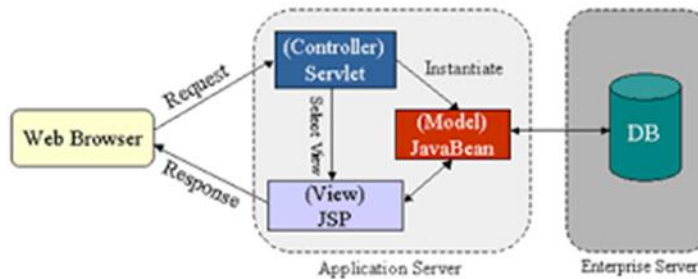


Unidade 5 – Cookies

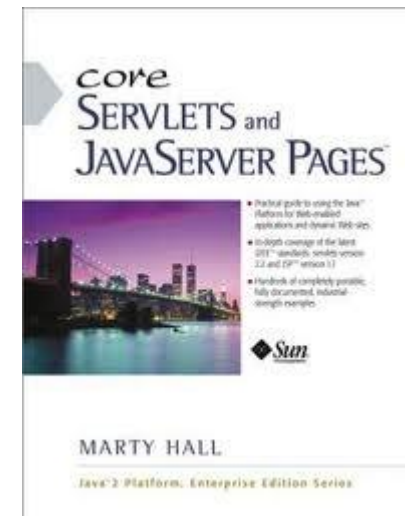
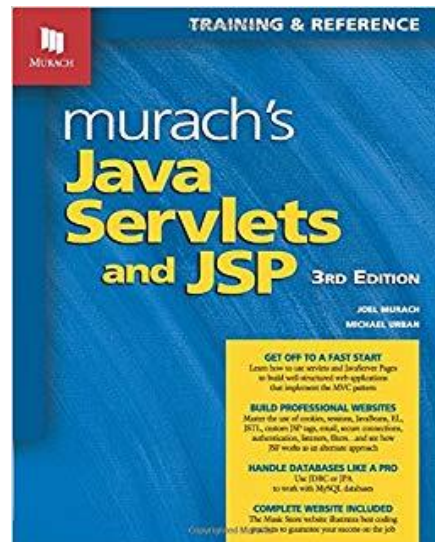


Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP



Referências

- Head First Servlets & JSP – Bryan Basham, Kathy Sierra & Bert Bates
- Core Servlets and Java Server Pages – Marty Hall
- Java Servlets and JSP – Joel Murach – 3rd Edition



Introdução

- **Cookies** são pequenas quantidades de informações que o Web Server envia para um browser;
- Mais tarde, em outra transação, o browser pode devolver esta informação (sem modificá-la) para o **Web Server**;
- Esta facilidade pode prover diversas conveniências para aplicações Web;
- Não são interpretados ou executados e assim não oferecem riscos de ataques de viroses, porém podem representar ameaças à privacidade do usuário.



Benefícios dos Cookies

- Permite a identificação de um usuário durante uma sessão e-commerce;
- Guardar nomes de usernames e passwords;
- Customização de sites para lembrar preferências de usuários;
- Direcionar usuários a tópicos de interesse e relevância.



Enviando e recebendo cookies

- ✿ Para enviar cookies ao cliente, um Servlet deve usar o construtor `Cookie` para criar um ou mais cookies;
- ✿ Para setar atributos dos cookies, a função `cookie.setXxx` deve ser usada;
- ✿ Para ler estes atributos, usa-se a função `cookie.getXxx`;
- ✿ A inserção de cookies nos headers response HTTP, usa-se `response.addCookie`;
- ✿ Para processar a leitura de cookies, um servlet deve chamar `request.getCookies`, o qual retorna um array de objetos `Cookie` ou `null` se não houver cookies no request;



Enviando cookies para o cliente

1. Criando um objeto cookie. Por meio do construtor **Cookie**.
2. Definindo a idade máxima. Deve-se usar **setMaxAge** para especificar quanto tempo (em segundos) o cookie será válido.
3. Inserindo o cookie nos headers HTTP response. Feito por **response.addcookie**. Se esquecer disto, o cookie não será enviado ao browser.



Criando cookies

- Ⓢ O construtor **Cookie** recebe dois parâmetros do tipo **String**: o nome do cookie e o valor.
- Ⓢ Nem o nome nem o valor devem ter espaços em branco ou os seguintes caracteres: [] () = , " / ? @ : ;
- Ⓢ Exemplo: **Cookie x = new Cookie("userid", "a1234");**



Definindo a idade máxima

- Se você criar um cookie e enviá-lo ao browser, por default, é um cookie session-level, ou seja: o cookie é armazenado na memória do browser e deletado quando o usuário sair do browser.
- Para definir persistência de cookie, usa-se a função **setMaxAge** com o parâmetro em segundos. Exemplo:

```
x.setMaxAge(60*60*24*7);           //1 semana
```

Obs.: Definindo 0 como parâmetro, instrui o browser a deletar o cookie.



Inserindo cookie no response

⊕ Feito por meio do método `addCookie` do objeto **`HttpServletResponse`**.

⊕ Exemplo:

```
Cookie userCookie = new Cookie("user", "uid1234");  
userCookie.setMaxAge(60*60*24*365); // cookie válido por 1 ano  
response.addCookie(userCookie);
```



Lendo cookies do cliente

1. Call **request.getCookies**. Isto retorna um array de objetos Cookies.
2. Processa-se um loop no array indexando **getName** em cada elemento até encontrar o cookie de interesse.

```
String cookieName = "userID";
Cookie[] cookies = request.getCookies();
if (cookies != null ) {
    for (int i=0; i <cookies.length; i++) {
        Cookie cookies[i];
        if (cookieName.equals(cookie.getName() ) ) {
            doSomething(cookie.getValue() );
        }
    }
}
```



Atividade – 14

- ✚ Escrever um servlet (**Atividade_14**) que envia uma página de boas vindas se o usuário não tiver cookies associados à sessão e página de “Feliz Retorno” se existirem cookies.

Obs. Utilizar o ambiente Eclipse / Tomcat



```
package qualit;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class Atividade_14
 */
@WebServlet("/Atividade_14")
public class Atividade_14 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Atividade_14() {
        super();
        // TODO Auto-generated constructor stub
    }
}
```



```
/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
    boolean newbie = true;
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
        for (int i=0; i<cookies.length ; i++){
            Cookie c = cookies[i];
            if (( c.getName().equals("repeatVisitor")) && (c.getValue().equals("yes"))) ) {
                newbie=false;
                break;
            }
        }
    }
}
```



```
String title;
if (newbie) {
    Cookie returnVisitorCookie = new Cookie("repeatVisitor", "yes");
    response.addCookie(returnVisitorCookie);
    title = " Atividade 14 - Bem vindo ...";
}
else title = "Atividade 14 - Olá novamente...";

response.setContentType("text/html");
PrintWriter out = response.getWriter();

String docType =
    "<!DOCTYPE HTML PUBLIC \"/>";
out.println(docType +
    "<HTML>\n" +
    "<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
    "<BODY BGCOLOR=\"<H1 ALIGN=\"<H1>\n" +
    "</BODY></HTML>");
}
```

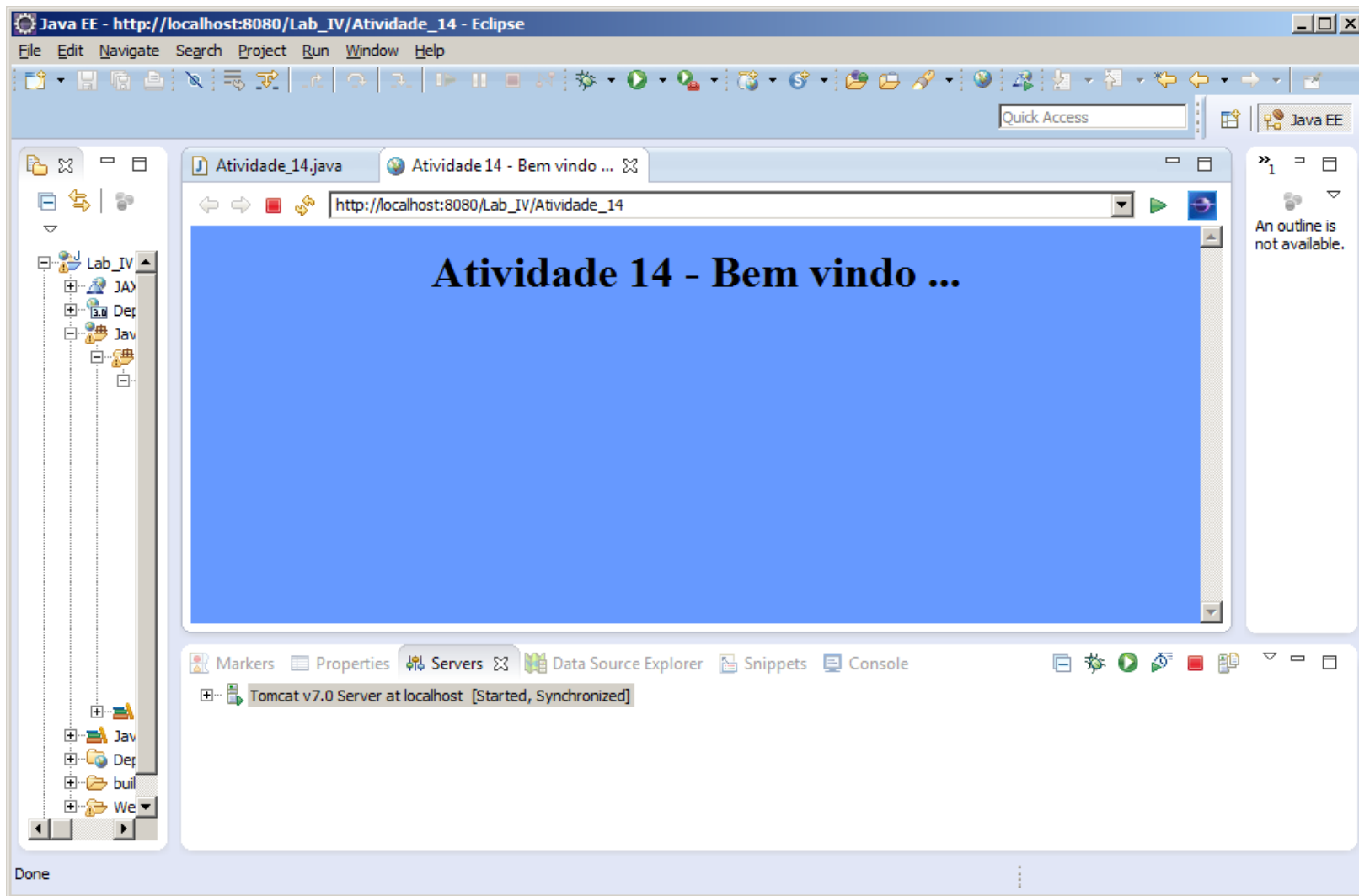


```
/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
// TODO Auto-generated method stub
}

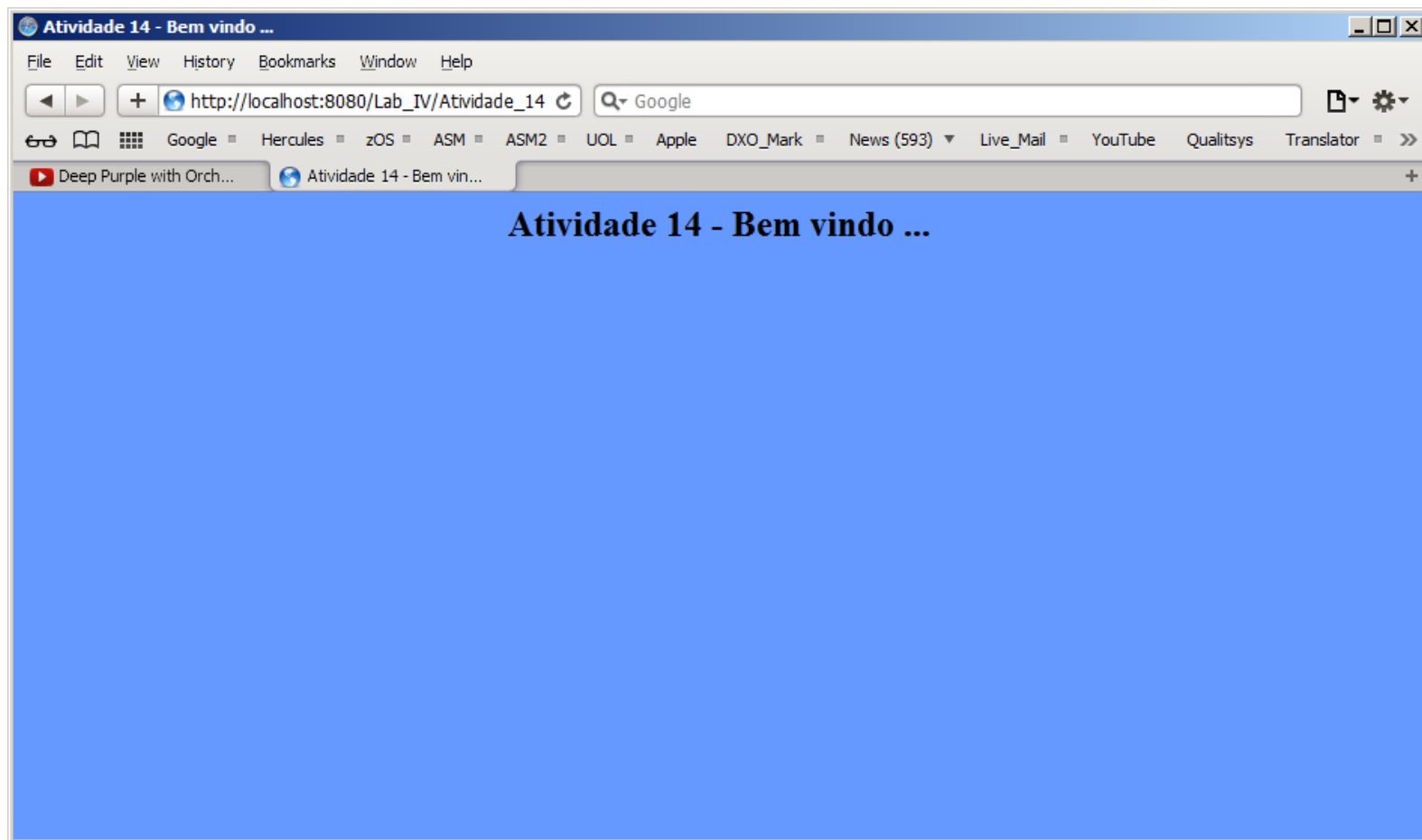
}
```



http://localhost:8080/Lab_IV/Atividade_14



http://localhost:8080/Lab_IV/Atividade_14



http://localhost:8080/Lab_IV/Atividade_14

