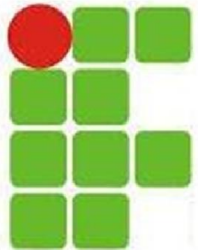
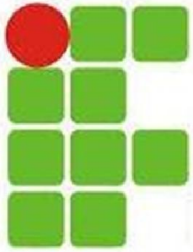

Programação Web

Professor: Diego Oliveira



Conteúdo 03:
HTTP

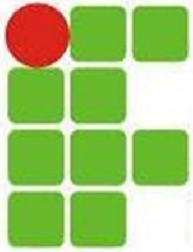




HTTP

- Como já visto, o protocolo HTTP em Java para Web é utilizado através dos objetos Response e Request
- Estes objetos encontram-se no pacote `javax.servlet.http`
- As respostas são dadas pelo objeto `javax.servlet.http.HttpServletResponse`
- E as requisições são feitas pelo objeto `javax.servlet.http.HttpServletRequest`

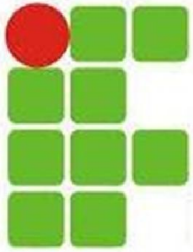




Request

- O Objeto `HttpServletRequest` é criado a cada vez que um cliente faz uma requisição ao JSP Engine
- Este objeto provê métodos para pegar as informações do cabeçalho HTTP:
 - dados
 - cookies
 - métodos
 - outras

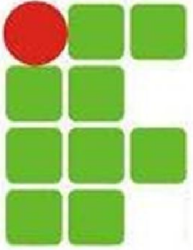




Request

- Os principais métodos do HTTP que podem ser utilizados em um programa JSP são:
 1. `Cookie[] getCookies()`
 2. `Enumeration getAttributeNames()`
 3. `Enumeration getHeaderNames()`
 4. `Enumeration getParameterNames()`
 5. `HttpSession getSession()`
 6. `HttpSession getSession(boolean create)`

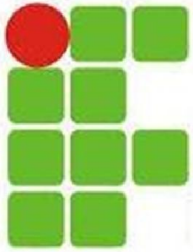




Request

- ...continuando os métodos:
 7. Locale getLocale()
 8. Object getAttribute(String name)
 9. ServletInputStream getInputStream()
 10. String getAuthType()
 11. String getCharacterEncoding()
 12. String getContentType()

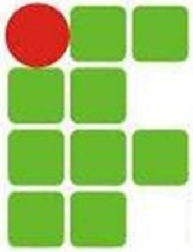




Request

- ...continuando os métodos:
 - 13. String getContextPath()
 - 14. String getHeader(String name)
 - 15. String getMethod()
 - 16. String getParameter(String name)
 - 17. String getPathInfo()
 - 18. String getProtocol()

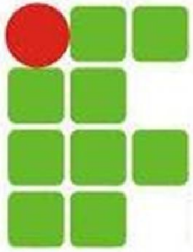




Request

- ...continuando os métodos:
 - 19. String getQueryString()
 - 20. String getRemoteAddr()
 - 21. String getRemoteHost()
 - 22. String getRemoteUser()
 - 23. String getRequestURI()
 - 24. String getRequestedSessionId()

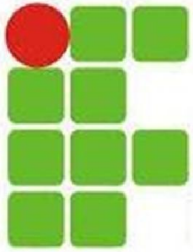




Request

- ...continuando os métodos:
 - 25. String getServletPath()
 - 26. String[] getParameterValues(String name)
 - 27. boolean isSecure()
 - 28. int getContentLength()
 - 29. int getIntHeader(String name)
 - 30. int getServerPort()

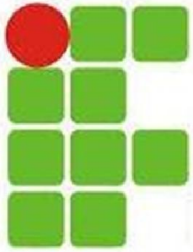




Exemplo de Request

```
10 <html>
11   <head>
12     <title>Exemplo de Leitura do Cabeçalho HTTP</title>
13   </head>
14   <body>
15     <center>
16       <h2>Exemplo de Leitura do Cabeçalho HTTP via Objeto Request</h2>
17       <table width="100%" border="1" align="center">
18         <tr bgcolor="GRAY">
19           <th>Nome do Cabeçalho</th><th>Valor do Cabeçalho</th>
20         </tr>
21         <%
22           Enumeration nomesCabecalhos = request.getHeaderNames();
23           while (nomesCabecalhos.hasMoreElements()) {
24             String nomeParametro = (String) nomesCabecalhos.nextElement();
25             out.print("<tr><td>" + nomeParametro + "</td>\n");
26             String paramValue = request.getHeader(nomeParametro);
27             out.println("<td> " + paramValue + "</td></tr>\n");
28           }
29         %>
30       </table>
31     </center>
32   </body>
33 </html>
```



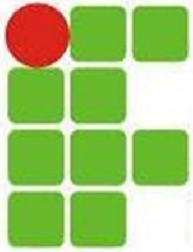


Exemplo de Request

Exemplo de Leitura do Cabeçalho HTTP via Objeto Request

Nome do Cabeçalho	Valor do Cabeçalho
host	localhost:8084
connection	keep-alive
accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
user-agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.111 Safari/537.36
accept-encoding	gzip, deflate, sdch
accept-language	en-US,en;q=0.8,es;q=0.6,fr;q=0.4,pt;q=0.2
cookie	JSESSIONID=1171944ECAC8F124EA9CC5D056F9071D

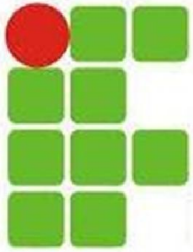




Response

- O objeto Response é uma instância de `javax.servlet.http.HttpServletResponse`
- O servidor cria os objetos Request e também cria os objetos Response
- Através do objeto Response, o programador pode adicionar cookies, datas, códigos de STATUS HTTP e outras informações importantes

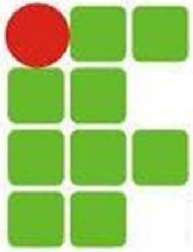




Response

- Os métodos disponíveis para o Response são:
 1. String encodeRedirectURL(String url)
 2. String encodeURL(String url)
 3. boolean containsHeader(String name)
 4. boolean isCommitted()
 5. void addCookie(Cookie cookie)
 6. void addDateHeader(String name, long date)

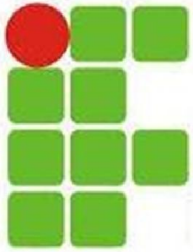




Response

- ...continuando os métodos:
 7. void addHeader(String name, String value)
 8. void addIntHeader(String name, int value)
 9. void flushBuffer()
 10. void reset()
 11. void resetBuffer()
 12. void sendError(int sc)

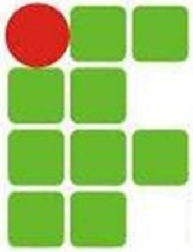




Response

- ...continuando os métodos:
 - 13.void sendError(int sc, String msg)
 - 14.void sendRedirect(String location)
 - 15.void setBufferSize(int size)
 - 16.void setCharacterEncoding(String charset)
 - 17.void setContentLength(int len)
 - 18.void setContentType(String type)

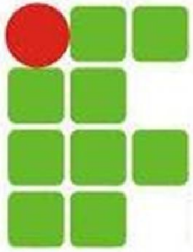




Response

- ...continuando os métodos:
 19. void setDateHeader(String name, long date)
 20. void setHeader(String name, String value)
 21. void setIntHeader(String name, int value)
 22. void setLocale(Locale loc)
 23. void setStatus(int sc)

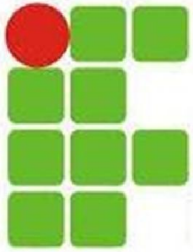




Exemplo de Response

```
8 <html>
9   <head>
10      <title>Cabeçalho Auto-Atualizável</title>
11   </head>
12   <body>
13     <center>
14       <h2>Cabeçalho Auto-Atualizável via Response</h2>
15     <%
16       response.setIntHeader("Refresh",1);
17       Calendar calendario = new GregorianCalendar();
18       String am_pm;
19       int hora = calendario.get(Calendar.HOUR);
20       int minuto = calendario.get(Calendar.MINUTE);
21       int segundo = calendario.get(Calendar.SECOND);
22       if (calendario.get(Calendar.AM_PM) == 0) {
23         am_pm = "AM";
24       } else {
25         am_pm = "PM";
26       }
27       String CT = hora + ":" + minuto + ":" + segundo + " " + am_pm;
28       out.println("Horário atual é: " + CT + "\n");
29     %>
30   </center>
31 </body>
</html>
```

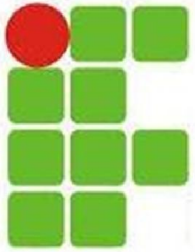




Exercício

- Desenvolva duas páginas JSP
- Em uma delas demonstre a utilização de pelo menos 5 dos métodos apresentados para o Request
- Em outra, demonstre a utilização de pelo menos 5 métodos para o Response, tendo obrigatoriamente que atualizar a página a cada segundo, de maneira que ela alterne a sua apresentação





Perguntas?

