Java és web programozás

Kovács Kristóf, Rimay Zoé

Budapesti Műszaki Egyetem

2013. november 13.

9. Előadás

Ismétlés

- Üres szerver indítása:

```
HttpServer server = HttpServer.createSimpleServer();
try {
    server.start();
    System.out.println("Press any key to stop the server");
    System.in.read();
} catch (Exception e) {
    System.err.println(e);
}
```

- Nem teljesen ezt fogjuk használni, csak a *try* blokkon belüli részt.

Ismétlés 2

- Egy oldal hozzáadása:

```
...
HttpServer server = HttpServer.createSimpleServer();
server.getServerConfiguration().addHttpHandler(
    new HttpHandler() {
        public void service(Request request,
                Response response) throws Exception {
            final String respString = "Elso oldal";
            response.setContentType("text/plain");
            response.setContentLength(respString.length(
            response.getWriter().write(respString);
        }
    },
    "/time");
try {
...
```

- De ez elég csúnya, így inkább külön osztályokat írtunk az

Ismétlés 3

- *main*ben:

```
HttpServer server = new HttpServer();
NetworkListener nl = new NetworkListener(
    "main-listener", "localhost", 1234);
server.addListener(nl);

server.getServerConfiguration().addHttpHandler(
     new IndexPage(), "/time");
```

- Ez által, ha a *IndexPage* osztályt úgy definiáljuk, hogy a *HttpHandler* osztály az őse, akkor minden renden lesz.

Ismétlés 4

```
import org.glassfish.grizzly.http.server.*;

public class IndexPage extends HttpHandler  {

    @Override
    public void service(Request request, Response response)
            throws Exception {
        final String respString =
            HtmlUtilities.createSimplePage(
                "Oldal", "Főoldal");
        response.setContentType("text/html");
        response.setContentLength(respString.length());
        response.getWriter().write(respString);
    }
}
```

Request ismétlés

```
<form action="tovabb" method="get">
  <label>User: <input type="text"
                      name="username" value=""></label>
  <label>Password: <input type="password"
                      name="password"></label>
  <button type="submit">Login</button>
</form>
```

Request ismétlés

```
<form action="tovabb" method="get">
  <label>User: <input type="text"
                      name="username" value=""></label>
  <label>Password: <input type="password"
                      name="password"></label>
  <button type="submit">Login</button>
</form>
```

- Ide ugrik:

```
/tovabb?username=Tofi&password=kutya
```

Request ismétlés 2

- A paramétereket a következő módon tudjuk lekérni:

```
request.getParameter("parameterNeve");
```

- Tehát a mostani esetünkben ezt irhatnánk pl:

```
String user = request.getParameter("username");
String pass = request.getParameter("password");
```

- A paraméterek értékét mindig *String*ként kapjuk meg. Még akkor is ha számokat írtunk az adott mezőbe.
- Lekérhetjük a teljes query Stringet (? utáni részt) a *request.getQueryString()* metódussal.

Sütik

- Képzeljük magunkat a webszerver helyébe.
- Másodpercenként akár több 100 felhasználó küld nekünk kéréseket, ezekre válaszokat kell küldenünk.
- Viszont sokszor kell olyan adatot küldenünk amihez nem feltétlen férhet hozzá mindenki. Ekkor el kell döntenünk, hogy épp kivel *beszelünk*.

Sütik

- Képzeljük magunkat a webszerver helyébe.
- Másodpercenként akár több 100 felhasználó küld nekünk kéréseket, ezekre válaszokat kell küldenünk.
- Viszont sokszor kell olyan adatot küldenünk amihez nem feltétlen férhet hozzá mindenki. Ekkor el kell döntenünk, hogy épp kivel *beszelünk*.
- Erre vannak kitalálva a sütik.
- A felhasználó gépére menthetünk kevés adatot, amivel azonosítani tudjuk, és sok más dolgot is megvalósíthatunk vele.
- Ezek a sütik is a *request* - *response*-okon keresztül mennek.

Süti küldése grizzly szerveren

```
public class CookieSenderPage extends HttpHandler {

    @Override
    public void service(Request request, Response response)
            throws Exception {
        final String respString =
            HtmlUtilities.createSimplePage(
                "Sender", "Cookie sent");
        response.setContentType("text/html");
        response.setContentLength(respString.length());
        response.getWriter().write(respString);

        response.addCookie(new Cookie("nev", "ertek"));
    }
}
```

Értelmezes

- Az *addCookie* metódussal tudunk hozzáadni egy új sütit a *response*-hoz.
- Ezt majd azonnal küldi az oldal ha megnyitottuk.
- A következő kóddal raktam be az két oldalt:

```
server.getServerConfiguration().addHttpHandler(
    new CookieSenderPage(), "/send");
server.getServerConfiguration().addHttpHandler(
    new CookieRecieverPage(), "/");
```

Süti ellenőrzese

```
public class CookieRecieverPage extends HttpHandler {
    @Override
    public void service(Request request, Response response)
            throws Exception {
        String respString;
        Cookie[] cookies = request.getCookies();
        boolean letezik = false;
        for(Cookie c: cookies) {
            if (c.getName().equals("nev")) {
                letezik = true;
            }
        }
        ...
```

Süti ellenőrzése 2

```
            ...
            if (letezik) {
                respString = HtmlUtilities.createSimplePage(
                    "Reciever", "Cookie :)");
            } else {
                respString = HtmlUtilities.createSimplePage(
                    "Reciever", "No cookie :(");
            }
            response.setContentType("text/html");
            response.setContentLength(respString.length());
            response.getWriter().write(respString);
        }
    }
```

Értelmezés

- Az oldalhoz tartozó sütiket minden *request*ben megkapja a webszerver, így ezeken bármikor vegigfuthatunk.
- A *request getCookies* metódusával kérdezhetjük le a sütik tömbjét.
- Majd ezen egy ciklussal végigmehetünk.
- A *Cookie getName* metódusával lekérhetjük a nevet, *getValue*-val az értékét, és még sok más dolgot is le lehet kérni.

Sütik élettartama

- Ha létrehozunk egy sütit:

```
Cookie c = new Cookie("nev", "ertek");
```

- Akkor sok paraméterét állíthatjuk még a nevén és értékén kívül, pl az élettartamát (hány másodpercig él mielőtt törlődik):

```
c.setMaxAge(15);
```