

Руковање клијентским подацима из веб форми помоћу Сервлета

У овој вежби су приказани основни принципи руковања клијентским подацима из веб форми помоћу сервлета.

Подаци се из Веб форме сервлету шаљу методама HTTP GET и HTTP POST, па се у позваном сервлету у зависности од одабране методе позивају методе:

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException
```

или

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException
```

У методама `doGet()` и `doPost()` треба преузети све параметре који се шаљу из Веб форми у порукама HTTP GET и HTTP POST.

Параметри које прима сервлет су спаковани у променљивој типа `HttpServletRequest`, а за више детља треба погледати документацију:

<https://tomcat.apache.org/tomcat-9.0-doc/servletapi/javax/servlet/http/HttpServletRequest.html>

Вредности параметара се могу добити позивом методе

```
java.lang.String getParameter(java.lang.String name)
```

која је наслеђена из Интерфејса

<https://tomcat.apache.org/tomcat-9.0-doc/servletapi/javax/servlet/ServletRequest.html>

ПРИМЕР (handlingforms)

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

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

1. Пренос параметара из Веб форме методама HTTP GET и HTTP POST

Креирати две Веб форме које имају иста поља, али параметре сервлету прослеђују помоћу метода HTTP GET и HTTP POST, што се специфицира помоћу атрибута METHOD којем се додељује одговарајућа вредност "GET" или "POST".

Код Веб форме (**GetForm.html**) која податке шаље методом HTTP GET је приказан у следећем листингу.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>A Web form using HTTP GET Method</title>
</head>
<body>
<center>
<h1>A Web form using HTTP GET Method</h1>
<form ACTION="GetPostFormsServlet" METHOD="GET">
  First name: <input TYPE="TEXT" NAME="firstName" VALUE="Miki"><br>
  Last name: <input TYPE="TEXT" NAME="lastName" VALUE="Maus"><p>
  <input TYPE="SUBMIT" VALUE="Submit with HTTP GET">
</form>
</center>
</body>
</html>
```

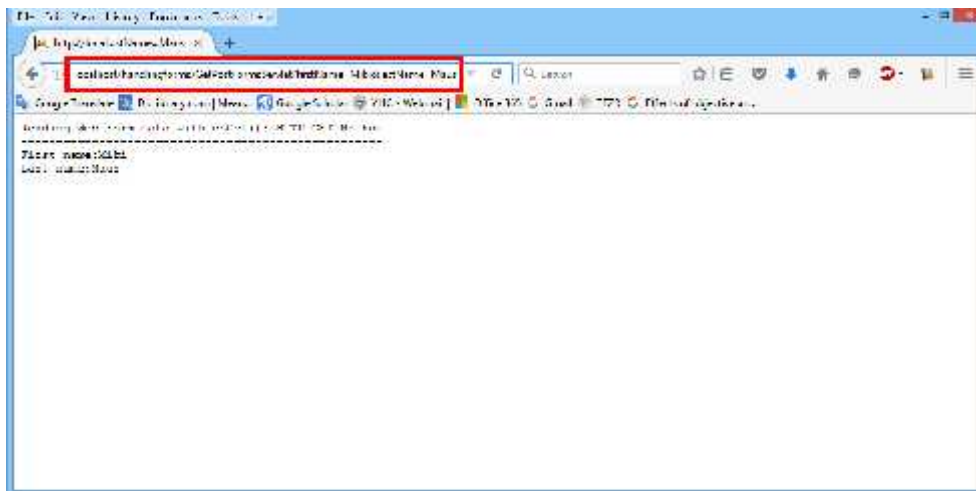
Веб форма **GetForm.html** је.



Из заглавља Веб форме

```
<form ACTION="GetPostFormsServlet" METHOD="GET">
```

се види да се извршавање акције притиском дугмета `input TYPE="SUBMIT"` позива метода `doGet` сервлета `GetPostFormsServlet`, која приказује податке које добија помоћу параметра `HttpServletRequest request`.



Приликом употребе методе HTTP GET вредности које се уносе у поља Веб форме се виде у адресној линији претраживача што се може приметити на претходној слици. Адресна линија је:

`http://localhost/handlingforms/GetPostFormsServlet?firstName=Miki&lastName=Maus`

Формат адресне линије је следећи:

- ❑ Адреса Веб апликације која рукује HTTP GET поруком, што је у овом случају сервлет `http://localhost/handlingforms/GetPostFormsServlet`.
- ❑ Знак упита (?) којим почиње стринг са параметрима који се преузимају из Веб форме.
- ❑ За сваку вредност из поља Веб форме се формира пар у формату `parameter=value`, па се тако у адресном стрингу могу уочити следеће вредности које су прочитане из поља у Веб форми `firstName=Miki` и `lastName=Maus`. Парови са именима поља и учитаним вредностима су раздвојени знаком `&`.

Код Веб форме (`PostForm.html`) која податке шаље методом HTTP POST је приказан у следећем листингу.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>A Web form using HTTP POST Method</title>
</head>
<body>
<center>
<h1>A Web form using HTTP POST Method</h1>
<form ACTION="GetPostFormsServlet" METHOD="POST">
  First name: <input TYPE="TEXT" NAME="firstName" VALUE="Paja"><br>
  Last name: <input TYPE="TEXT" NAME="lastName" VALUE="Patak"><p>
  <input TYPE="SUBMIT" VALUE="Submit with HTTP POST">
</form>
</center>
</body>
</html>
```

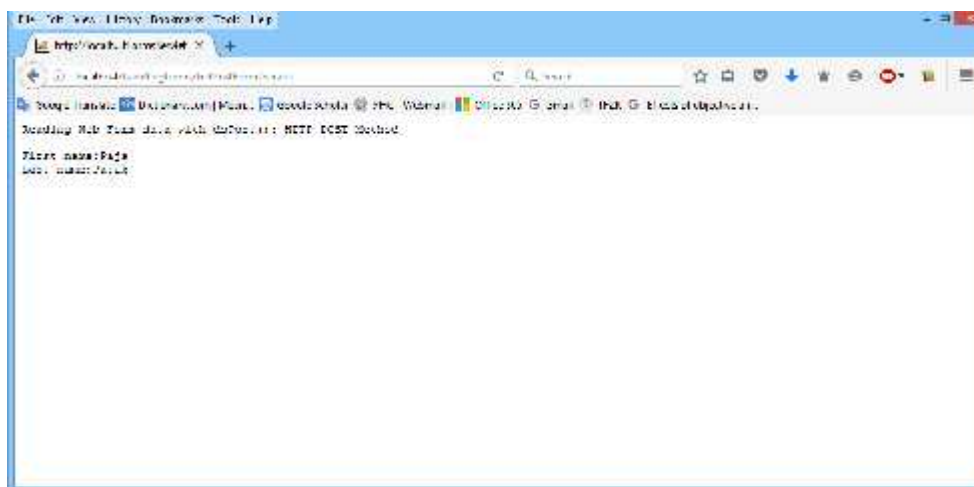
Веб форма **PostForm.html** је.



Из заглавља Веб форме

```
<form ACTION="GetPostFormsServlet" METHOD="POST">
```

се види да се извршавање акције притиском дугмета **input TYPE="SUBMIT"** позива метода **doPost** сервлета **GetPostFormsServlet**, која приказује податке које добија помоћу параметра **HttpServletRequest request**.



Приликом употребе методе HTTP POST вредности које се уносе у поља Веб форме се не виде у адресној линији претраживача, што се може приметити на претходној слици.

Сервлет **GetPostFormsServlet** који рукује подацима са форми треба пријавити у **web.xml** датотеци Веб апликације.

```
<servlet>
    <servlet-name>GetPostFormsServlet</servlet-name>
    <servlet-class>GetPostFormsServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>GetPostFormsServlet</servlet-name>
    <url-pattern>/GetPostFormsServlet</url-pattern>
</servlet-mapping>
```

2. Руковање подацима из Веб форме

Креирати Веб форму која омогућује унос података за плаћање чланарине у библиотеци. Веб форма треба да омогући унос следећих података:

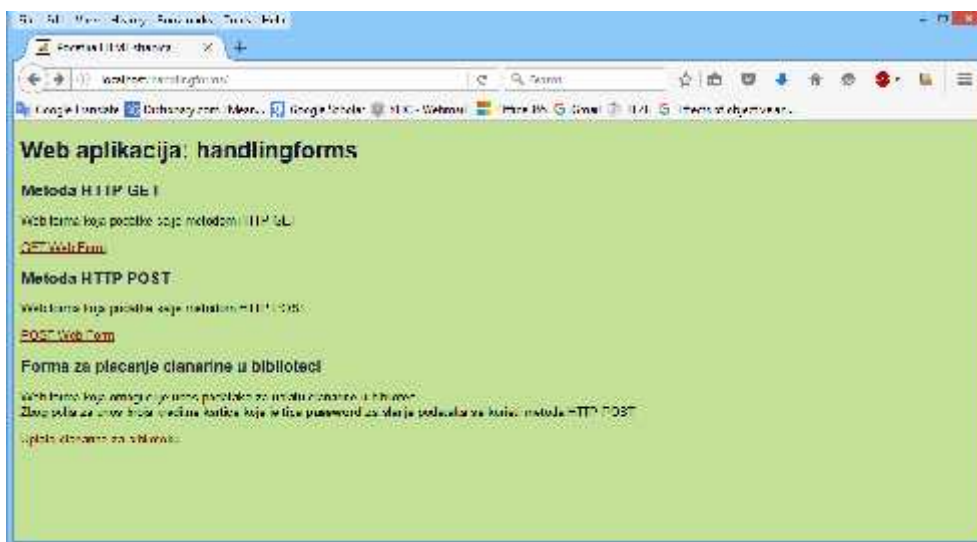
- име особе,
- презиме особе,
- број чланске карте,
- избор типа кредитне картице за плаћање,
- број кредитне картице,
- приказ износа чланарине (нпр. 300 динара)
- поље у којем се потврђује да ли се чланарина наставља од претходне године, у којем случају се износ чланарине умањује 10%.

Написати сервлет (**UplataClanarineServlet**) који учитава унете податке и уписује их у датотеку са именом које одговара чланском броју и годином за коју се уплаћује чланарина (нпр. име фајла је **3257-2018.cbt**). За прихватање података из Веб форме пројектовати одговарајућу **JavaBean** класу, коју потом треба икористити за серијализацију објекта у датотеку.

Решење

Најпре треба обезбедити позив форме за плаћање чланарине из главне странице, на пример додавањем следећег кода:

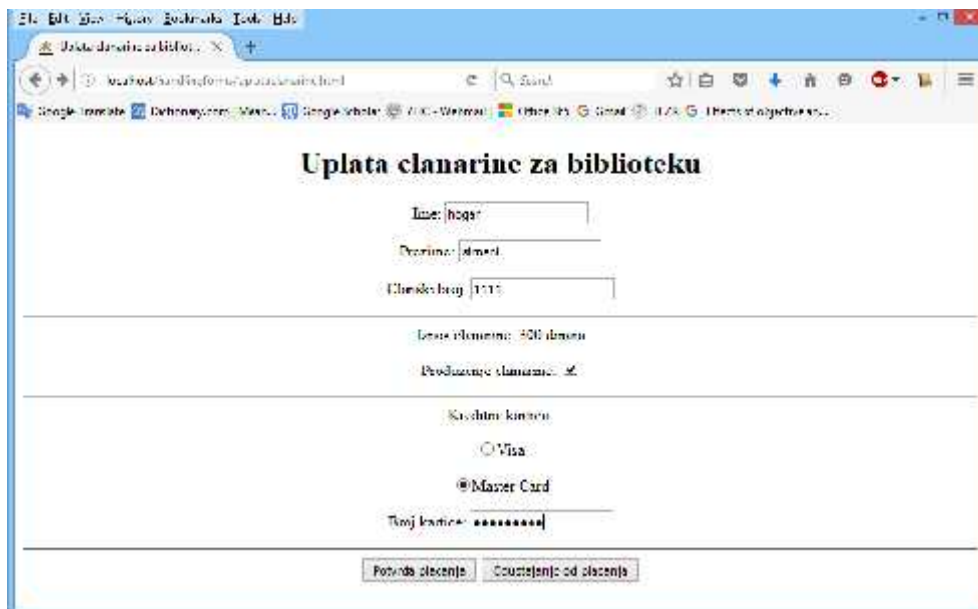
```
<h2>Forma za placanje clanarine u biblioteci</h2>
<p>Web forma koja omogucuje unos podataka za uplatu clanarine u biblioteci.<br>
Zbog polja za unos broja kreditne kartice koje je tipa <b>password</b> za slanje
podataka se koristi metoda HTTP POST</p>
<p><a href="uplataclanarine.html">Uplata clanarine za biblioteku</a></p>
```



Након тога треба креирати веб форму која садржи сва потребна поља за унос података за плаћање чланарине у библиотеци. При томе треба обезбедити да се потврда уплате реализује дугметом типа **SUBMIT** са одговарајућим натписом (нпр. Потврда плаћања), али и да се обезбеди опција за одустајање од плаћања која враћа корисника на почетну страну (**index.html**).

HTML код Веб форме је приказан у следећем листингу.

```
<form ACTION="UplataClanarineServlet" METHOD="POST">
  Име: <input TYPE="TEXT" NAME="ime"><p>
  Презиме: <input TYPE="TEXT" NAME="prezime"><p>
  Clanski broj: <input TYPE="TEXT" NAME="clanskiBroj"><p>
  <hr>
  Iznos clanarine: 300 dinara<p>
  Produzenje clanarine: <INPUT TYPE="CHECKBOX"
    NAME="produzenjeClanarine"><p>
  <hr>
  Kreditna kartica:<p>
  <INPUT TYPE="RADIO" NAME="kreditnaKartica" VALUE="visa">Visa<p>
  <INPUT TYPE="RADIO" NAME="kreditnaKartica"
    VALUE="mastercard">Master Card<p>
  Broj kartice: <INPUT TYPE="PASSWORD" NAME="brojKreditneKartice"><p>
  <hr>
  <input TYPE="SUBMIT" NAME="potvrda" VALUE="Potvrda placanja">
  <input TYPE="SUBMIT" NAME="odustajanje" VALUE="Odustajanje od placanja">
</form>
```



Сервлет **UplataClanarineServlet** треба пријавити у датотеку **web.xml**.

```
<servlet>
  <servlet-name>UplataClanarineServlet</servlet-name>
  <servlet-class>UplataClanarineServlet</servlet-class>
</servlet>
...
<servlet-mapping>
  <servlet-name>UplataClanarineServlet</servlet-name>
  <url-pattern>/UplataClanarineServlet</url-pattern>
</servlet-mapping>
```

С обзиром да се приликом преноса параметара из Веб форме појављује поље поље чији садржај треба да буде невидљив у Веб форми и приликом преноса на сервер, користи се HTTP POST метода за пренос података на сервер. То значи да сервлет треба код за обраду уплате чланарине да

имплементира у методи `doPost()`, при чему треба да утврди да ли је притиснуто дугме за потврду уплате чланарине, или дугме за одустајање и повратак (редирекција) на почетну страну.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    if (request.getParameter("potvrda") != null) {
        // uplata clanarine
        PrintWriter out = response.getWriter();
        out.println("Potvrda uplate clanarine");
        ...
        // serijalizacija podataka u fajl pomocu JavaBean klase
        ...
    } else if (request.getParameter("odustajanje") != null) {
        // odustajanje od uplate clanarine
        // redirekcija na glavnu stranu
        response.sendRedirect("index.html");
    }
}
}
```

За серијализацију података о чланарини је креирана класа `ClanarinaBiblioteke` која имплементира интерфејс `Externalizable`. Серијализација је имплементирана у оквиру `doPost()` методе сервлета `UplataClanarineServlet`, што је приказано у следећем листингу (фајл је снимљен у контексту веб апликације).

```
String tipKartice = request.getParameter("kreditnaKartica");
if (tipKartice.equals("visa")){
    out.println("Tip kartice: Visa");
    tipKartice = "Visa";
}
if (tipKartice.equals("mastercard")){
    out.println("Tip kartice: Master Card");
    tipKartice = "Master Card";
}

// serijalizacija u fajl
ClanarinaBiblioteke cb = new ClanarinaBiblioteke();
cb.setIme(request.getParameter("ime"));
cb.setPrezime(request.getParameter("prezime"));
cb.setClanskiBroj(request.getParameter("clanskiBroj"));
cb.setTipKreditneKartice(tipKartice);
cb.setBrojKreditneKartice(request.getParameter("brojKreditneKartice"));

// izbor foldera veb aplikacije
String webAppFolder = getServletContext().getRealPath("/");
String datoteka = webAppFolder + cb.getClanskiBroj() + "-2018.cbt";
FileOutputStream fos;
try {
    fos = new FileOutputStream(datoteka);
    ObjectOutputStream oos = new ObjectOutputStream(fos);
    oos.writeObject(cb);
    oos.close();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

Задачи за самостални рад

Задатак 1

Решени пример са са уписом појединачних уплата чланарине помоћу Веб форме проширити тако да се омогући:

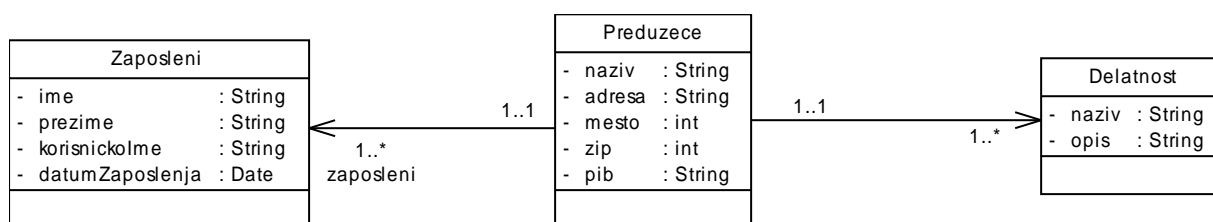
- (1) Чување свих уплата у листи.
- (2) Написати сервлет који чита снимљене уплате у фајловима и приказује их у форматираној HTML страници.

Задатак 2

Написати и тестирати Веб апликацију која омогућује:

- (1) Ажурирање података о предузећу помоћу Веб форме. Податке снимити у датотеци **preduzece.pred**.
- (2) Евидентирање запослених у предузећу тако што се подаци о новим запосленима уносе у Веб форми. Такође написати сервлет који приказује списак (листу) свих запослених у форматираној HTML страници. Податке о запосленима чувати у листи која се снима у датотеку **zaposleni.pred**.
- (3) Евидентирање делатности предузећа тако што се подаци о новим делатностима уносе у Веб форми, а списак делатности се чува у листи која се серијализује у датотеку **delatnosti.pred**. Такође обезбедити приказ свих делатности предузећа у форматираној HTML страници.
- (4) Креирати почетну страницу Веб апликације из које се покрећу остале Веб странице потребне за реализацију апликације. Обезбедити унифицирано форматирање свих страница применом CSS стилова.

Ентитети потребни за реализацију Веб апликације су приказани на следећој слици.



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 19: Creating and Processing HTML Forms.
- [3] Marty Hall and Larry Brown. *Core Servlets and JavaServer Pages, Free Online Version of Second Edition.* <http://pdf.coreservlets.com/>. Chapter 4: Handling the Client Request: Form Data.
- [4] <https://tomcat.apache.org/tomcat-9.0-doc/servletapi/overview-summary.html>
- [5] <https://tomcat.apache.org/tomcat-9.0-doc/servletapi/index.html>.