

Java Servlets

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

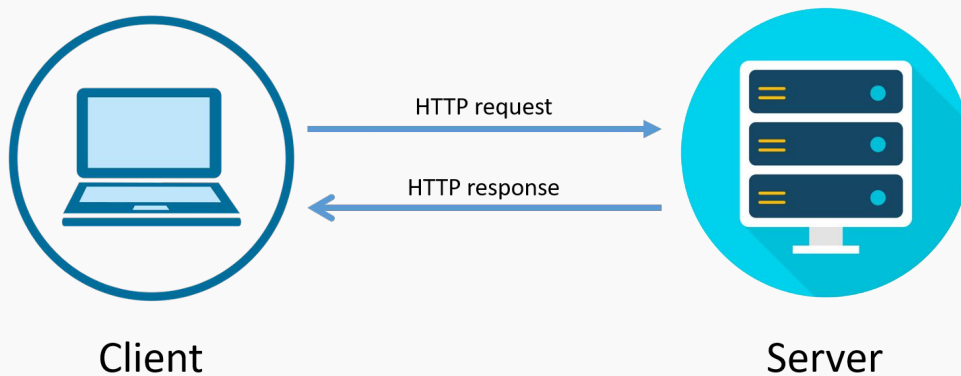
AGENDA

- Conceito
- Protocolo WEB
- Criação do primeiro projeto

Java Servlets

Tecnologia que representa a base de desenvolvimento web em Java.

Servlets é uma classe Java executada em um servidor responsável por atender requisições de um cliente e devolver as respostas de acordo com cada requisição.



Request

Uma requisição temos algumas características importantes, como o endereço do endpoint, os headers e, dependendo do método utilizado, o body na qual enviaremos o payload da requisição.

Em cada chamada, utilizaremos os métodos como por exemplo:

Endpoint	Método	Ação
/users	GET	Retorna a lista de usuários
/users	POST	Insere um novo usuário
/users/{id}	GET	Retorna o usuário com id = {id}
/users/{id}	PUT	Substitui os dados do usuário com id = {id}
/users/{id}	PATCH	Altera itens dos dados do usuário com id = {id}
/users/{id}	DELETE	Remove o usuário com id = {id}

Retornará para o cliente os dados juntamente os headers e status code, que é importante para efetuarmos os tratamentos necessários no frontend.

HTTP Status Codes



O que vamos precisar?

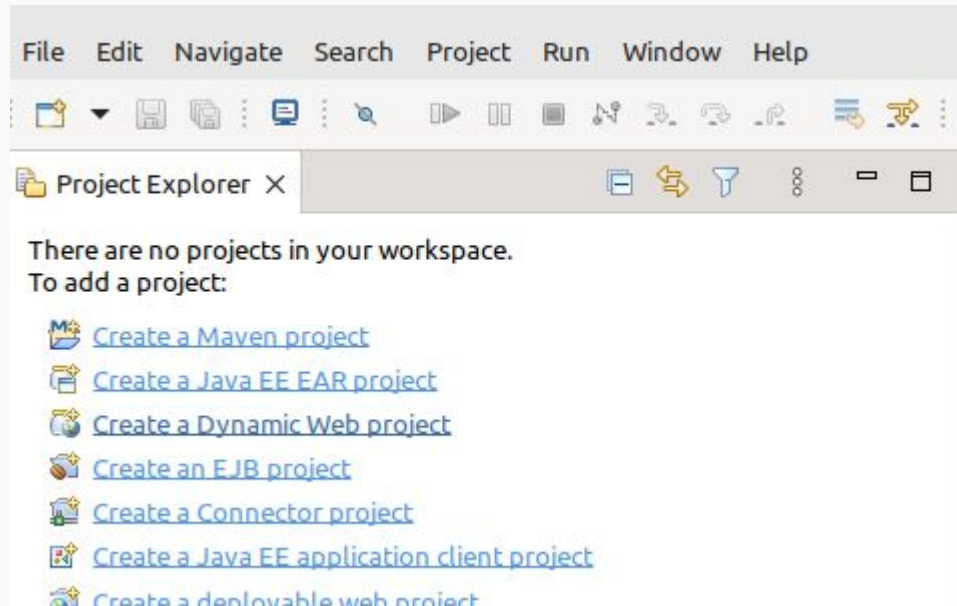
Para iniciarmos com Servlet, irem precisar:

- Eclipse for Web Development
- Apache TomCat



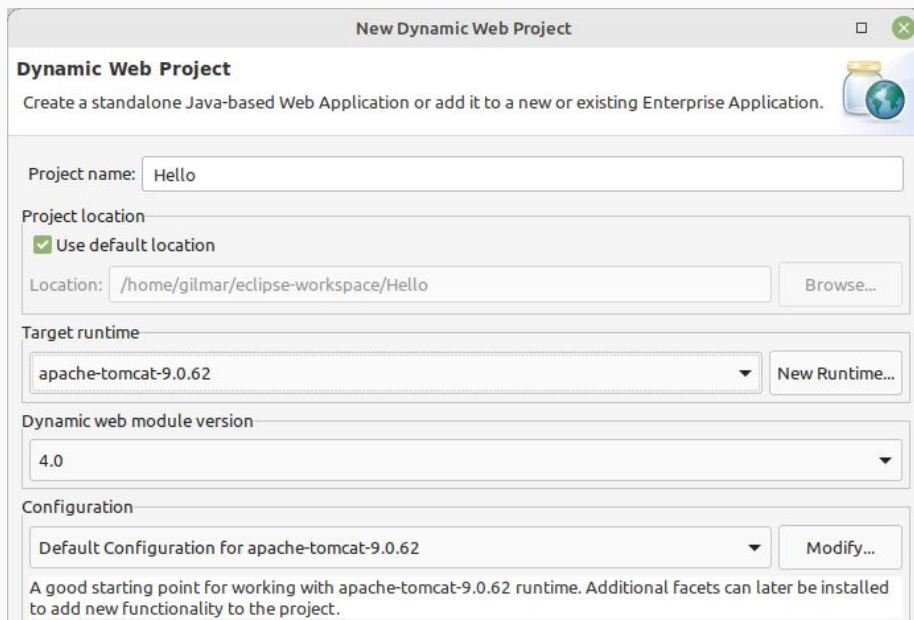
Primeiro projeto

Primeiramente vamos criar um projeto do tipo “Dynamic Web project”



Primeiro projeto

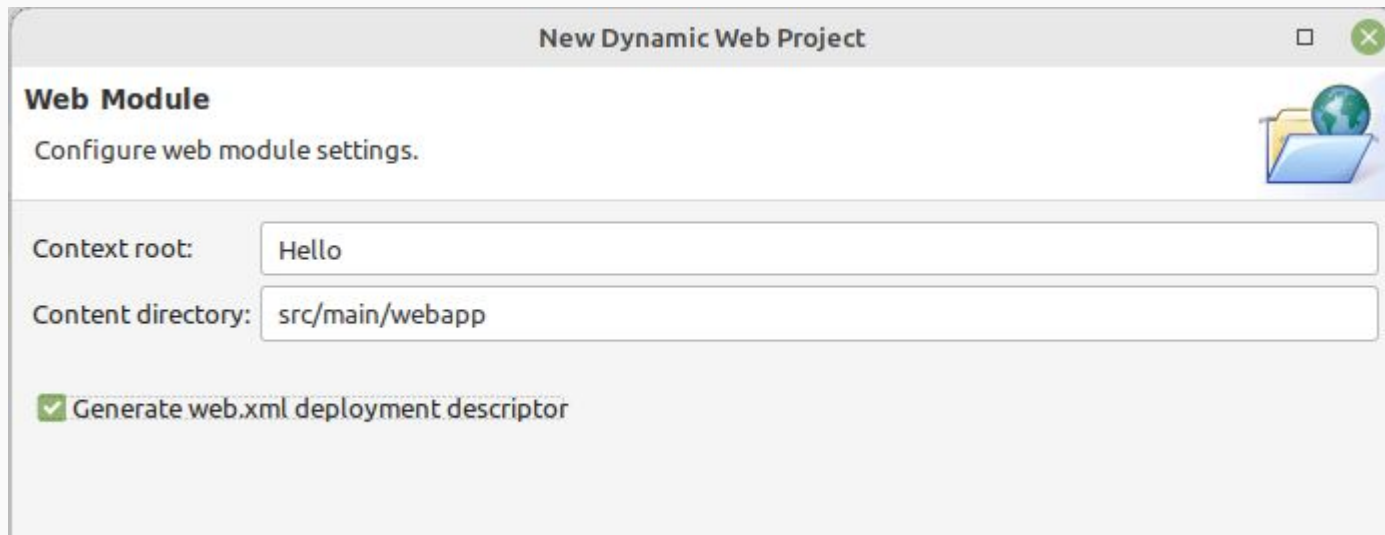
Iremos dar o nome e selecionar um servidor TomCat (Se não houver, será necessário o download pela própria IDE)



The screenshot shows the 'New Dynamic Web Project' dialog box in the Eclipse IDE. The dialog has a title bar 'New Dynamic Web Project' with standard window controls. Below the title bar, the text 'Dynamic Web Project' is followed by a description: 'Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.' and a small icon of a jar and a globe. The dialog is divided into several sections: 'Project name:' with a text field containing 'Hello'; 'Project location:' with a checked checkbox 'Use default location' and a text field showing the path '/home/gilmar/eclipse-workspace/Hello' next to a 'Browse...' button; 'Target runtime:' with a dropdown menu showing 'apache-tomcat-9.0.62' and a 'New Runtime...' button; 'Dynamic web module version:' with a dropdown menu showing '4.0'; and 'Configuration:' with a dropdown menu showing 'Default Configuration for apache-tomcat-9.0.62' and a 'Modify...' button. At the bottom, there is a note: 'A good starting point for working with apache-tomcat-9.0.62 runtime. Additional facets can later be installed to add new functionality to the project.'

Primeiro projeto

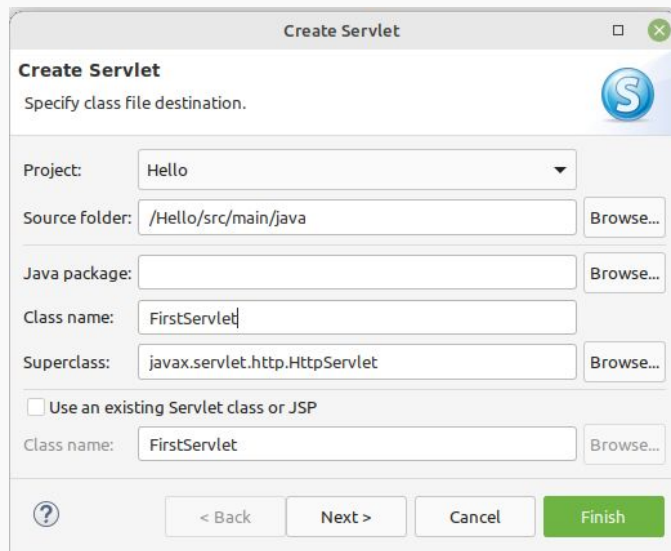
Iremos dar “next” até a última tela onde terá a opção “Generate web.xml deployment descriptor”, que será as configurações do projeto:



The screenshot shows a dialog box titled "New Dynamic Web Project". Inside, there is a section labeled "Web Module" with the instruction "Configure web module settings." and a folder icon. Below this, there are two text input fields: "Context root:" with the value "Hello" and "Content directory:" with the value "src/main/webapp". At the bottom, there is a checked checkbox labeled "Generate web.xml deployment descriptor".

Primeiro projeto

Com o botão direito no package principal vamos em: New > Servlet, gerando nossa primeira Servlet:



The screenshot shows the 'Create Servlet' dialog box. The title bar says 'Create Servlet'. Below the title bar, it says 'Create Servlet' and 'Specify class file destination.' There is a blue 'S' icon in the top right corner. The dialog has several fields: 'Project:' with a dropdown menu showing 'Hello'; 'Source folder:' with a text field containing '/Hello/src/main/java' and a 'Browse...' button; 'Java package:' with a text field and a 'Browse...' button; 'Class name:' with a text field containing 'FirstServlet'; 'Superclass:' with a text field containing 'javax.servlet.http.HttpServlet' and a 'Browse...' button. There is a checkbox labeled 'Use an existing Servlet class or JSP' which is unchecked. Below this, there is a 'Class name:' field with 'FirstServlet' and a 'Browse...' button. At the bottom, there are four buttons: a help button with a question mark, '< Back', 'Next >', and 'Cancel'. A green 'Finish' button is on the right.

Primeiro projeto

Após isso, irá gerar nossa primeira Servlet com método GET e POST:

```
4
30 import java.io.IOException;
9
10 /**
11  * Servlet implementation class FirstServlet
12  */
13 @WebServlet("/FirstServlet")
14 public class FirstServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     /**
18      * @see HttpServlet#HttpServlet()
19      */
20     public FirstServlet() {
21         super();
22         // TODO Auto-generated constructor stub
23     }
24
25     /**
26      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
27      */
28     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
29         // TODO Auto-generated method stub
30         response.getWriter().append("Served at: ").append(request.getContextPath());
31     }
32
33     /**
34      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
35      */
36     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
37         // TODO Auto-generated method stub
38         doGet(request, response);
39     }
40
41 }
42
```

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>