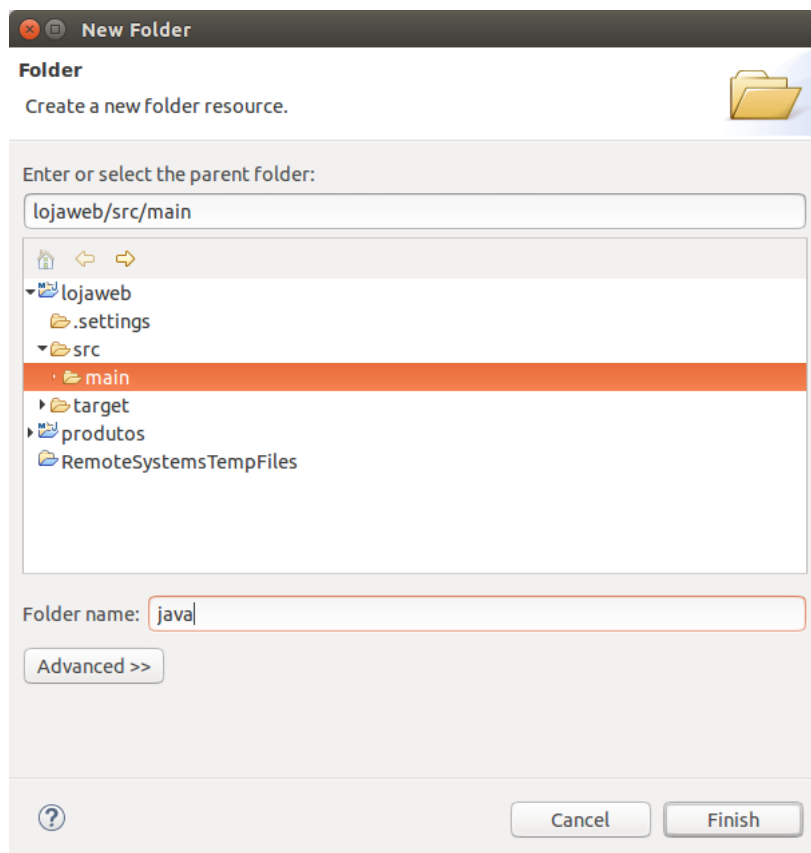


## Testando o hot deploy

Agora vamos criar uma `Servlet` para que no próximo exercício seja possível testar algumas outras configurações do *plugin* do Jetty.

Você pode aprender mais sobre o assunto no nosso curso [Servlet Parte 1: Fundamentos da programação web Java](https://cursos.alura.com.br/course/servlet-3-e-fundamentos-web) (<https://cursos.alura.com.br/course/servlet-3-e-fundamentos-web>).

Como vimos no projeto anterior ( `produtos` ), os códigos das classes ficam no diretório `src/main/java` . Dentro de `src/main` , crie `java` . Clique com o botão direito no diretório `main` , escolha as opções `New > Folder` . A seguinte janela será exibida, na qual digitaremos o nome do diretório:



Clique no projeto, no Explorer do Eclipse e utilize a tecla "F5" para atualizar o projeto. A pasta criada deve ser reconhecida como uma *source folder* dentro de "Java Resources", na perspectiva Java EE do Eclipse.

Na *source folder* `src/main/java` , dentro do pacote `br.com.alura.maven.lojaweb` , crie a `Servlet` `ContatoServlet` (Botão direito em `src/main/java` , "New > Class").

**New Java Class**

Create a new Java class.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

Ela deverá conter o seguinte conteúdo:

```
package br.com.alura.maven.lojaweb;

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

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

@WebServlet(urlPatterns={"/contato"})
public class ContatoServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException {
        PrintWriter writer = resp.getWriter();
        writer.println("<html><h2>Entre em contato</h2></html>");
        writer.close();
    }
}
```

Caso esteja com o Jetty "startado", encerre utilizando "Ctrl + C". No Windows, após o atalho, ele perguntará se você deseja finalizar o arquivo em lotes. Digite "S" e pressione "Enter".

Configure o Jetty para que as alterações nas classes sejam carregadas automaticamente de tempo em tempo, para evitar que pausemos e iniciemos o servidor sempre que fizermos alguma alteração nas classes. Configure o tempo em 10 segundos. Você pode obter ajuda [aqui \(http://www.eclipse.org/jetty/documentation/current/jetty-maven-plugin.html#jetty-run-goal\)](http://www.eclipse.org/jetty/documentation/current/jetty-maven-plugin.html#jetty-run-goal).

Após realizar as configurações, inicie o servidor, acesse `localhost:8080/contato` e veja a mensagem `Entre em contato` aparecer. Troque a mensagem na classe e aguarde uns dez segundos. Atualize a página e veja que a mensagem mudou.