

Universidade Presbiteriana Mackenzie  
Faculdade de Computação e Informática

# Cookies e sessões

NOTAS DE AULA - Teoria 19

Linguagem de Programação II  
2º semestre de 2015  
Versão 1.0

# Referências

As referências para esta aula são os capítulos 8 e 9 de:

HALL, M.; et al. **Core Servlets and Javasever Pages: vol. 1: Core Technologies**. New jersey: Prentice-Hall, 2003.

## Observações:

- As notas de aula são material de apoio para a aula e não têm o objetivo de apresentar o assunto de maneira exaustiva. Não deixe de ler o material de referência da disciplina.
- Para reduzir o número de linhas de código, os exemplos apresentados omitem propositalmente a importação das classes. Para esta aula, a maior parte das classes pertencem aos pacotes **javax.servlet** e **javax.servlet.http**. As principais IDEs Java possuem recursos para auxiliar a inclusão das importações das classes.
- Os trechos de código fornecidos são simplistas e curtos para podermos focar no assunto que está sendo estudado, por isso não devem ser utilizados diretamente em produção.

# Informações de estado

- O protocolo HTTP **não** mantém **informações de estado**.
- A aplicação pode utilizar os recursos de **cookies** e **sessões** para manter informações de estado.

# Cookies

- Pequenas porções de informação textual.
- A informação é enviada pelo servidor para o navegador.
- O navegador salva os dados no cliente e retorna esta informação sem modificações para o servidor ao visitar o mesmo site ou domínio.

# Benefícios obtidos com cookies

- Identificação do usuário durante uma sessão de e-commerce.
- Recuperação de *usernames* e *passwords*.
- Customização de sites.
- Apresentação de anúncios de acordo com tópicos de interesse do usuário.

# Limitações dos cookies

- Só é possível armazenar **informação textual**.
- Os navegadores limitam a quantidade de cookies (geralmente 20 cookies por site e 300 cookies no total).
- A informação de cada cookie é limitada a 4 kilobytes.

# Possíveis problemas com cookies

- Dois sites podem compartilhar dados de um usuário carregando um recurso (por exemplo, uma imagem) de um terceiro site que utiliza cookies.
- Embora não seja usual, um site poderia gravar dados “sensíveis” em cookies, tais como senhas ou números de cartão de crédito, que poderiam ser acessados no seu computador.

# Envio de cookies para o cliente

Para enviar um cookie para o cliente, o seu servlet deve:

- Criar uma instância de **Cookie** utilizando o construtor

**Cookie(String name, String value)**

- Atribuir um prazo de validade para o cookie utilizando o método **setMaxAge**. Se este método não for chamado, o valor padrão para **maxAge** é -1, o que torna o cookie válido apenas para a sessão corrente.

**void setMaxAge(long seconds)**

- Adicionar o cookie na resposta HTTP utilizando o método **addCookie** da instância de **HttpServletResponse**.

**void addCookie(Cookie cookie)**



# Exemplo de gravação de cookies

Formulário para entrada dos dados do cookie:

```
<html>
  <head>
    <title>Gravar cookie</title>
  </head>
  <body bgcolor=red>
    <h1>Gravação de cookie</h1>
    <form method="POST" action="GravarCookieServlet">
      <p>Nome: <input type="text" name="nome_cookie" /></p>
      <p>Valor: <input type="text" name="valor_cookie" /></p>
      <p><input type="submit" /></p>
    </form>
  </body>
</html>
```

# Exemplo de gravação de cookies (cont.)

Servlet que recebe os dados do formulário e grava o cookie:

```
public class GravarCookieServlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String nome = request.getParameter("nome_cookie");
        String valor = request.getParameter("valor_cookie");
        Cookie meuCookie = new Cookie(nome, valor);
        meuCookie.setMaxAge(5*60);
        response.addCookie(meuCookie);
        out.println("<html><head>");
        out.println("<title>Resultado da gravação do cookie</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Cookie " + nome + " gravado!</h1>");
        out.println("</body></html>");
    }
}
```

# Leitura de cookies enviados pelo cliente

Para ler um cookie enviado pelo cliente, o seu servlet deve:

- Chamar o método **getCookies** para receber um array com todos os cookies da instância de **HttpServletRequest**.

**Cookie[] getCookies()**

- Utilizar os métodos **getName** e **getValue** da instância de **Cookie**.

**String getName()**

**String getValue()**

# Exemplo de leitura de cookies

Servlet que lê o valor do cookie “cor\_de\_fundo” e usa seu valor como cor de fundo da página

```
public class HelloServlet extends HttpServlet {
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {

        PrintWriter out;
        out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Resultado do login</title></head>");
        Cookie[] cookies = request.getCookies();
        String corDeFundo = "";
        for (Cookie c: cookies) {
            if (c.getName().equals("cor_de_fundo")) {
                corDeFundo = c.getValue();
            }
        }
        out.println("<body bgcolor=" + corDeFundo + ">");
        out.println("<h1>Data e hora: " + new java.util.Date() + "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

# Atualizar o valor de um cookie

Para atualizar o valor de um cookie já existente, o seu servlet deve:

- Ler o cookie enviado pelo cliente ou criar uma nova instância de **Cookie** com o mesmo nome.
- Chamar o método **setValue** da instância de **Cookie** passando como argumento o novo valor.
- Chamar o método **addCookie** de **HttpServletResponse**, passando a referência do cookie como argumento.

# Remoção de um cookie

Para remover um cookie, o seu servlet deve:

- Ler o cookie enviado pelo cliente ou criar uma nova instância de **Cookie** com o mesmo nome.
- Chamar o método **setMaxAge** da instância de **Cookie** passando o argumento igual a 0.
- Chamar o método **addCookie** de **HttpServletResponse**, passando a referência do cookie como argumento.

# Navegadores com cookies desabilitados

- Caso o navegador web **não permita o uso de cookies**, além de não ser possível utilizar os cookies para manter as informações de estado também não será possível armazenar o identificador da sessão como um cookie.
- Soluções alternativas para manter informações de estado em navegadores web que **não** permitem o uso de cookies:
  - *URL rewriting* (reescrita da URL)
  - *Hidden form fields* (campos de formulário ocultos)
- Para maiores informações, consulte as referências desta aula.

# Sessão e variáveis de sessão

- Uma sessão corresponde à navegação que um cliente específico está realizando através dos recursos da sua aplicação web.
- O servidor de aplicações permite o armazenamento de **objetos diversos** associados à navegação que está sendo realizada pelo cliente.
- Estes objetos são armazenados no servidor de aplicações em **variáveis de sessão**.



# Criação de variável na sessão corrente

Para criar uma nova variável na sessão associada à requisição recebida pelo servlet:

- Chamar o método **getSession** de **HttpServletRequest**.

**HttpSession getSession()**

- Chamar o método **setAttribute** de **HttpSession**, passando como argumento o objeto a ser armazenado

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

# Exemplo de gravação na sessão

Classe cujas instâncias desejamos gravar na sessão:

```
public class ItemVenda {  
    private String descricao;  
    private int quantidade;  
    public ItemVenda(String d, int q) {  
        descricao = d;  
        quantidade = q;  
    }  
    public String getDescricao() { return descricao; }  
    public int getQuantidade() { return quantidade; }  
}
```

# Exemplo de gravação na sessão (cont.)

Formulário para a entrada de dados do item de venda:

```
<html>
  <head>
    <title>Gravar item de venda</title>
  </head>
  <body bgcolor=green>
    <h1>Gravação de item de venda</h1>
    <form method="POST" action="GravarItemVendaServlet">
      <p>Descrição do produto: <input type="text" name="descricao" /></p>
      <p>Quantidade: <input type="text" name="quantidade" /></p>
      <p><input type="submit" /></p>
    </form>
  </body>
</html>
```

## Exemplo de gravação na sessão (cont.)

Servlet que adiciona uma instância de **ItemVenda** na sessão com os valores do formulário:

```
public class GravarItemVendaServlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String descricao = request.getParameter("descricao");
        String quantidade = request.getParameter("quantidade");
        ItemVenda item = new ItemVenda(descricao, Integer.parseInt(quantidade));
        HttpSession session = request.getSession();
        List<ItemVenda> itens = (List<ItemVenda>) session.getAttribute("itensVendidos");
        if (itens == null) {
            itens = new ArrayList<ItemVenda>();
        }
        itens.add(item);
        session.setAttribute("itensVendidos", itens);
        out.println("<html><head>");
        out.println("<title>Resultado da gravação do item de venda</title>");
        out.println("</head><body>");
        out.println("<h1>Item de venda gravado!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

# Leitura das variáveis da sessão corrente

Para acessar as variáveis da sessão associada à requisição recebida pelo servlet:

- Chamar o método **getSession** de **HttpServletRequest**.

**HttpSession getSession()**

- Recuperar a variável da sessão utilizando o método **getAttribute** de **HttpSession**.

**Object getAttribute(String key)**

# Exemplo de leitura de dados da sessão

Servlet que lê e imprime a lista com os itens de venda:

```
public class ListarItensServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        HttpSession session = request.getSession();
        out.println("<html><head>");
        out.println("<title>Resultado da gravação do item de venda</title>");
        out.println("</head><body>");
        List<ItemVenda> itens = (List<ItemVenda>) session.getAttribute("itensVendidos");
        if (itens == null) { out.println("<h1>Não há itens na lista!</h1>"); }
        else {
            out.println("<table border=1>");
            for (ItemVenda item: itens) {
                out.println("<tr><td>" + item.getQuantidade() + "</td>");
                out.println("<td>" + item.getDescricao() + "</td></tr>");
            }
            out.println("</table>");
        }
        out.println("</body></html>");
    }
}
```

# Remoção de dados da sessão corrente

Para remover dados da sessão associada à requisição recebida pelo servlet:

- Chamar o método **getSession** de **HttpServletRequest**.

**HttpSession getSession()**

- Chamar o método **removeAttribute** de **HttpSession**, passando como argumento o nome da variável de sessão

**void removeAttribute(String key)**

- Chamar o método **invalidate** de **HttpSession** para remover todas as informações da sessão corrente

**void invalidate()**

# Armazenamento do identificador da sessão

- O mecanismo padrão é o servidor de aplicações armazenar um número identificador da sessão como um **cookie**.
- Caso o navegador esteja com o armazenamento de cookies desabilitado, é necessário armazenar o identificador na URL, utilizando o método **encodeURL** de **HttpServletResponse**. Exemplo:

```
String urlOriginal = "basket";  
String urlCodificada = response.encodeURL(urlOriginal);  
out.println("<A HREF=\"" + urlCodificada + "\">...</A>");
```



# Acesso à sessão em páginas JSP

- Nas páginas JSP pode-se utilizar nos *scriptlets* e nas expressões JSP a variável **session**.

## Exercício

Refaça o exemplo desta apresentação (sobre a inserção e listagem de itens de venda) utilizando a abordagem MVC com servlets e JSP.

**Bom estudo!**

