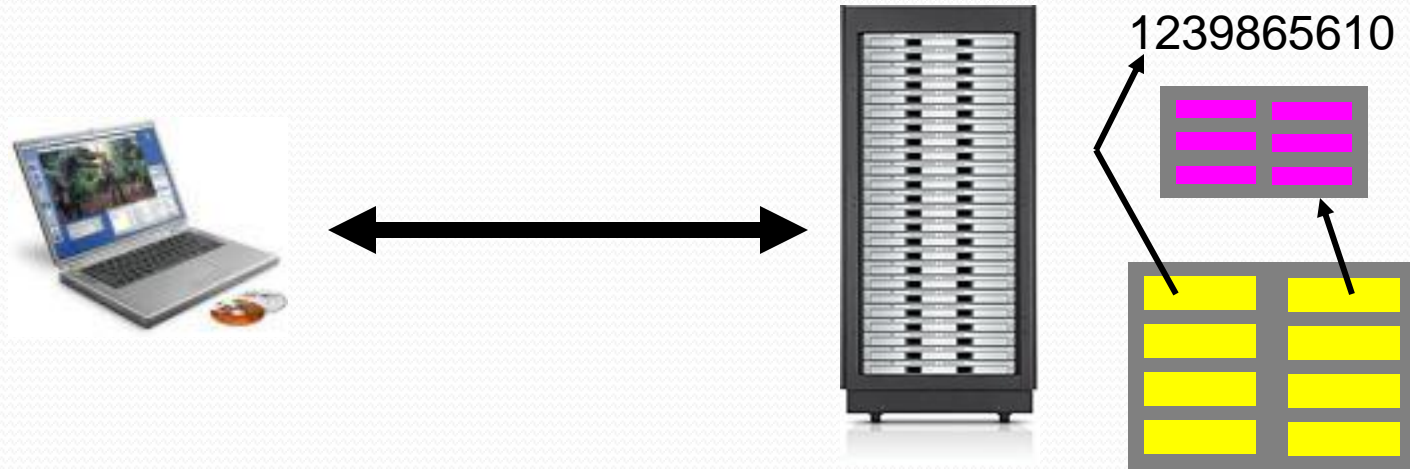
Cookies



```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

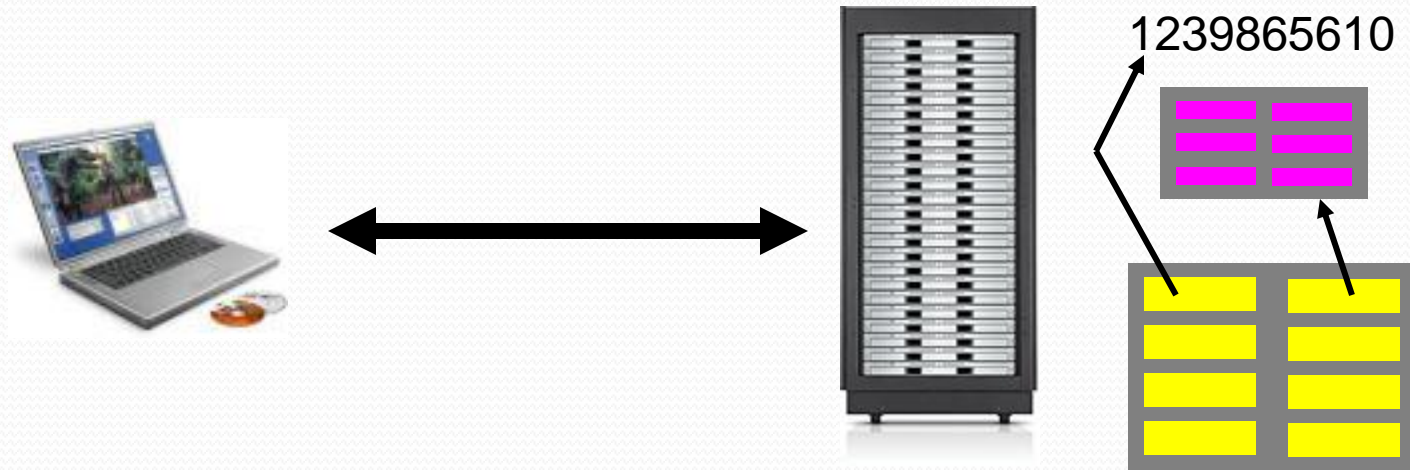
Cookies



```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

Cookies

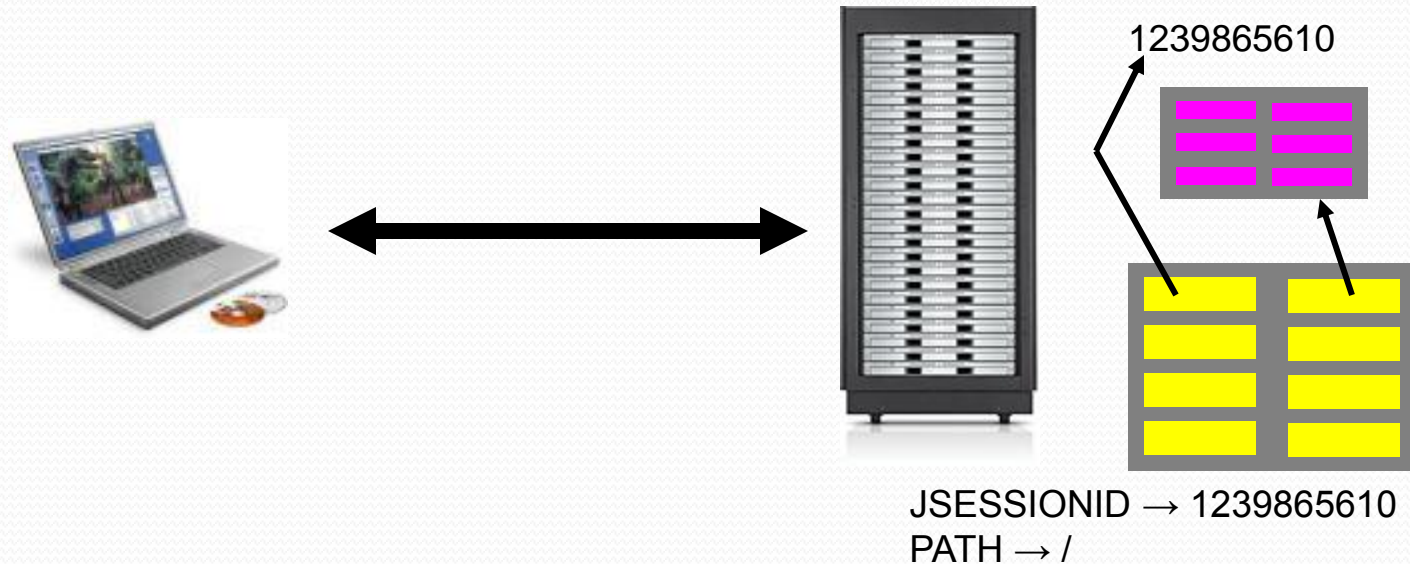


JSESSIONID → 1239865610

```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

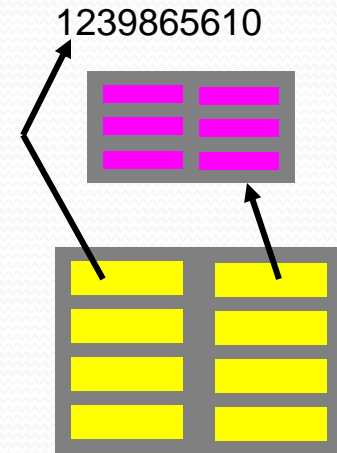
Cookies



```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

Cookies



Set-Cookie: JSESSIONID=1239865610; path=/;

```
String sID = makeUniqueString();  
Hashtable sessionInfo = new Hashtable();  
Hashtable globalTable = findTableStoringSessions();  
globalTable.put(sID, sessionInfo);  
Cookie sessionCookie = new Cookie("JSESSIONID", sID);  
sessionCookie.setPath("/");  
response.addCookie(sessionCookie);
```

Kelly Shaw Stanford

Cookies



Cookie: JSESSIONID=1239865610;

```
// On request
```

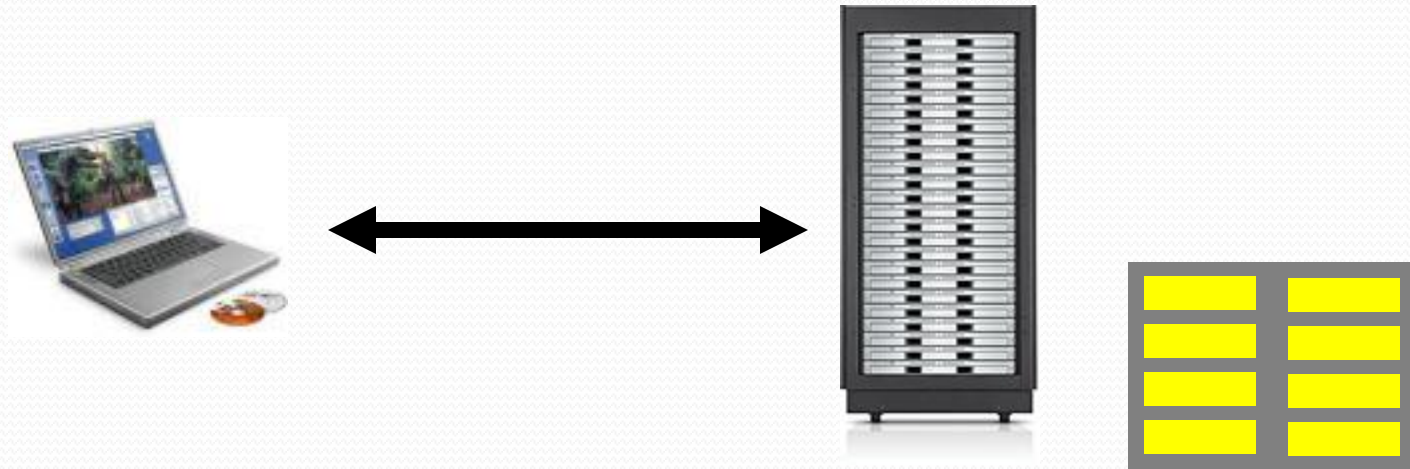
```
String sID = request.getCookie("JSESSIONID");
```

```
Hashtable globalTable = findTableStoringSessions();
```

```
Hashtable sInfo = (Hashtable) globalTable.get(sID);
```

Kelly Shaw Stanford

Cookies

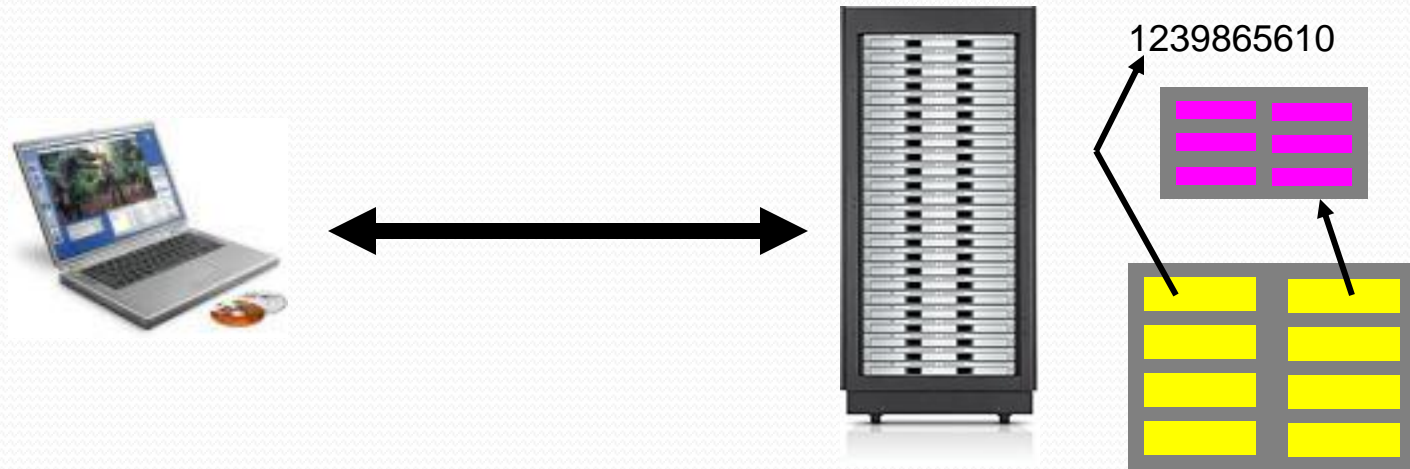


Cookie: JSESSIONID=1239865610;

```
// On request  
String sID = request.getCookie("JSESSIONID");  
Hashtable globalTable = findTableStoringSessions();  
Hashtable sInfo = (Hashtable) globalTable.get(sID);
```

Kelly Shaw Stanford

Cookies



Cookie: JSESSIONID=1239865610;

```
// On request  
String sID = request.getCookie("JSESSIONID");  
Hashtable globalTable = findTableStoringSessions();  
Hashtable sInfo = (Hashtable) globalTable.get(sID);
```

Kelly Shaw Stanford

Cookies

Klassi `Cookie` meetodeid:

Cookie (`String name`, `String value`)
Konstruktor.

`String` **getName** ()
Tagastab `Cookie` nime.

`String` **getValue** ()
Tagastab `Cookie` väärtuse.

`Void` **setMaxAge** (`int seconds`)
Seab `Cookie` aegumise sekundites.

Cookies

DEMO: CookieCounter.java – ei ole mõistlik järele teha!

doGet ()

```
private static final int oneYear = 60*60*24*365;
...
int count = 0;
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for (int i=0; (i<cookies.length) && (count==0); i++) {
        if (cookies[i].getName().equals("COUNT")) {
            count = Integer.parseInt(cookies[i].getValue()); } } }
count++;
Cookie cookie = new Cookie("COUNT",
    new Integer (count).toString());
cookie.setMaxAge(oneYear);
response.addCookie(cookie);
...
```

URL ülekirjutamine

- HTTP vastuses üle kirjutada `URLid`, et need sisaldaks `JSESSIONID` (`href`, `action`, ..)
- `HttpResponse` objekti meetod `encodeURL(String url)` kirjutab seansi id `URLi` järele, nt
- `URLEncodedGreeting;jsessionid=CB3B1DE19418D6C1E0E790E3CE3494BA`

- Päringu `URList` eraldada `ID`

URL ülekirjutamine

Klassil `HttpServletRequest`-il on meetodid

```
isRequestedSessionFromCookie()
```

ja

```
isRequestedSessionFromURL()
```

DEMO `URLEncodedGreeting.java`, `Greeting.java`

keela brauseril cookied

Peidetud <form> väljad

- `<input type="hidden" name="sessionID" value="...">`
- Eelised:
 - Nõuab vähe teadmisi: Parameetrite kirjutamine ja lugemine
- Puudused:
 - Ei hoita alles seansside vahel, ei saa hoida kasutaja kohta püsivandmeid
 - Iga vastusena tagastatavHTML tuleb dünaamiliselt genereerida

```
<form method="POST" action="/exec/obidos/handle-buy-box=/  
ref=bp_add/103-4591077-2490203">
```

```
<input type="hidden" name="colid" value="">  
<input type="hidden" name="template-name" value="">  
<input type="hidden" name="store-name" value="gateway">  
<input type="hidden" name="maw" value="1">  
<input type="hidden" name="coliid" value="">
```

```
<input type="hidden" name="dropdown-selection"  
value="default-address">
```

```
<table border="0" width="100%" cellspacing="0"  
cellpadding="6">
```


HttpServletRequest meetodeid

`String getRemoteAddr()`

Tagastab pärigu saatja arvuti IP

`String getRemoteHost()`

Tagastab pärigu saatja arvuti hosti

`String getProtocol()`

Tagastab suhtlusprotokolli (HTTP/1.1)

`boolean isSecure()`

`StringBuffer getRequestURL()`

HttpServletResponse meetodeid

`void setHeader(String name, String value)`
Seab vastuse päisevälja.

`void setBufferSize(int size)`
Seab väljundpuhvri suuruse.

DEMO
PrintHttpInfo

`void setStatus(int statusCode)`
Seab HTTP vastuse staatuse (200 OK).

`void sendError(int statusCode, String msg)`
Seab vastuse seisundikoodiks veakoodi (algab 4 või 5-ga) ja kehas HTML vealeht, mille sees sõnum msg.

`void sendRedirect(String url)`
Põhjustab HTTP vastuse 307 (ajutine ümbersuunamine), klient peab saatma uue päringu.

Java: Enumerator

Näide:

```
Enumeration e = myVector.elements();  
while (e.hasMoreElements()) {  
    System.out.println(e.nextElement());  
}
```

Java: Andmed sõnest

- Parameetrite väärtused on `String` kujul
- Sageli esitavad need numbreid, st on vaja teisendada
 - `int n = new Integer(param).intValue();`
 - `double d = new Double(param).doubleValue();`
 - `byte b = new Byte(param).byteValue();`
 - Analoogiliselt `short`, `float`, ja `long`
 - Võib tekkida erind `NumberFormatException`, mis on `RuntimeException` alamklass
 - `boolean p = new Boolean(param).booleanValue();`
- Kuid:
 - `char c = param.charAt(0);`