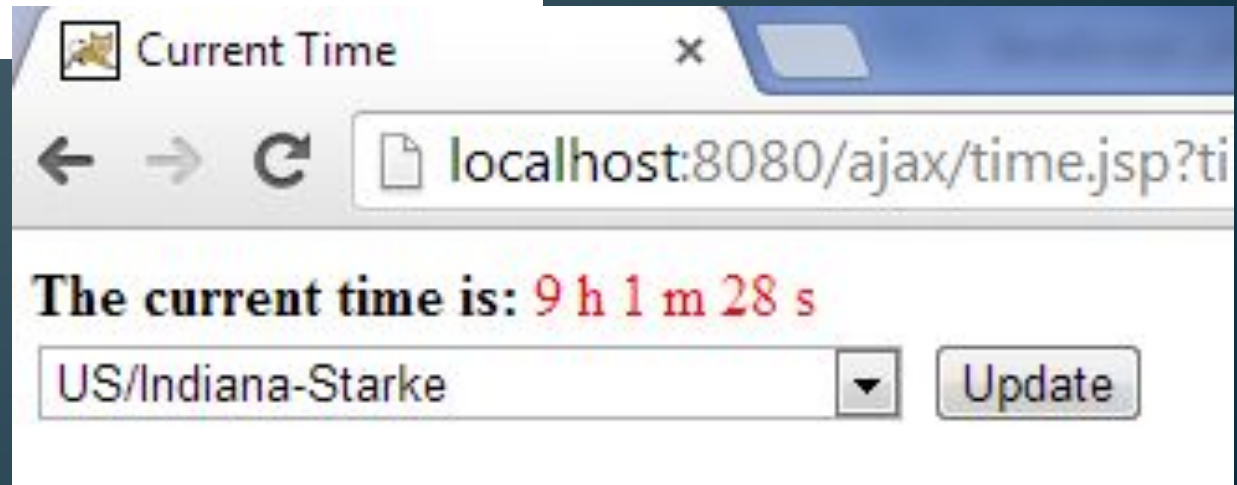
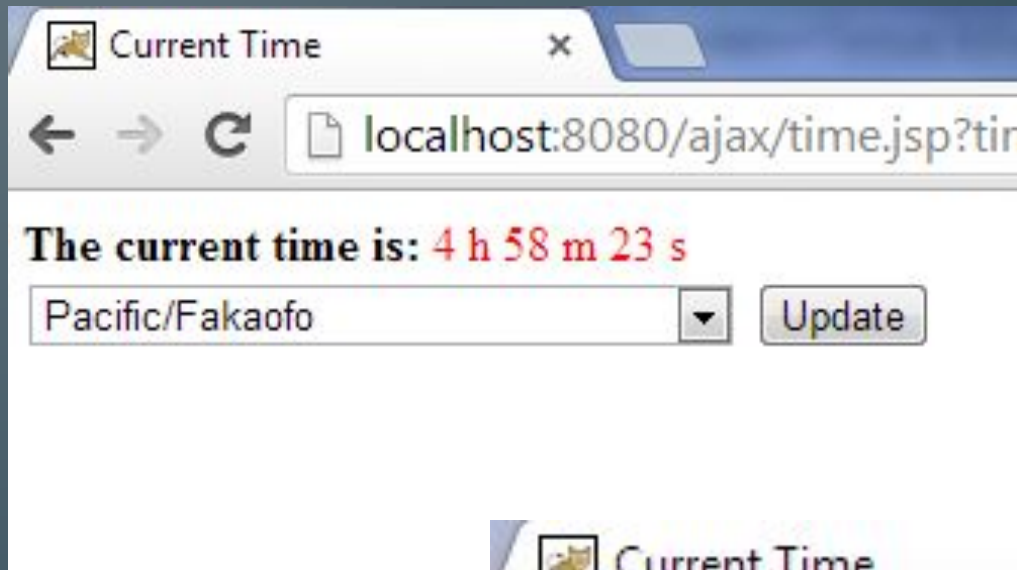


Web

Тема 9.

Традиционный подход к веб-программированию на основе перезагрузки страницы

Рассмотрим пример



```
<%@ page import="java.util.*"%>
```

```
<%
    String tz = request.getParameter("timezone");
    if (tz == null) {
        tz = TimeZone.getAvailableIDs()[0];
    }
    Calendar c = Calendar.getInstance(TimeZone.getTimeZone(tz));
    int hs = c.get(Calendar.HOUR);
    int ms = c.get(Calendar.MINUTE);
    int ss = c.get(Calendar.SECOND);
%>
```

```
<html>
  <head><title>Current Time</title></head>

  <body>
    <strong>The current time is:</strong>
    <span style="color:red">
      <% out.println(hs + " h " + ms + " m " + ss + " s"); %>
    </span>
    <br>
    <form action="time.jsp" method="get">
      <select name="timezone">
        <%
          for (String x : TimeZone.getAvailableIDs()) {
            if (x.equals(tz)) {
              out.println("<option selected value=\'" + x + "\'>" + x + "</option>");
            } else {
              out.println("<option value=\'" + x + "\'>" + x + "</option>");
            }
          }
        %>
      </select>
      <input type="submit" value="Update">
    </form>
  </body>
</html>
```

Место для отображения времени

```
<strong>The current time is:</strong>  
<span style="color:red">  
    <% out.println(hs + " h " + ms + " m " + ss + " s"); %>  
</span>  
<br>
```

Форма для отправки запроса

```
<form action="time.jsp" method="get">
  <select name="timezone">
    <%
      for (String x : TimeZone.getAvailableIDs()) {
        if (x.equals(tz)) {
          out.println("<option selected value=\'" + x + "\'>" + x + "</option>");
        } else {
          out.println("<option value=\'" + x + "\'>" + x + "</option>");
        }
      }
    %>
  </select>
  <input type="submit" value="Update">
</form>
```

Сгенерированная страница

The current time is: 9 h 1 m 28 s

US/Indiana-Starke

Update

Назад

Вперед

Перезагрузить

Сохранить как...

Печать...

Перевести на русский

Просмотр кода страницы

Просмотр сведений о странице

Просмотр кода элемента

```
14         <form action="time.jsp" method="get">
15             <select name="timezone">
16                 <option value='Etc/GMT+'>Etc/GMT+
17 <option value='Etc/GMT+11'>Etc/GMT+11</option>
18 <option value='MIT'>MIT</option>
19 <option value='Pacific/Apia'>Pacific/Apia</option>
20 <option value='Pacific/Midway'>Pacific/Midway</option>
21 <option value='Pacific/Niue'>Pacific/Niue</option>
22 <option value='Pacific/Pago_Pago'>Pacific/Pago_Pago</option>
23 <option value='Pacific/Samoa'>Pacific/Samoa</option>
24 <option value='US/Samoa'>US/Samoa</option>
25 <option value='America/Adak'>America/Adak</option>
26 <option value='America/Atka'>America/Atka</option>
27 <option value='Etc/GMT+10'>Etc/GMT+10</option>
28 <option value='HST'>HST</option>
```

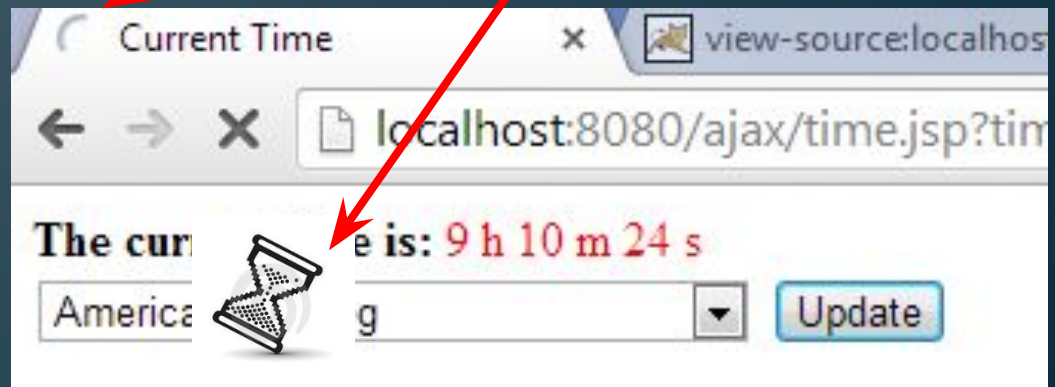
Предварительные расчеты

```
<%@ page import="java.util.*"%>
```

```
<%  
    String tz = request.getParameter("timezone");  
    if (tz == null) {  
        tz = TimeZone.getAvailableIDs()[0];  
    }  
    Calendar c = Calendar.getInstance(TimeZone.getTimeZone(tz));  
    int hs = c.get(Calendar.HOUR);  
    int ms = c.get(Calendar.MINUTE);  
    int ss = c.get(Calendar.SECOND);  
%>
```

Эксперимент – проверим, что запрос синхронный и что происходит перезагрузка страницы

```
<8 Thread.sleep(5000);  
String tz = request.getParameter("timezone");  
if (tz == null) {
```



Свойства традиционного подхода

- + достаточно простой код
- + не нужно включать JavaScript
- происходит перезагрузка страницы
- запрос синхронный
- По сети передается много дублирующей информации

Реализация серверного блога

The image shows a screenshot of a web application interface for a server-side blog. At the top, there is a text input field containing "А теперь - спать!" and a button labeled "Добавить запись". A red arrow points from a purple callout box "Обычный запрос на сервер" to the "Добавить запись" button. Below the input field is a list of blog entries, each consisting of a timestamp and a text snippet. A second purple callout box, "Это генерирует сервер", has two red arrows pointing to the timestamp and text of the second entry: "7:10:00 19.11.2013" and "Умываюсь...".

А теперь - спать!

7:00:00 19.11.2013
Я проснулся!

7:10:00 19.11.2013
Умываюсь...

7:20:00 19.11.2013
Пошел варить кофе

7:50:00 19.11.2013
Куда опять ключи подевались?

8:05:00 19.11.2013
Опять опаздываю

21:25:02 18.10.2013
А теперь - спать!

Обычный запрос на сервер

Это генерирует сервер

Blog.jsp

целиком

Создание списка записей

Обработка запроса

Генерация html-страницы

```
<%@ page import="java.util.*"%>
<%! class Record {
    Date data;
    String message;
    Record(Date data, String message) {
        this.data = data;
        this.message = message;
    }
};
ArrayList<Record> records = new ArrayList<Record>();

if (records.size() == 0) {
    records.add(new Record(new Date(2013, 12, 02, 8, 10), "Проснулся!"));
    records.add(new Record(new Date(2013, 12, 02, 8, 30), "Умылся."));
    records.add(new Record(new Date(2013, 12, 02, 8, 45), "Варю кофе..."));
}

String message = request.getParameter("message");
if (message != null && message.trim().length() != 0) {
    records.add(new Record(new Date(), message));
}

<html>
<head><title>My Blog</title></head>
<body>
    <form action="blog.jsp" method="post">
        <input type="text" name="message"/>
        <input type="submit" value="Отправить">
    </form>
    <% int i=0;
    for (Record r : records) {
        if (i%2==0) {
            out.println("<p style='background-color:gray'>");
        } else {
            out.println("<p style='background-color:lightgray'>");
        }
        out.println("<strong>" + r.data + "</strong>" + "<br>");
        out.println(r.message + "<br>");
        out.println("</p>");
        i++;
    }
    <%>
</body></html>
```

Создание списка записей

```
<%@ page import="java.util.*"%>
<%! class Record {
    Date data;
    String message;
    Record(Date data, String message) {
        this.data = data;
        this.message = message;
    }
};
ArrayList<Record> records = new ArrayList<Record>();
%>
```

Обработка запроса и добавление записей

```
<&  
if (records.size() == 0) {  
    records.add(new Record(new Date(2013, 12, 02, 8, 10), "Проснулся!"));  
    records.add(new Record(new Date(2013, 12, 02, 8, 30), "Умылся."));  
    records.add(new Record(new Date(2013, 12, 02, 8, 45), "Варю кофе..."));  
}  
  
String message = request.getParameter("message");  
if (message != null && message.trim().length() != 0) {  
    records.add(new Record(new Date(), message));  
}  
&>
```

Генерация html-страницы

```
<html>
  <head><title>My Blog</title></head>
  <body>
    <form action="blog.jsp" method="post">
      <input type="text" name="message"/>
      <input type="submit" value="Отправить">
    </form>
    <%
      int i=0;
      for (Record r : records) {
        if (i%2==0) {
          out.println("<p style='background-color:gray'>");
        } else {
          out.println("<p style='background-color:lightgray'>");
        }
        out.println("<strong>" + r.data + "</strong>" + "<br>");
        out.println(r.message + "<br>");
        out.println("</p>");
        i++;
      }
    %>
  </body></html>
```

Сгенерированная страница

```
<html>
  <head><title>My Blog</title></head>
  <body>
    <form action="blog.jsp" method="post">
      <input type="text" name="message"/>
      <input type="submit" value="Отправить">
    </form>
    <p style='background-color:gray'>
<strong>Fri Jan 02 08:10:00 NOVT 3914</strong><br>
Проснулся!<br>
</p>
<p style='background-color:lightgray'>
<strong>Fri Jan 02 08:30:00 NOVT 3914</strong><br>
Умылся.<br>
</p>
<p style='background-color:gray'>
<strong>Fri Jan 02 08:45:00 NOVT 3914</strong><br>
Варю кофе...<br>
</p>
<p style='background-color:lightgray'>
<strong>Tue Dec 03 16:27:05 NOVT 2013</strong><br>
dsfdfsdfs<br>
</p>

  </body></html>
```

Практическое задание

- Реализуйте одно из заданий на свой выбор:
 - Реализуйте страницу с отображением времени в зависимости от часового пояса.
 - Реализуйте блог в виде одной серверной страницы.
- После выполнения задания:
 - Просмотрите сгенерированный сервером код.
 - Проведите эксперимент с 5-ти секундной задержкой.