



Prática-01: JEE e Webapps

por Frederico C. G. Pereira

© Copyright 2025

Objetivos

- Usar o container web
- Consultar recursos estáticos e remotos
- Analisar a requisição e resposta HTTP

ATENÇÃO: Não copie e cole os trechos de código, DIGITE-OS VOCÊ MESMO! Isto faz uma gigantesca diferença para o aprendizado e é uma recomendação sempre encontrada em livros de programação. Sempre após digitar, certifique-se de que entendeu o que você fez, do contrário a prática é apenas um exercício de CtrlC CtrlV. Siga corretamente os passos neste roteiro. Obedeça às versões indicadas do que deve ser baixado e não baixe alternativas mais recente porque pode quebrar seu código. Faça estas experiências depois de completar a prática.

1) Baixe a versão mais atual do Apache Tomcat®.

O Apache Tomcat® é a implementação de referência da especificação Java para aplicações web. Sempre que uma nova versão da API de Servlets é lançada, uma nova versão do Tomcat® a implementa integralmente. Este container é muitas vezes embutido em servidores de aplicação mais poderosos para gerir a parte mais básica de uma aplicação web que é a API de Servlets.

- Vá em <https://tomcat.apache.org/download-11.cgi>
- Baixe a versão **zip**, localizada na seção *Binary Distribution* da página de download.
- Descompacte o arquivo zip baixado.
- Pronto, seu container web está instalado!

2) Colocando o container para executar.

- Vá para a pasta **bin/** de onde você descompactou o Tomcat®.
- Localize o arquivo **startup.bat**, um arquivo de script do Windows e clique 2x nele.
- Uma janela de terminal é exibida e nela uma série de mensagens. A última mensagem deve ser similar a "Server startup in [645] milliseconds", que indica que o servidor está no ar. **Mantenha esta janela aberta!**
- Se esta mensagem não for exibida ou se deu algum erro, você precisa garantir que o terminal é capaz de encontrar o JDK a partir de comandos do terminal. Para isso, localize o caminho (*path*) do JDK na sua máquina. Aqui na minha ele está em C:\arquivos de programas\Java\jdk-17. Copie este caminho e edite o arquivo startup.bat para adicionar as linhas a seguir, logo antes da linha contendo **setlocal**:

```
set JAVA_HOME=C:\arquivos de programas\Java\jdk-17 ←cole aqui o path que você copiou  
set PATH=%JAVA_HOME%\bin;%PATH%
```
- Salve o arquivo e abra um terminal¹ no seu diretório e digite startup.bat (esta segunda forma é melhor para ver algum erro que possa surgir).
- Em um navegador, digite: <http://localhost:8080/>.


¹ Você sabia que, para abrir um terminal em um diretório, basta ir para o diretório no Windows Explorer e digitar **cmd** na barra de comandos?




- g) Se o navegador mostrar a página inicial do Tomcat® (abaixo) é sinal que está tudo ok. Um dos problemas que pode ocorrer nesta fase é alguma outra aplicação estar usando a porta 8080 que é padrão do Tomcat®. Neste caso, desligue a aplicação que está usando a porta 8080 e tente novamente. Caso não possa desligar a outra aplicação, você terá que alterar a porta padrão do Tomcat. Veja a seguir como fazer.
- h) Para mudar a porta padrão do Tomcat (8080):
- edite o arquivo `server.xml` localizado dentro da pasta **conf** da pasta raiz do Tomcat.
 - Localize a string "8080" que esteja num elemento xml `<Connector>` descomentado.
 - Substitua o valor 8080 pelo valor que você deseja, como exemplo, 8090 ou 8081.
 - Salve o arquivo `server.xml` e tente executar o `startup.bat` novamente.

[Home](#) [Documentation](#) [Configuration](#) [Examples](#) [Wiki](#) [Mailing Lists](#) [Find Help](#)

Apache Tomcat/11.0.5



If you're seeing this, you've successfully installed Tomcat. Congratulations!



Recommended Reading:
[Security Considerations How-To](#)
[Manager Application How-To](#)
[Clustering/Session Replication How-To](#)

[Server Status](#)
[Manager App](#)
[Host Manager](#)

Developer Quick Start

[Tomcat Setup](#) [Realms & AAA](#) [Examples](#) [Servlet Specifications](#)
[First Web Application](#) [JDBC DataSources](#) [Tomcat Versions](#)

Managing Tomcat

For security, access to the [manager webapp](#) is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 11.0 access to the manager application is split between different users.
[Read more...](#)

[Release Notes](#)
[Changelog](#)
[Migration Guide](#)
[Security Notices](#)

Documentation

[Tomcat 11.0 Documentation](#)
[Tomcat 11.0 Configuration](#)
[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

[Tomcat 11.0 Bug Database](#)
[Tomcat 11.0 JavaDocs](#)
[Tomcat 11.0 Git Repository at GitHub](#)

Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

[tomcat-announce](#)
Important announcements, releases, security vulnerability notifications. (Low volume).

[tomcat-users](#)
User support and discussion

[taglibs-user](#)
User support and discussion for [Apache Taglibs](#)

[tomcat-dev](#)
Development mailing list, including commit messages

Other Downloads

[Tomcat Connectors](#)
[Tomcat Native](#)
[Taglibs](#)
[Deployer](#)

Other Documentation

[Tomcat Connectors](#)
[mod_jk Documentation](#)
[Tomcat Native](#)
[Deployer](#)

Get Involved

[Overview](#)
[Source Repositories](#)
[Mailing Lists](#)
[Wiki](#)

Miscellaneous

[Contact](#)
[Legal](#)
[Sponsorship](#)
[Thanks](#)



Apache Software Foundation

[Who We Are](#)
[Heritage](#)
[Apache Home](#)
[Resources](#)

Copyright ©1999-2025 Apache Software Foundation. All Rights Reserved



3) Utilizando aplicações prontas do Tomcat®.

- Na página principal do Tomcat®, procure por um link **Examples**, na seção azul chamada **Developer Quick Start**.
- Em seguida, clique em **Servlet examples**. Diversos exemplos de aplicações web simples de demonstração e que utilizam diretamente a API de Servlets são exibidas.
- Clique em  [Execute](#) da aplicação *Hello World*.
- Você deverá ver a tradicional mensagem "Ola Mundo!" em uma página em branco. Clique com o botão direito do mouse e selecione *Inspecionar*. Esta página simples não é um arquivo HTML editado por um humano e colocado no Tomcat®, mas sim uma página que foi dinamicamente gerada por uma classe Java especial chamada Servlet. Este é o Servlet mais simples que se pode ter numa aplicação web e você pode ver o seu código fonte.
- Ainda na página "Ola Mundo", clique no botão voltar do navegador e, desta vez, clique em  [Source](#) ainda na aplicação *Hello World*.
- Uma página (que também foi gerada dinamicamente!) exibe o código fonte da classe Servlet que produziu a página "Ola Mundo!". Note que as linhas do código HTML foram geradas uma a uma pela classe, mais especificamente por seu método `doGet()`. Este método é quem atende a requisições HTTP GET enviadas para esta aplicação web. Ela não atende mais a nenhum outro método do protocolo (POST, PUT etc).
- Ainda no código fonte, é de chamar a atenção os parâmetros do método `doGet()`, lá estão, sob a forma de objetos da API de Servlets, um objeto representando a requisição (parâmetro *request*) e outro a resposta gerada (parâmetro *response*). Note que o *request* nem sequer foi usado neste exemplo.
- Volte para a página de exemplos de servlets e desta vez execute a aplicação *Request Info*. Em seguida, veja seu código fonte. Nesta aplicação o parâmetro *request* foi usado para ler diversas informações da requisição, tais como, o método HTTP usado, a URL usada para chegar a esta classe Servlet (`getRequestURI()`), o protocolo HTTP usado dentre outros. Experimente digitar na barra de endereços no navegador, ao final da URL que mostra as informações (não na que mostra o código fonte) o seguinte trecho: `/teste/123` (figura a seguir). O path info é todo pedaço da URL que vem depois do recurso solicitado, neste caso, a classe servlet `RequestInfoExample`. O endereço remoto e o endereço IPv6 do navegador usado (deve mudar, dependendo da máquina usada).



Exemplo da Informacao do Pedido

Método:	GET
URI do Pedido:	/examples/servlets/servlet/RequestInfoExample/teste/123
Protocolo:	HTTP/1.1
Informacao do Caminho:	/teste/123
Endereco Remoto:	0:0:0:0:0:0:1

- Repita o experimento acima executando a aplicação *Request Headers*.

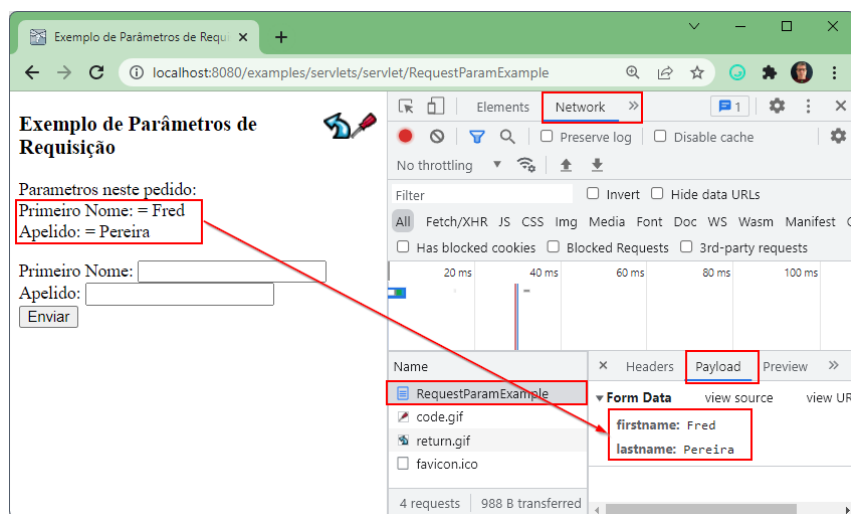


- j) Agora abra a ferramenta de depuração do seu navegador (geralmente associada a tecla F12).
- k) Clique na aba **Network** e limpe o conteúdo da aba.
- l) Dê um *reload* (F5) na página e clique em RequestHeaderExample na aba network.
- m) No painel direito, procure a opção Headers e dentro dela Request Header.
- n) Compare o conteúdo mostrado na ferramenta de debug do navegador com o que é mostrado na página. Veja que esta aplicação apenas gera uma página com os cabeçalhos HTTP da requisição.

The screenshot shows a web browser window with the URL `localhost:8080/examples/servlets/servlet/RequestHeaderExample`. The page content is a list of HTTP headers. The Network tab in the developer tools is open, showing a request to the same URL. The Headers panel on the right displays the request headers, including:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
- Accept-Encoding: gzip, deflate, br
- Accept-Language: pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7
- Cache-Control: max-age=0
- Connection: keep-alive
- Cookie: _ga=GA1.1.443463066.1626285411
- Host: localhost:8080
- Referer: http://localhost:8080/examples/servlets/
- Sec-CH-UA: "Not A;Brand";v="99", "Chromium";v="98", "Google Chrome";v="98"
- Sec-CH-UA-Mobile: ?0
- Sec-CH-UA-Platform: "Windows"
- Sec-Fetch-Dest: document
- Sec-Fetch-Mode: navigate
- Sec-Fetch-Site: same-origin
- Sec-Fetch-User: ?1
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4759.70 Safari/537.36

- o) Execute a aplicação *Request Parameters*. Nos campos do formulário apresentado digite seu nome e sobrenome (apelido em espanhol). Repita o experimento com o depurador do navegador ligado (F12). Eis o meu experimento:



Note que os dados foram enviados no *payload*, isto é, depois da área de cabeçalhos da requisição HTTP. Os nomes **firstname** e **lastname** devem ser os atributos **id** ou **name** dos campos input do formulário. Você pode verificar isto clicando com o botão direito sobre os campos de entrada de texto (*inputs*) e selecionando Inspeccionar. Confirme que o método usado no formulário é POST, por isto os dados foram no *payload* da requisição e não na URL (ficariam visíveis na barra de endereços do navegador).

- p) Por fim, execute a aplicação *Cookies*. Informe o nome **testeCookie** para o campo nome e **12345** para o campo valor do formulário e tecle Enviar.
- q) Agora clique com o botão direito e selecione *Inspeccionar*. Localize e clique na aba **Application** (barra superior da ferramenta de depuração) e depois expanda a opção **Cookies** no painel esquerdo e, sob ela, a opção **http://localhost:8080**. Um cookie de nome **testeCookie** com valor **12345** deve ser exibido. Você pode encontrar outros cookies além deste que você criou. Clique com o botão direito sobre a opção **http://localhost:8080** e selecione **Clear**. Em seguida, dê enter na barra de endereços do navegador para gerar novo *request* na URL do servlet (não use F5 para recarregar a página porque não vai funcionar!).
- 4) Vamos criar uma aplicação web com um servlet nosso? Na IDE é muito mais fácil, mas podemos fazer no braço, como faziam os Incas e Astecas. Para facilitar, eu vou fornecer o Servlet já compilado, você só vai precisar implantá-lo na aplicação web. (O arquivo br.zip a ser usado deve ser baixado do classroom, no mesmo local deste roteiro).
 - a) Baixe e descompacte o arquivo br.zip disponível no mesmo local em que você baixou este roteiro.
 - b) Pare o Tomcat. Neste caso, faça como no passo (5) deste roteiro.
 - c) Na pasta **webapps** do Tomcat, crie uma pasta **teste**.
 - d) Dentro da pasta teste, crie uma pasta **WEB-INF** (exatamente assim)
 - e) Dentro da pasta WEB-INF, crie uma pasta **classes** (no plural!)
 - f) Agora copie a pasta **br/** que foi criada ao deszipar o arquivo br.zip para dentro da pasta classes. Pronto! Está implantada nossa aplicação web com um componente dinâmico. Se você seguir dentro da pasta br até o final, verá um arquivo bytecode AloServlet.class.
 - g) Agora suba novamente o Tomcat e digite: <http://localhost:8080/teste/alo>



- h) Siga o que ele pede a você.
 - i) Para criar um recurso estático, vá para a pasta **teste** e crie ali um arquivo chamado `index.html` com um conteúdo qualquer da sua preferência.
 - j) No navegador, digite: **`http://localhost:8080/teste`**
- 5) Encerre o Tomcat©.
- a) Para finalizar o container, ou você tecla Ctrl+C na janela de terminal que ficou aberta no exercício 2), ou
 - b) Na pasta **bin/** do Tomcat©, procure o arquivo **shutdown.bat** e clique 2x sobre ele. O servidor é finalizado e a janela do terminal que fez o *startup* é fechada.
 - c) Você pode confirmar que o servidor está "fora do ar" tentando ir para sua página principal em `http://localhost:8080` ou teclando F5 na última aplicação executada no navegador.