

[Community \(/community/index.html\)](/community/index.html)[Participate \(/participate/index.html\)](/participate/index.html)[Apache
NetBeans
\(/\)](#)[Blog \(https://blogs.apache.org/netbeans/\)](https://blogs.apache.org/netbeans/)[Get Help \(/help/index.html\)](/help/index.html)[Plugins \(https://plugins.netbeans.apache.org/\)](https://plugins.netbeans.apache.org/)[Download \(/download/index.html\)](/download/index.html)

Latest release

Apache NetBeans 12.4

[Find out more \(/download/nb124/nb124.html\)](/download/nb124/nb124.html)

Criando uma aplicação Web Simples Usando um Banco de Dados MySQL



This tutorial needs a review. You can [open a JIRA issue \(https://issues.apache.org/jira/projects/NETBEANS/issues\)](https://issues.apache.org/jira/projects/NETBEANS/issues), or [edit it in GitHub \(https://github.com/apache/netbeans-website/blob/master/netbeans.apache.org/src/content/kb/docs/web/mysql-webapp_pt_BR.asciidoc\)](https://github.com/apache/netbeans-website/blob/master/netbeans.apache.org/src/content/kb/docs/web/mysql-webapp_pt_BR.asciidoc) following these [contribution guidelines. \(/kb/docs/contributing.html\)](/kb/docs/contributing.html)

Escrito por Troy Giunipero

Este documento descreve como criar uma aplicação Web simples que se conecta ao servidor de banco de dados MySQL. Também aborda algumas ideias e tecnologias básicas em desenvolvimento da Web, como [JavaServer Pages \(http://www.oracle.com/technetwork/java/overview-138580.html\)](http://www.oracle.com/technetwork/java/overview-138580.html) (JSP), [JavaServer Pages Standard Tag Library \(http://www.oracle.com/technetwork/java/index-jsp-135995.html\)](http://www.oracle.com/technetwork/java/index-jsp-135995.html) (JSTL), a API [JDBC \(http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html\)](http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html) (Java Database Connectivity) e duas camadas, arquitetura de servidor do cliente. Este tutorial está destinado a iniciantes com conhecimento básico em desenvolvimento Web e que gostariam de utilizar seu conhecimento com o banco de dados MySQL.

O [MySQL \(http://www.mysql.com\)](http://www.mysql.com) é um sistema de gerenciamento de banco de dados de código-fonte aberto popular geralmente utilizado em aplicações Web devido à sua velocidade, flexibilidade e confiabilidade. O MySQL emprega a linguagem SQL, ou *Structured Query Language*, para acessar e processar os dados contidos em bancos de dados.

Este tutorial é a continuação do tutorial [Conectando a um Banco de Dados MySQL \(/ide/mysql.html\)](/ide/mysql.html) e pressupõe que um banco de dados MySQL denominado MyNewDatabase já tenha sido criado, para o qual você registrou uma conexão no NetBeans IDE. Os dados de tabela usados nesse tutorial estão em [ifpwafcad.sql \(https://netbeans.org/projects/samples/downloads/download/Samples%252Fjava%2520Web%252Fifpwafcad.sql\)](https://netbeans.org/projects/samples/downloads/download/Samples%252Fjava%2520Web%252Fifpwafcad.sql) e também são necessários para este tutorial. Este arquivo SQL cria duas tabelas, Subject e Counselor, a seguir, preenche-as com os dados de amostra. Se necessário, salve este arquivo no seu computador, a seguir, abra-o no NetBeans IDE e execute-o no banco de dados MySQL com o nome MyNewDatabase.

[Planejando a Estrutura](#)[index.jsp](#)[response.jsp](#)[Criando um Novo Projeto](#)[Preparando a Interface da Web](#)[Configurando a página de boas-vindas](#)[Criando a página de resposta](#)[Criando uma folha de estilo](#)[Preparando a Comunicação entre a Aplicação e o Banco de Dados](#)[Configurando uma fonte de dados JDBC e o pool de conexões](#)[Fazendo referência à fonte de dados da aplicação](#)[Adicionando o arquivo JAR do driver do banco de dados ao servidor](#)[Adicionando Lógica Dinâmica](#)[Adicionando a biblioteca JSTL ao classpath do projeto](#)

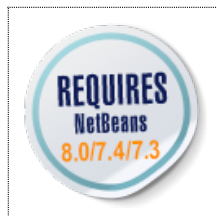


Figure 1. O conteúdo desta página se aplica ao NetBeans IDE 7.2, 7.3, 7.4 e 8.0

Para seguir este tutorial, os recursos e softwares a seguir são necessários.

Software ou Recurso	Versão Necessária
NetBeans IDE (https://netbeans.org/downloads/index.html)	Pacote Java EE 7.2, 7.3, 7.4, 8.0
JDK (Java Development Kit) (http://www.oracle.com/technetwork/java/javase/downloads/index.html)	versão 7 ou 8
Servidor de banco de dados MySQL (http://dev.mysql.com/downloads/mysql/)	5.x
Driver MySQL Connector/J JDBC	versão 5.x
GlassFish Server Open Source Edition	3.x ou 4.x

Observações:

- O pacote de download Java do NetBeans IDE permite instalar o GlassFish Server. É necessário ter o GlassFish Server para trabalhar neste tutorial.
- O [Driver MySQL Connector/J JDBC \(http://dev.mysql.com/downloads/connector/j/\)](http://dev.mysql.com/downloads/connector/j/), necessário para a comunicação entre as plataformas Java e o protocolo do banco de dados MySQL, está incluído no NetBeans IDE.
- Se você precisar comparar seu projeto com uma solução que funciona, [faça download da aplicação de amostra \(https://netbeans.org/projects/samples/downloads/download/Samples%252Fjava%2520Web%252FIFPWAFCAD.zip\)](https://netbeans.org/projects/samples/downloads/download/Samples%252Fjava%2520Web%252FIFPWAFCAD.zip).

Planejando a Estrutura

Aplicações Web simples podem ser projetadas utilizando uma *arquitetura de duas camadas*, na qual um cliente se comunica diretamente com um servidor. Neste tutorial, uma aplicação Java Web se comunica diretamente com um banco de dados MySQL utilizando a API de Conectividade do Banco de Dados Java. Em essência, é o [Driver MySQL Connector/J JDBC \(http://dev.mysql.com/downloads/connector/j/\)](http://dev.mysql.com/downloads/connector/j/) que permite a comunicação entre o código Java entendido pelo servidor de aplicações (o GlassFish Server) e qualquer conteúdo da SQL, a linguagem entendida pelo servidor do banco de dados (MySQL).

A aplicação construída neste tutorial envolve a criação de duas páginas JSP. Em cada uma dessas páginas, você utiliza HTML e CSS para implementar uma interface simples e aplica a tecnologia JSTL para executar a lógica que consulta diretamente o banco de dados e insere os dados recuperados nas duas páginas. As duas tabelas do banco de dados, `Subject` e `Counselor`, estão contidas no banco de dados MySQL, `MyNewDatabase`, criado

[Implementando o código JSTL](#)

[Executando a Aplicação Pronta](#)

[Solução de Problemas](#)

[Existem recursos do banco de dados?](#)

[O pool de conexões e a fonte de dados existem no servidor?](#)

[O driver MySQL Connector/J está acessível para o GlassFish Server?](#)

[O banco de dados é protegido por senha?](#)

[As propriedades do pool de conexões estão definidas corretamente?](#)

[Consulte Também](#)

Criando uma aplicação Web Simples Usando um Banco de Dad... https://netbeans.apache.org/kb/docs/web/mysql-webapp_pt_BR...
com a conclusão do tutorial [Conectando ao Banco de Dados MySQL \(../ide/mysql.html\)](#). Considere o cenário de duas camadas a seguir.

image::images/ifpwafcad-structure.png[title="Estrutura de exemplo de uma aplicação Web de duas camadas"]

A página de boas-vindas (`index.jsp`) apresenta ao usuário um form HTML simples. Quando um browser requer `index.jsp` , o código JSTL da página inicia uma consulta em `MyNewDatabase` . Ele recupera dados da tabela do banco de dados `Subject` e os insere na página, antes de eles serem enviados ao browser. Quando o usuário submete sua seleção no form HTML da página de boas-vindas, a submissão inicia uma solicitação para a página de resposta (`response.jsp`). Novamente, o código JSTL da página inicia uma consulta em `MyNewDatabase` . Neste momento, ele recupera dados das tabelas `Subject` e `Counselor` e os insere na página, permitindo que o usuário exiba os dados com base na sua seleção quando a página é retornada para o browser.

Para implementar o cenário descrito acima, desenvolva uma aplicação simples para uma organização fictícia com o nome IFPWAFCAD, Associação Internacional de Ex-Lutadores Profissionais para Aconselhamento e Desenvolvimento.

index.jsp



Figure 2. `index.jsp` exibido em um browser

response.jsp

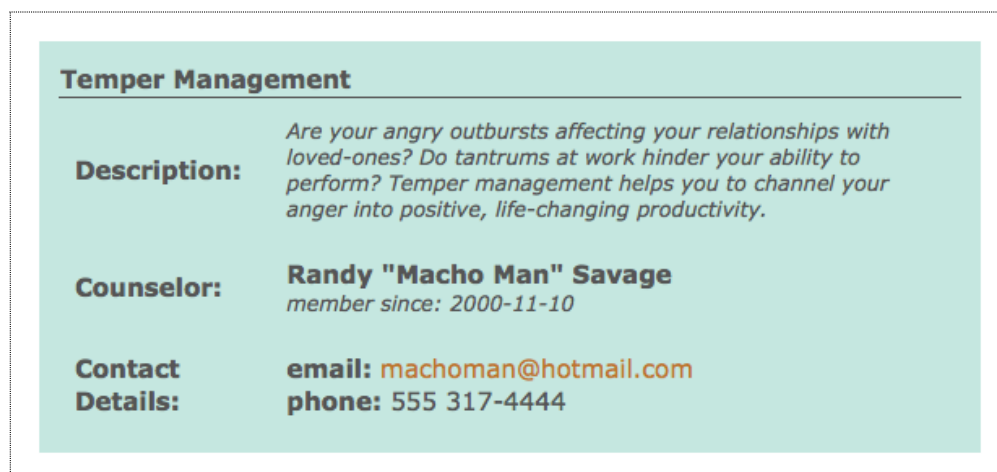


Figure 3. `response.jsp` exibido em um browser

Criando um Novo Projeto

Comece criando um novo projeto Java Web no IDE:

1. Escolha Arquivo > Novo Projeto (Ctrl-Shift-N; ⌘-Shift-N no Mac) no menu principal. Selecione a

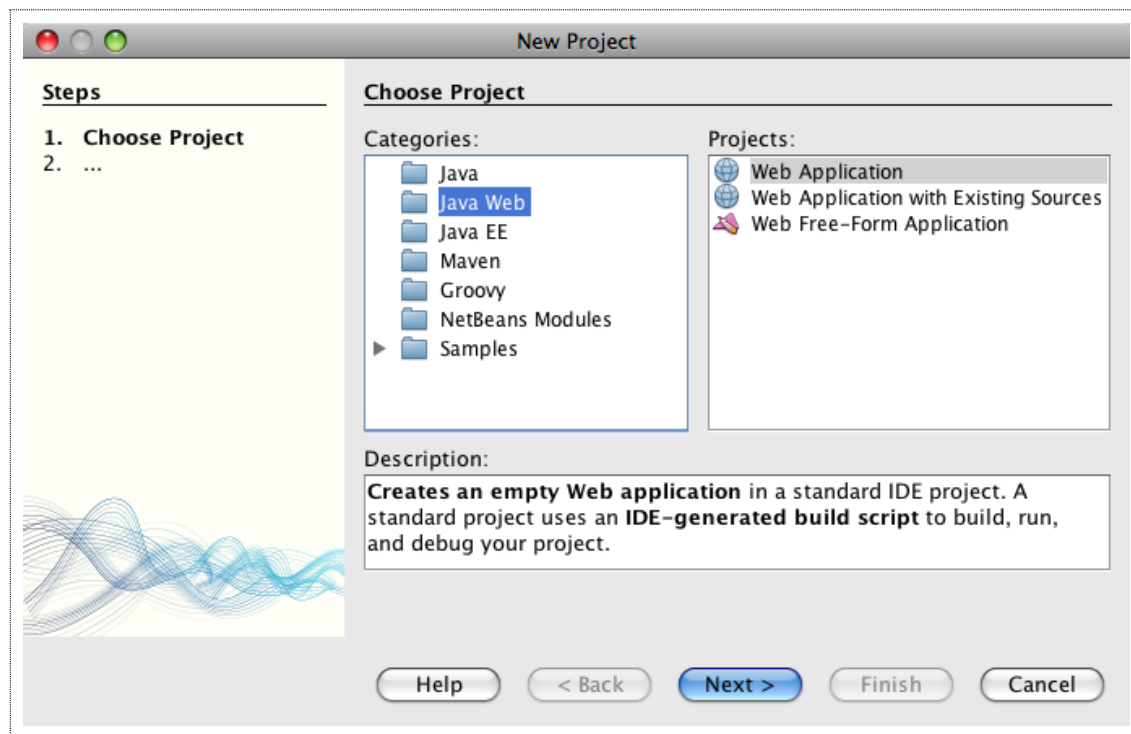


Figure 4. Utilize o assistente de Novo Projeto do IDE para criar um novo projeto

O assistente de Novo Projeto permite criar uma aplicação Web vazia em um projeto IDE padrão. O projeto padrão utiliza um script de construção Ant gerado pelo IDE para compilar, implantar e executar a aplicação.

1. Em Nome do Projeto, digite **IPWAFCAD**. Além disso, especifique o local para o projeto no computador. (Por default, o IDE coloca projetos em uma pasta `NetBeansProjects` localizada no diretório home). Clique em Próximo.
2. No painel Servidor e Configurações, especifique o GlassFish Server como o servidor que será utilizado para executar a aplicação.

*Observação.*O GlassFish Server será exibido no campo drop-down Servidor se você tiver instalado a versão Java do [NetBeans IDE \(https://netbeans.org/downloads/index.html\)](https://netbeans.org/downloads/index.html). Como o GlassFish Server está incluído no download, ele é automaticamente registrado no IDE. Se você quiser utilizar um servidor diferente para esse projeto, clique no botão Adicionar localizado ao lado do campo drop-down Servidor e registre um servidor diferente no IDE. No entanto, trabalhar com servidores que não o GlassFish Server está além do escopo deste tutorial.

1. No campo Versão do Java EE, selecione **Java EE 5**.

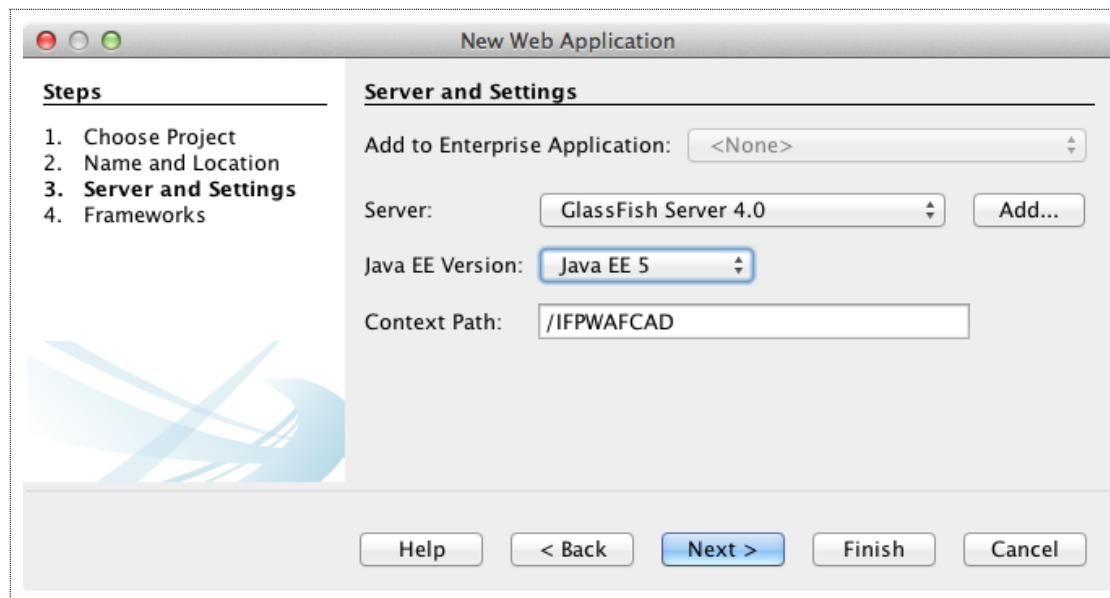


Figure 5. Especificar configurações de servidor no assistente de Nova Aplicação Web

Os projetos da Web Java EE 6 e Java EE 7 não exigem o uso do descritor de implantação `web.xml` e o modelo de projeto do NetBeans não inclui o arquivo `web.xml` em projetos Java EE 6 e Java EE 7. No entanto, este tutorial demonstra como declarar uma fonte de dados no descritor de implantação e não conta com nenhuma funcionalidade específico do Java EE 6 ou Java EE 7, portanto, é possível definir a versão do projeto como Java EE 5.

Observação. Você pode igualmente definir a versão do projeto como Java EE 6 ou Java EE 7 e, em seguida, criar um descritor de implantação `web.xml`. (No assistente de Novo Arquivo, selecione a categoria Web e, em seguida, Descritor de Implantação Padrão).

1. Clique em Finalizar. O IDE cria um modelo de projeto para toda a aplicação e abre uma página JSP vazia (`index.jsp`) no editor. O arquivo `index.jsp` serve como página de boas-vindas da aplicação.

Preparando a Interface da Web

Comece preparando as páginas de boas-vindas (`index.jsp`) e de resposta (`response.jsp`). A página de boas-vindas implementa um form HTML que é utilizado para capturar dados do usuário. As duas páginas implementam uma tabela HTML para exibir os dados de forma estruturada. Nesta seção, você também criará uma folha de estilo que irá melhorar a aparência das duas páginas.

- [Configurando a Página de Boas-Vindas](#)
- [Criando a página de resposta](#)
- [Criando uma folha de estilo](#)

Configurando a página de boas-vindas

Confirme se `index.jsp` está aberto no editor. Se o arquivo ainda não estiver aberto, clique duas vezes no `index.jsp` no nó das Páginas Web no projeto IFPWAFCAD na janela Projetos.

1. No editor, altere o texto entre as tags `<title>` para Homepage da IFPWAFCAD .
2. Altere o texto entre as tags `<h1>` para: Bem-vindo à IFPWAFCAD, a Associação internacional de Ex-Lutadores Profissionais para Aconselhamento e Desenvolvimento! .
3. Abra a Paleta do IDE selecionando Janela > Paleta (Ctrl-Shift-8; ⌘-Shift-8 no Mac) no menu principal. Coloque o ponteiro do mouse sobre o ícone Tabela na categoria HTML e observe que o snippet de código default do item é exibido.

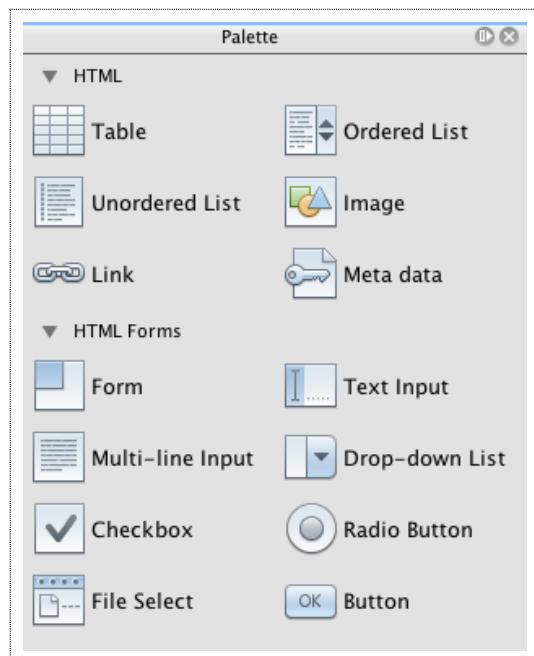



Figure 6. A Paleta exibe o snippet de código ao passar o mouse sobre um item

Você pode configurar a Paleta conforme desejar: clique com o botão direito do mouse na Paleta e selecione Mostrar Ícones Grandes e Ocultar Nomes dos Itens para que ela seja exibida como na imagem acima.

1. Coloque o cursor em um ponto logo depois das tags `<h1>` . (É aí que você deseja implementar a nova tabela HTML.) Em seguida, na Paleta, clique duas vezes no ícone Tabela.
2. Na caixa de diálogo Inserir Tabela exibida, especifique os seguintes valores e clique em OK:
 - **Linhas:** 2
 - **Colunas:** 1
 - **Tamanho da Borda:** 0 O código da tabela HTML é gerado e adicionado a sua página.
3. Adicione o conteúdo a seguir ao cabeçalho da tabela e à célula da primeira linha da tabela (novo conteúdo mostrado em **negrito**):

```
<table border="0">
  <thead>
    <tr>
      <th>*IFPWAFCAD offers expert counseling in a wide range of fields.*</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>*To view the contact details of an IFPWAFCAD certified former
        professional wrestler in your area, select a subject below:*</td>
    </tr>
```

1. Na linha inferior da tabela, insira um form HTML. Para isso, coloque o cursor entre o segundo par de tags `<td>` , em seguida, clique duas vezes no ícone do form HTML () na Paleta. Na caixa de diálogo Inserir Form, digite `response.jsp` no campo Texto da Ação e, em seguida, clique em OK.

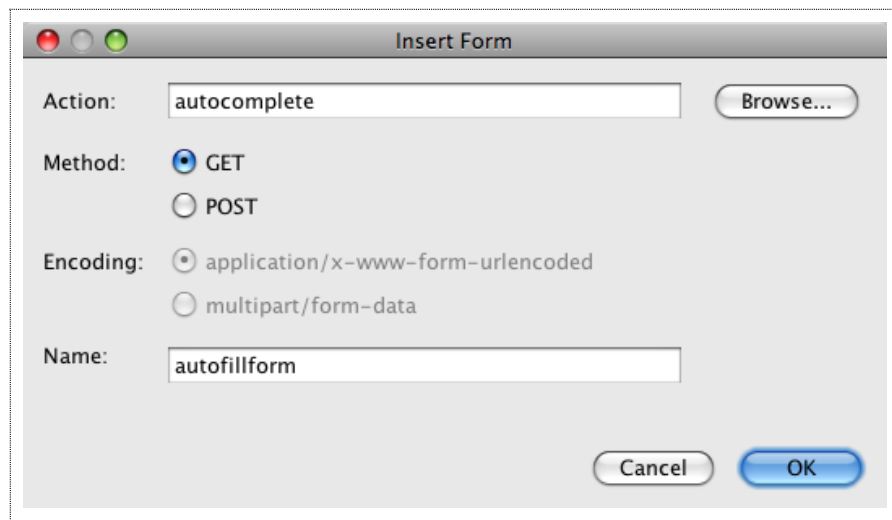



Figure 7. Especificar as configurações do form na caixa de diálogo Inserir Form

1. Digite o conteúdo a seguir entre as tags `<form>` (novo conteúdo mostrado em **negrito**):

```
<tr>
  <td>
    <form action="response.jsp">
      *<strong>Select a subject:</strong>*
    </form>
  </td>
</tr>
```

1. Pressione Enter para adicionar uma linha vazia após o conteúdo que você acabou de adicionar e, em seguida, clique duas vezes na Lista Drop-down na Paleta para abrir a caixa de diálogo Inserir Lista Drop-down.
2. Digite `subject_id` para o campo de texto Nome na caixa de diálogo Inserir Lista Drop-down e clique em OK. Observe que o snippet de código da lista drop-down será adicionado ao form.

O número de opções da lista drop-down não é importante no momento. Mais adiante neste tutorial, você adicionará tags JSTL que irão gerar dinamicamente as opções com base nos dados coletados da tabela do banco de dados Subject.

1. Adicione um item de botão Submeter () ao ponto bem ao lado da lista drop-down adicionada. É possível utilizar a Paleta para fazer isso ou chamar a funcionalidade autocompletar código do editor, conforme ilustrado na etapa anterior. Na caixa de diálogo Inserir Botão, digite `submit` nos campos de texto Nome e Label e clique em OK.
2. Para formatar seu código, clique com o botão direito do mouse no editor e selecione Formatar (Alt-Shift-F; Ctrl-Shift-F no Mac). Seu código será formatado automaticamente e deverá ter a aparência semelhante ao seguinte código:


```

<body>
  <h2>Welcome to <strong>IFPWAFCAD</strong>, the International Former
    Professional Wrestlers' Association for Counseling and Development!
  </h2>

  <table border="0">
    <thead>
      <tr>
        <th>IFPWAFCAD offers expert counseling in a wide range of fields.</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>To view the contact details of an IFPWAFCAD certified former
          professional wrestler in your area, select a subject below:</td>
      </tr>
      <tr>
        <td>
          <form action="response.jsp">
            <strong>Select a subject:</strong>
            <select name="subject_id">
              <option></option>
            </select>
            <input type="submit" value="submit" name="submit" />
          </form>
        </td>
      </tr>
    </tbody>
  </table>
</body>

```

Para exibir essa página em um browser, clique com o botão direito do mouse no editor e selecione Executar Arquivo (Shift-F6; Fn-Shift-F6 no Mac). Quando você fizer isso, a página JSP será automaticamente compilada e implantada no servidor. O IDE será aberto no browser default para exibir a página do seu local de implantação.

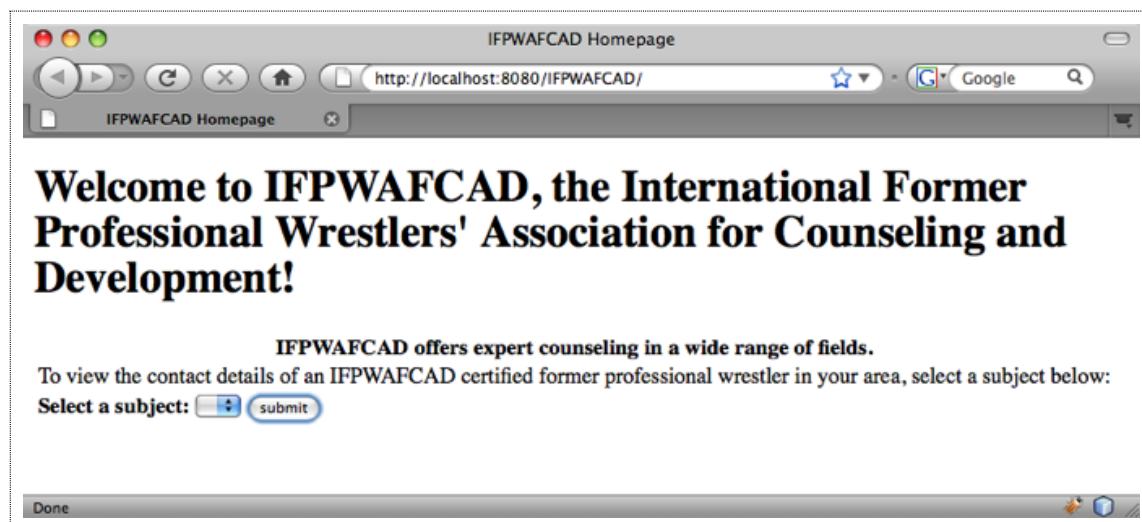


Figure 8. index.jsp exibido em um browser

Criando a página de resposta

Para preparar a interface de `response.jsp` é necessário primeiro criar o arquivo no projeto. Observe que a maioria do conteúdo exibido nesta página é gerado dinamicamente com a tecnologia JSP. Portanto, nas etapas a seguir você adicionará *espaços reservados* que posteriormente serão substituídos pelo código JSP.

1. Clique com o botão direito do mouse no nó do projeto IFPWAFCAD na janela Projetos e selecione Novo > JSP. A caixa de diálogo Novo Arquivo JSF será aberta.
2. No campo Nome do Arquivo JSP, digite `response`. Observe que a opção Páginas Web está atualmente selecionada no campo Localização, o que significa que o arquivo será criado no diretório `web` do projeto.

Trata-se do mesmo local em que se encontra a página de boas-vindas `index.jsp`.

3. Aceite as outras configurações default e clique em Finalizar. Um modelo, aberto no editor, será gerado para a nova página `response.jsp`. O novo nó JSP também será exibido abaixo de Páginas Web na janela Projetos.

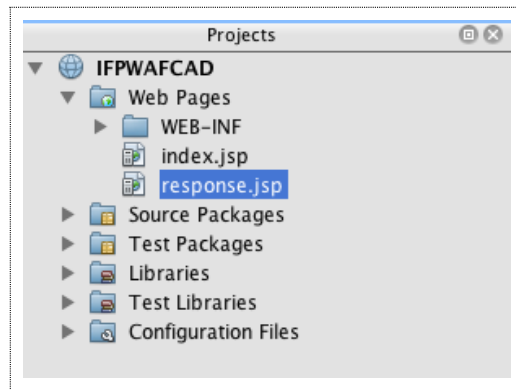


Figure 9. o nó `response.jsp` é exibido na janela Projetos

1. No editor, altere o título `IFPWAFCAD: {placeholder}`.
2. Remova a linha `<h1>Hello World!</h1>` alinhe entre as tags `<body>` e copie e cole a tabela HTML a seguir no corpo da página:

```
<table border="0">
  <thead>
    <tr>
      <th colspan="2">{placeholder}</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td><strong>Description: </strong></td>
      <td><span style="font-size:smaller; font-style:italic;">{placeholder}</span></td>
    </tr>
    <tr>
      <td><strong>Counselor: </strong></td>
      <td>{placeholder}
        <br>
        <span style="font-size:smaller; font-style:italic;">
          member since: {placeholder}</span>
        </td>
    </tr>
    <tr>
      <td><strong>Contact Details: </strong></td>
      <td><strong>email: </strong>
        <a href="mailto:{placeholder}">{placeholder}</a>
        <br><strong>phone: </strong>{placeholder}
      </td>
    </tr>
  </tbody>
</table>
```

Para exibir essa página em um browser, clique com o botão direito do mouse no editor e selecione Executar Arquivo (Shift-F6; Fn-Shift-F6 no Mac). A página será compilada, implantada no GlassFish Server e aberta no browser default.

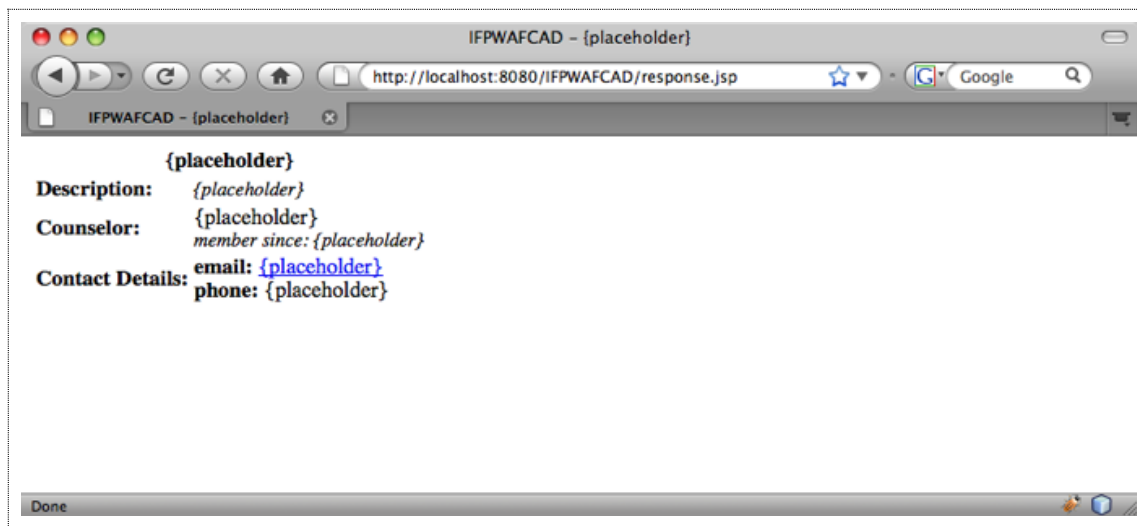



Figure 10. response.jsp exibido em um browser

Criando uma folha de estilo

Crie uma folha de estilo simples para melhorar a view da interface da Web. Este tutorial pressupõe que você sabe como funciona as regras de estilo e como elas afetam os elementos HTML correspondentes encontrados em `index.jsp` e `response.jsp`.

1. Abra o assistente de Novo Projeto pressionando o botão () Novo Arquivo na barra de ferramentas principal do IDE. Selecione a categoria Web, em seguida, Folha de Estilo em Cascata e clique em Próximo.
2. Digite `style` em Nome do Arquivo CSS e clique em Finalizar. O IDE cria um arquivo CSS vazio e o coloca no mesmo local do projeto onde estão `index.jsp` e `response.jsp`. Observe que agora será exibido um nó de `style.css` no projeto, na janela Projetos, e o arquivo será aberto no editor.
3. No editor, adicione o conteúdo a seguir ao arquivo `style.css`:

```

body {
    font-family: Verdana, Arial, sans-serif;
    font-size: smaller;
    padding: 50px;
    color: #555;
}

h1 {
    text-align: left;
    letter-spacing: 6px;
    font-size: 1.4em;
    color: #be7429;
    font-weight: normal;
    width: 450px;
}

table {
    width: 580px;
    padding: 10px;
    background-color: #c5e7e0;
}

th {
    text-align: left;
    border-bottom: 1px solid;
}

td {
    padding: 10px;
}

a:link {
    color: #be7429;
    font-weight: normal;
    text-decoration: none;
}

a:link:hover {
    color: #be7429;
    font-weight: normal;
    text-decoration: underline;
}

```

1. Vincule a folha de estilo a `index.jsp` e `response.jsp`. Nas duas páginas, adicione a linha a seguir entre as tags `<head>`:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Para navegar rapidamente entre arquivos que estão abertos no editor, pressione Ctrl-Tab e, em seguida, selecione o arquivo que deseja.

Preparando a Comunicação entre a Aplicação e o Banco de Dados

A maneira mais eficiente de implementar comunicação entre o servidor e o banco de dados é configurar um *pool de conexões* de banco de dados. A criação de uma nova conexão para cada solicitação do cliente pode ser muito demorada, especialmente em aplicações que recebem continuamente um grande número de solicitações. Para remediar isso, são criadas várias conexões que são mantidas em um pool de conexões. Todas as solicitações de entrada que requerem acesso à camada de dados da aplicação usam uma conexão já criada do pool. Igualmente, quando uma solicitação é concluída, a conexão é fechada, mas volta ao pool.

Depois de preparar a fonte de dados e o pool de conexões do servidor, é necessário instruir a aplicação a utilizar a fonte de dados. Isso normalmente é feito criando uma entrada no descritor de implantação `web.xml` da aplicação. Finalmente, você precisa garantir que o driver do banco de dados (Driver MySQL Connector/J JDBC) esteja acessível ao servidor.

*Importante: * Deste ponto em diante, você precisa assegurar-se de ter um banco de dados MySQL denominado `MyNewDatabase` configurado e com dados de amostra no [ifpwafcad.sql \(https://netbeans.org/projects/samples/downloads/download/Samples%252Fjava%2520Web%252Fifpwafcad.sql\)](https://netbeans.org/projects/samples/downloads/download/Samples%252Fjava%2520Web%252Fifpwafcad.sql). Este arquivo SQL cria duas tabelas, `Subject` e `Counselor`, a seguir, preenche-as com os dados de amostra. Se você ainda não tiver feito isso ou se precisar de ajuda, consulte [Conectando a um Banco de Dados MySQL \(../docs/ide/mysql.html\)](https://netbeans.org/docs/ide/mysql.html) antes de prosseguir com este tutorial.

Além disso, seu banco de dados precisa ser protegido por senha para criar uma fonte de dados e trabalhar com o GlassFish Server neste tutorial. Se você estiver utilizando a conta MySQL `root` sem senha, você poderá definir a senha usando um prompt de linha de comandos.

Este tutorial utiliza `nbuser` como senha de exemplo. Para definir sua senha como `nbuser`, navegue para o diretório `bin` de instalação do MySQL no prompt de linha de comandos e insira o seguinte:

```
shell> mysql -u root
mysql> UPDATE mysql.user SET Password = PASSWORD('_nbuser_')
-> WHERE User = 'root';
mysql> FLUSH PRIVILEGES;
```

Para mais informações, consulte o Manual de Referência MySQL oficial: [Proteção das Contas MySQL Iniciais \(http://dev.mysql.com/doc/refman/5.1/en/default-privileges.html\)](http://dev.mysql.com/doc/refman/5.1/en/default-privileges.html).

1. [Configurando uma fonte de dados JDBC e um pool de conexões](#)
2. [Fazendo referência à fonte de dados da aplicação](#)
3. [Adicionando o arquivo JAR do driver do banco de dados ao servidor](#)


Configurando uma fonte de dados JDBC e o pool de conexões

O GlassFish Server Open Source Edition contém bibliotecas de DBCP (Database Connection Pooling - Pool de Conexões do Banco de Dados) que oferecem a funcionalidade de pool de conexões de forma transparente a você como desenvolvedor. Para aproveitar esse recurso, é necessário configurar uma *fonte de dados JDBC* (<http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html>) (Java Database Connectivity) para o servidor que a sua aplicação pode utilizar para o pool de conexões.

Para obter mais informações sobre a tecnologia JDBC, consulte [Tutoriais Java: Conceitos Básicos do JDBC \(http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html\)](http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html).

É possível configurar a fonte de dados diretamente na Console de Administração do GlassFish Server ou, conforme descrito a seguir, você pode declarar os recursos de que a aplicação precisa em um arquivo `glassfish-resources.xml`. Quando a aplicação é implantada, o servidor lê as declarações de recursos e cria os recursos necessários.

As etapas a seguir demonstram como declarar um pool de conexões e uma fonte de dados com base no pool de conexões. O assistente de Recursos JDBC do NetBeans permite executar as duas ações.

1. Abra o assistente de Novo Projeto pressionando o botão () Novo Arquivo na barra de ferramentas principal do IDE. Selecione a categoria do GlassFish Server e, em seguida, Recurso JDBC e clique em Próximo.
2. Na Etapa 2, Atributos Gerais, selecione a opção Criar Novo Pool de Conexões JDBC e, em seguida, no campo de texto Nome JNDI, digite **jdbc/IFPWAFCAD**.

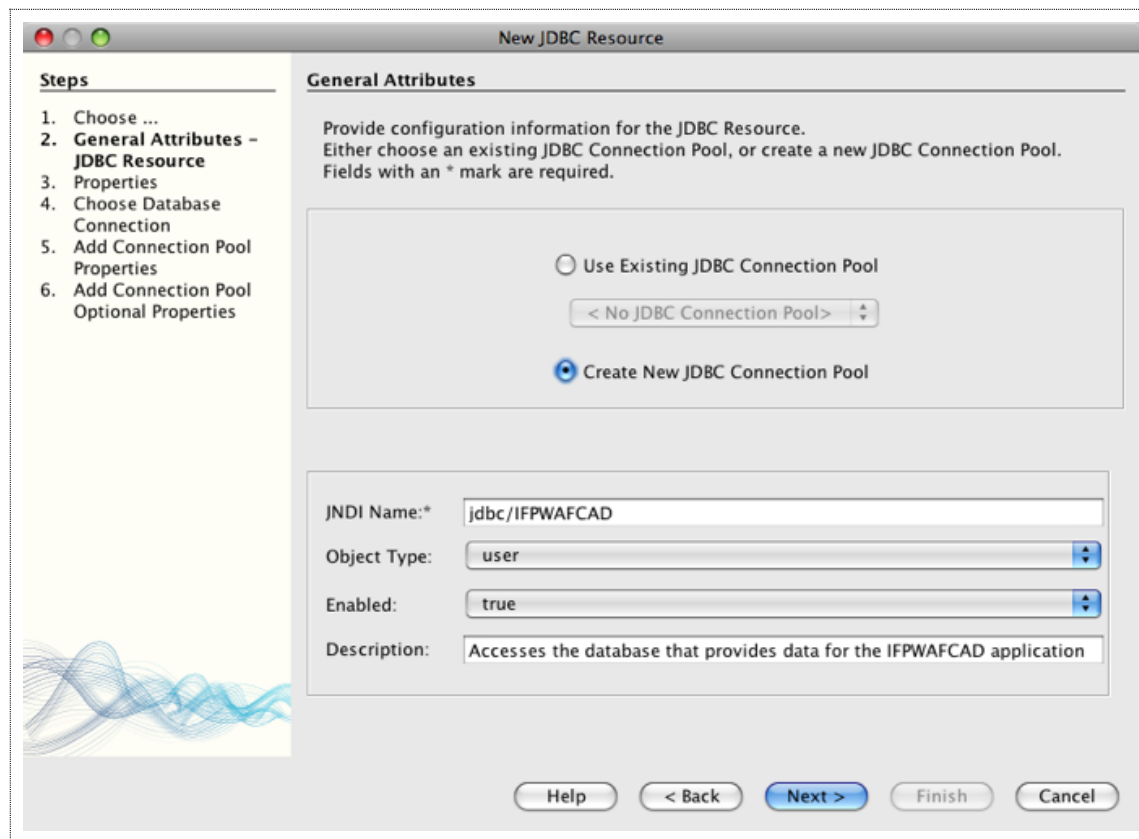


Figure 11. Especificar definições da fonte de dados no assistente de Recurso JDBC

A fonte de dados JDBC conta com a [JNDI \(http://www.oracle.com/technetwork/java/jndi/index.html\)](http://www.oracle.com/technetwork/java/jndi/index.html), a Interface de Nomenclatura e Diretório Java. A API JNDI fornece uma maneira uniforme de as aplicações encontrarem e acessarem as fontes de dados. Para obter mais informações, consulte [Tutorial do JNDI \(http://docs.oracle.com/javase/jndi/tutorial/\)](http://docs.oracle.com/javase/jndi/tutorial/).

1. Como opção, adicione uma descrição para a fonte de dados. Por exemplo, digite: `Accesses the database that provides data for the IFPWAFCAD application`.
2. Clique em **Próximo** e em **Próximo** novamente para ignorar a etapa 3, **Propriedades Adicionais**.
3. Na Etapa 4, digite **IfpwafcadPool** para Nome do Pool de Conexões JDBC. Certifique-se de que a opção **Extrair da Conexão Existente** esteja selecionada e escolha `jdbc:mysql://localhost:3306/MyNewDatabase` na lista drop-down. Clique em **Próximo**.

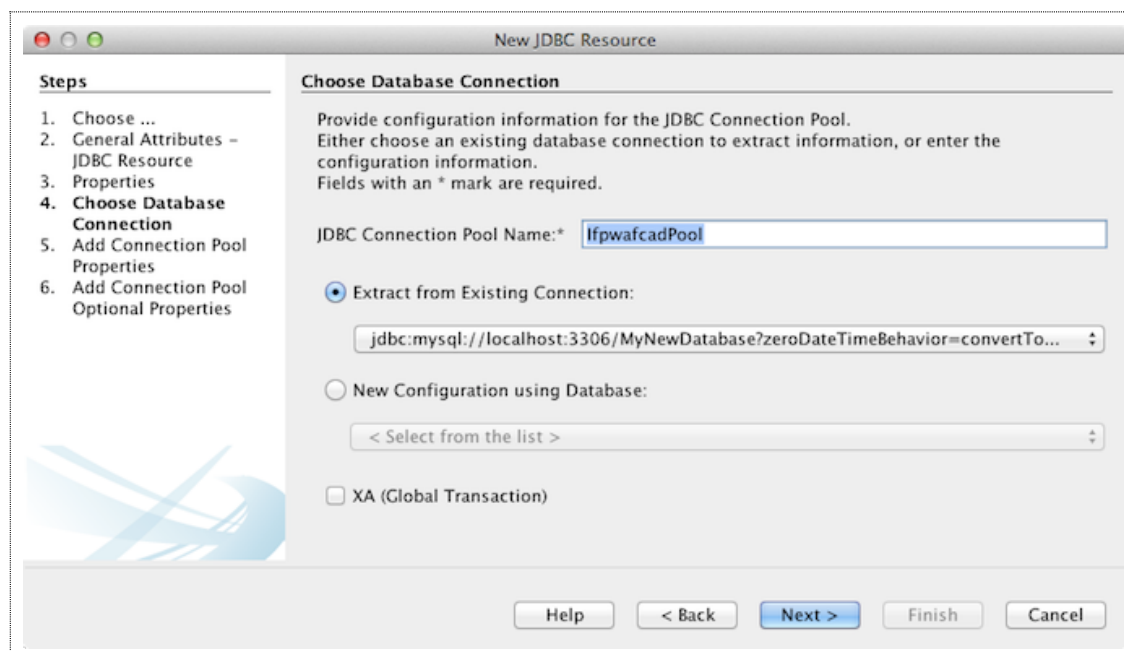


Figure 12. Especificar definições do pool de conexões no assistente de Recurso JDBC

*Observação: *O assistente detecta as conexões de banco de dados que tenham sido configuradas no IDE.

Portanto, neste ponto, é necessário que você já tenha criado uma conexão ao banco de dados `MyNewDatabase`. Você pode verificar as conexões que foram criadas ao abrir a janela **Serviços** (Ctrl-5; &8984-5 no Mac) e procurar nós de conexão (🔗) na categoria **Bancos de Dados**.

1. Na Etapa 5, selecione `javax.sql.ConnectionPoolDataSource` na lista drop-down **Tipo de Recurso**.

Observe que o IDE extrai informações da conexão de banco de dados especificada na etapa anterior e define as propriedades de nome e valor para o novo pool de conexões.

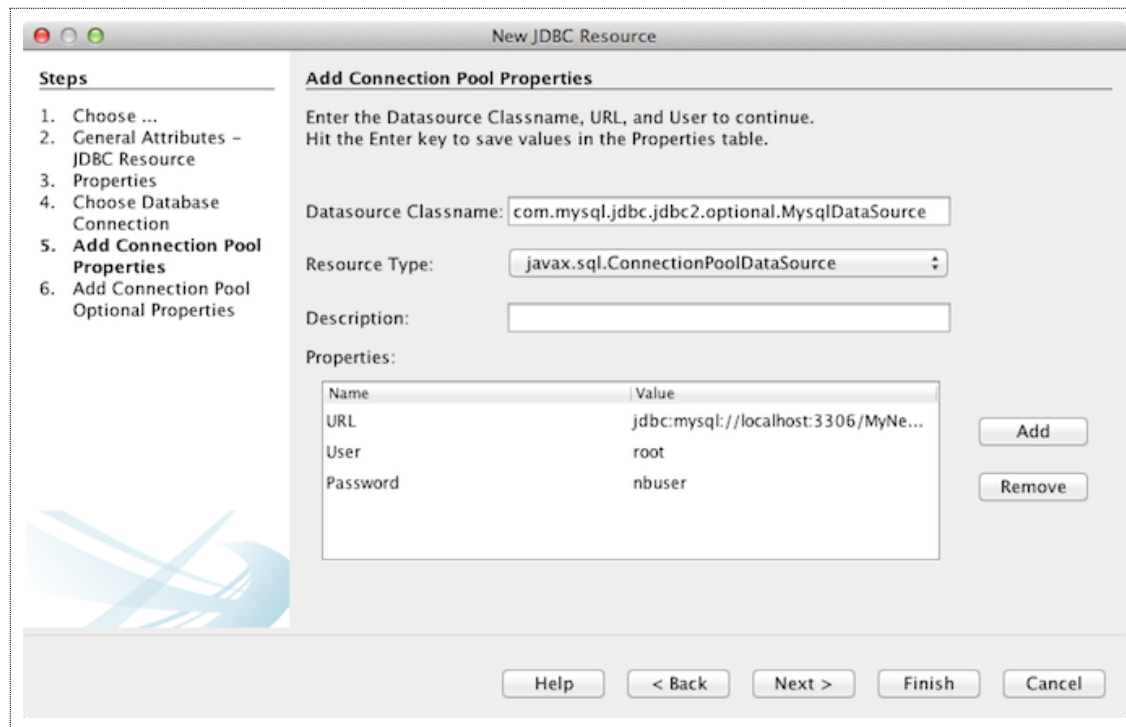


Figure 13. Os valores default são baseados nas informações extraídas da conexão de banco de dados selecionada

1. Clique em **Finalizar**. O assistente gera um arquivo `glassfish-resources.xml` que contém entradas para a fonte de dados e o pool de conexões especificados.

Na janela **Projetos**, você pode abrir o arquivo `glassfish-resources.xml` recém-criado em **Recursos do Servidor** e observar que, nas tags `<resources>`, foram declarados uma fonte de dados e um pool de conexões com os valores previamente especificados.

Para confirmar se uma nova fonte de dados e um pool de conexões estão realmente registrados no GlassFish Server, você pode implantar o projeto no servidor e, em seguida, localizar os recursos na janela **Serviços** do IDE:

1. Na janela **Projetos**, clique com o botão direito do mouse no nó do projeto **IFPWAFCAD** e selecione **Implantar**. O servidor será iniciado caso não esteja em execução e o projeto será compilado e implantado nele.
2. Abra a janela **Serviços** (Ctrl-5; &8984-5 no Mac) e expanda os nós **Servidores > GlassFish > Recursos > JDBC > Recursos JDBC e Pools de Conexões**. Observe que a nova fonte de dados e o pool de conexões agora são exibidos:

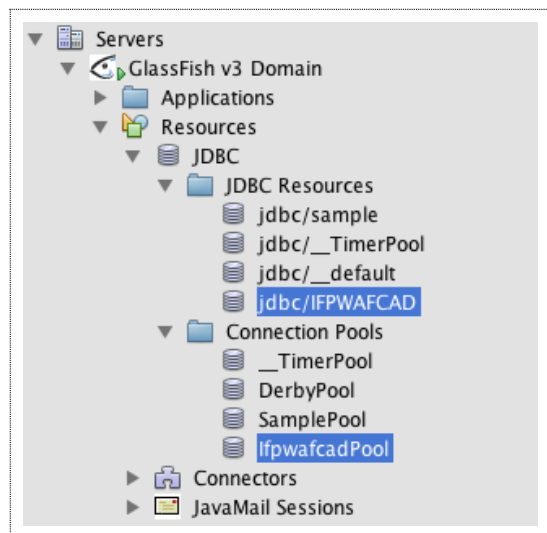


Figure 14. Nova fonte de dados e novo pool de conexões exibidos na janela Serviços

Fazendo referência à fonte de dados da aplicação

É necessário fazer referência ao recurso JDBC que acabou de configurar na aplicação Web. Para isso, você pode criar uma entrada no descritor de implantação `web.xml` da aplicação.

Os descritores de implantação são arquivos de texto com base em XML que contêm informações que descrevem como uma aplicação deve ser implantada em um ambiente específico. São usados normalmente, por exemplo, para especificar os parâmetros de contexto da aplicação, os padrões de comportamento e as configurações de segurança, bem como mapeamentos para servlets, filtros e listeners.

Observação. Se você tiver especificado o Java EE 6 ou Java EE 7 como a versão Java ao criar o projeto, será necessário criar o arquivo descritor de implantação selecionando Web > Descritor de Implantação Padrão, no assistente de Novo Arquivo.

Execute as etapas a seguir para fazer referência à fonte de dados no descritor de implantação da aplicação.

1. Na janela Projetos, expanda a pasta Arquivos de Configuração e clique duas vezes em `web.xml` para abrir o arquivo no editor.
2. Clique na guia Referências localizada na parte superior do editor.
3. Expanda o cabeçalho Referências de Recurso e clique em Adicionar para abrir a caixa de diálogo Adicionar Referência do Recurso.
4. Em Nome do Recurso, insira o nome do recurso que você forneceu ao configurar a fonte de dados no servidor acima (`jdbc/IFPWAFCAD`).
5. Digite `javax.sql.ConnectionPoolDataSource` no campo Tipo de Recurso. Clique em OK.

O campo Descrição é opcional, mas você pode inserir uma descrição do recurso que seja legível pelo usuário, por exemplo, Banco de Dados da aplicação IFPWAFCAD .

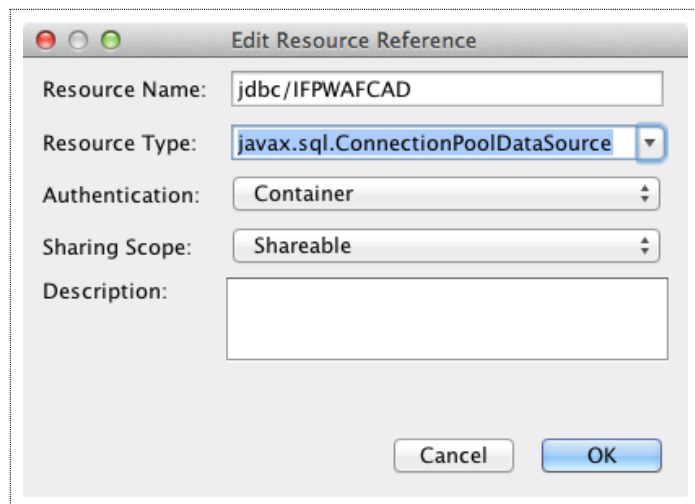


Figure 15. Especificar as propriedades do recurso na caixa de diálogo Adicionar Referência de Recurso
O novo recurso agora será listado abaixo do cabeçalho Referências de Recursos.

1. Para verificar se o recurso foi adicionado ao arquivo `web.xml`, clique na guia Código-fonte localizada na parte superior do editor. Observe que as tags `<resource-ref>` a seguir agora estão incluídas.

```
<resource-ref>
  <description>Database for IFPWAFCAD application</description>
  <res-ref-name>jdbc/IFPWAFCAD</res-ref-name>
  <res-type>javax.sql.ConnectionPoolDataSource</res-type>
  <res-auth>Container</res-auth>
  <res-sharing-scope>Shareable</res-sharing-scope>
</resource-ref>
```

Adicionando o arquivo JAR do driver do banco de dados ao servidor

A adição do arquivo JAR do driver do banco de dados é outra etapa vital para permitir que o servidor se comunique com o banco de dados. Normalmente, seria necessário localizar o diretório de instalação do driver do banco de dados e copiar o arquivo `mysql-connector-java-5.1.6-bin.jar` do diretório raiz do driver na pasta da biblioteca do servidor que estiver utilizando. Felizmente, o gerenciamento de servidor do IDE é capaz de detectar na implantação se o arquivo JAR foi adicionado e, caso não tenha sido, ele o adiciona automaticamente.

Para demonstrá-lo, abra o gerenciador Servidores (Selecione Ferramentas > Servidores). O IDE fornece uma opção de implantação do driver JDBC. Se a opção estiver ativada, ela iniciará uma verificação para determinar se serão necessários drivers para as aplicações implantadas do servidor. No caso do MySQL, se o driver for necessário e não for encontrado, o driver incorporado do IDE será implantado no local adequado do servidor.

1. Selecione Ferramentas > Servidores para abrir o gerenciador Servidores. Selecione o servidores do GlassFish Server no painel esquerdo.
2. No painel principal, selecione a opção Ativar Implantação do Driver JDBC.

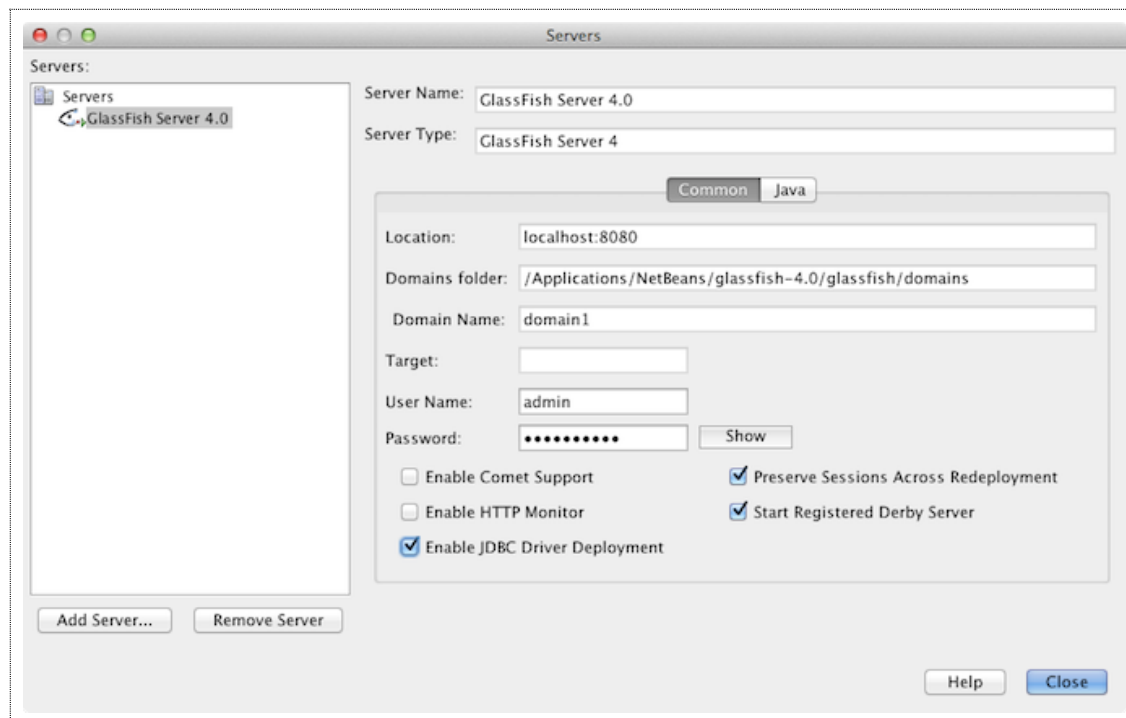


Figure 16. Opção Implantação de Driver JDBC ativa a implantação de driver automática

1. Antes de fechar o gerenciador de Servidores, anote o caminho indicado no campo de texto da pasta Domínios. Ao se conectar ao GlassFish Server no IDE, você está se conectando, na realidade, a uma *instância* do servidor de aplicações. Cada instância executa aplicações em um domínio exclusivo e o campo Nome do Domínio indica o nome do domínio que o servidor está utilizando. Conforme mostrado na imagem acima, o arquivo JAR do driver deverá estar localizado em `domain1`, que é o domínio default criado na instalação do GlassFish Server.
2. Clique em Fechar para sair do gerenciador Servidores.
3. No computador, vá até o diretório de instalação do GlassFish Server e entre na subpasta `domínios > domain1 > lib.` Como você já deve ter implantando o projeto IFPWAFCAD no servidor, você verá o arquivo ``mysql-connector-java-5.1.6-bin.jar``. Se você não vir o arquivo JAR do driver, execute a etapa a seguir.
4. Implante seu projeto no servidor. Na janela Projetos do IDE, selecione Implantar no menu de contexto do nó do projeto. Você pode ver o progresso na janela de Saída do IDE (Ctrl-4; ⌘-4 no Mac). A saída indica que o driver MySQL está implantado no GlassFish Server.

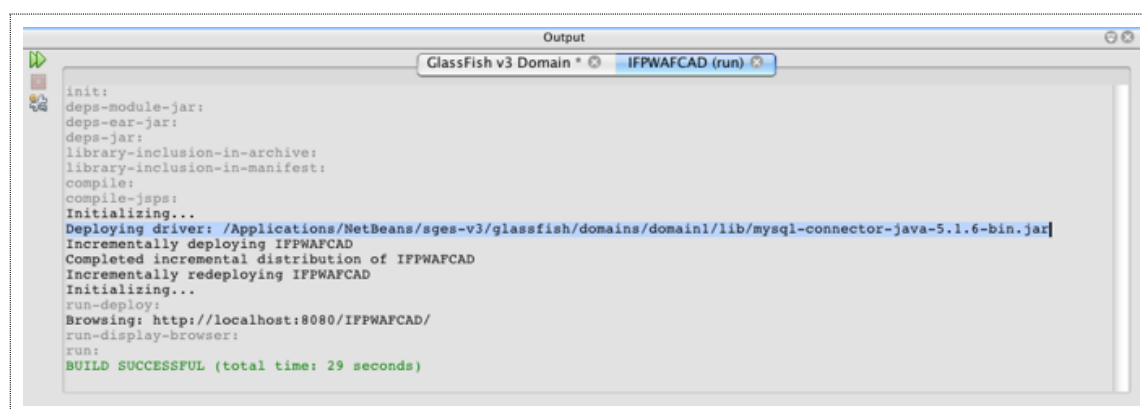


Figure 17. A janela de Saída indica que o driver MySQL foi implantado

Agora, se você voltar à subpasta `domain1/lib` no computador, você verá que o arquivo `mysql-connector-java-5.1.6.x-bin.jar` foi adicionado automaticamente.

Adicionando Lógica Dinâmica

Retornando aos espaços reservados `index.jsp` e `response.jsp` que você criou anteriormente no tutorial, você

pode implementar o código JSTL que permite às páginas gerar conteúdo *dinamicamente*, ou seja, com base na entrada do usuário. Para isso, execute as três tarefas a seguir.

1. [Adicione a biblioteca JSTL ao classpath do projeto](#)
2. [Implemente código JSTL](#)

Adicionando a biblioteca JSTL ao classpath do projeto

É possível aplicar a [JavaServer Pages Standard Tag Library \(http://www.oracle.com/technetwork/java/index-jsp-135995.html\)](http://www.oracle.com/technetwork/java/index-jsp-135995.html) (JSTL) para acessar e exibir dados obtidos do banco de dados. O GlassFish Server inclui a biblioteca JSTL por default. Você pode verificar isso expandindo o nó do GlassFish Server no nó de Bibliotecas, na janela Projetos e procurando a biblioteca `javax.servlet.jsp.jstl.jar`. (As versões mais antigas do GlassFish server usam a biblioteca `jstl-impl.jar`.) Como as bibliotecas do GlassFish Server são, por default, adicionadas ao classpath do projeto, não é necessário executar nenhuma etapa dessa tarefa.

A JSTL fornece as seguintes quatro áreas básicas de funcionalidade.

- `core` : tarefas comuns e estruturais, como iteradores e condicionais para manipular controle de fluxo
- `fmt` : formatação da mensagem de internacionalização e localização
- `sql` : acesso a banco de dados simples
- `xml` : manipulação de conteúdo XML

Este tutorial foca no uso das bibliotecas de tags `core` e `sql`.

Implementando o código JSTL

Agora você pode implementar o código que recupera, de maneira dinâmica, os dados, exibindo-os para cada página. As duas páginas requerem que você implemente uma consulta SQL que utilize a fonte de dados criada anteriormente no tutorial.

O IDE fornece vários snippets de JSTL específicos do banco de dados que podem ser selecionados na Paleta (Ctrl-Shift-8; ⌘-Shift-8 no Mac).

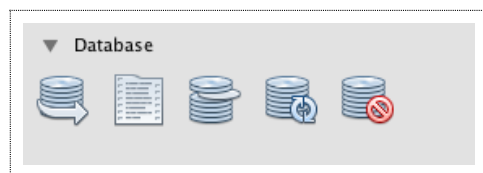


Figure 18. Selecione os snippets JSTL específicos do banco de dados na Paleta

index.jsp

Para exibir dinamicamente o conteúdo do form no `index.jsp`, você precisa acessar os ``name`s` da tabela de banco de dados Subject.

1. Coloque o mouse sobre o item Relatório do BD na Paleta.

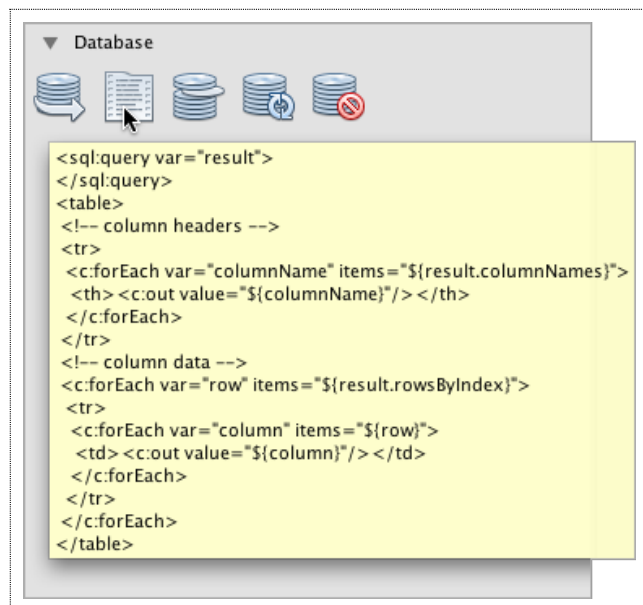


Figure 19. Digite 'db' e pressione Ctrl-Espaço para acessar os snippets JSTL específicos do banco de dados. O item Relatório do BD utiliza a tag `<sql:query>` para criar uma consulta SQL e, em seguida, utiliza a tag `<c:forEach>` para passar pelo `resultset` da consulta e obter os dados recuperados.

1. Coloque o cursor acima da declaração `<%@page ... %>` (linha 7) e, em seguida, clique duas vezes no item Relatório do BD na Paleta. Na caixa de diálogo que é exibida, insira os seguintes detalhes:

- **Nome da Variável:** subjects
- **Escopo:** page
- **Fonte de Dados:** jdbc/IFPWAFCAD
- **Instrução da Consulta:** SELECT subject_id, name FROM Subject

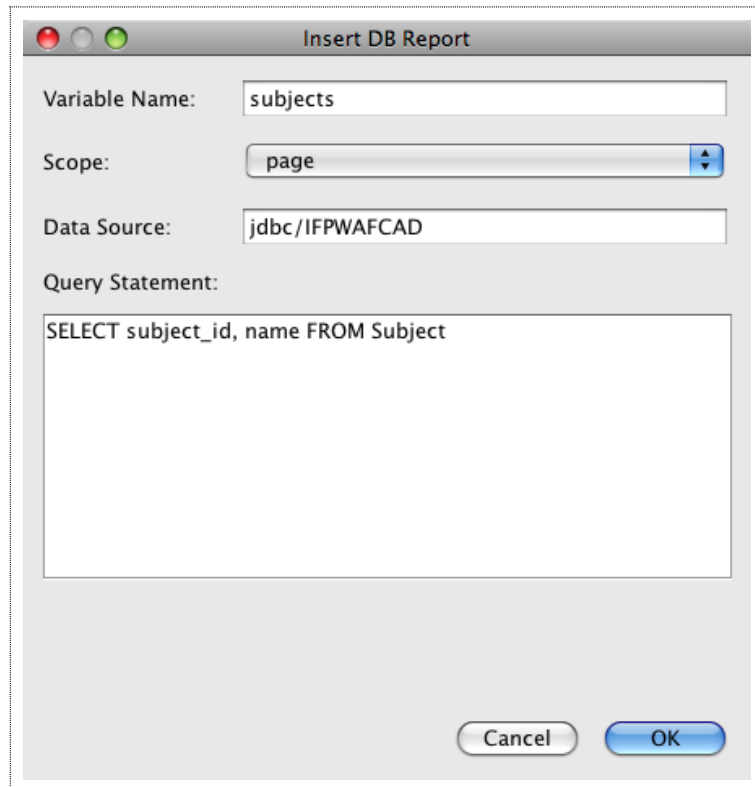


Figure 20. Use a caixa de diálogo Inserir Relatório do BD para especificar detalhes específicos da consulta. 1. Clique em OK. O conteúdo a seguir é gerado no arquivo `index.jsp`. (Novo conteúdo mostrado em **negrito**.)

```

*<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>*
<%--
    Document    : index
    Author      : nbuser
--%>

*<sql:query var="subjects" dataSource="jdbc/IFPWAFCAD">
    SELECT subject_id, name FROM Subject
</sql:query>

<table border="1">
    <!-- column headers -->
    <tr>
        <c:forEach var="columnName" items="${subjects.columnNames}">
            <th><c:out value="${columnName}"/></th>
        </c:forEach>
    </tr>
    <!-- column data -->
    <c:forEach var="row" items="${subjects.rowsByIndex}">
        <tr>
            <c:forEach var="column" items="${row}">
                <td><c:out value="${column}"/></td>
            </c:forEach>
        </tr>
    </c:forEach>
</table>*

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

```

Observe que o IDE adicionou automaticamente as diretivas `taglib` necessárias para as tags JSTL utilizadas no conteúdo gerado (`<sql:query>` e `<c:forEach>`). Uma diretiva `taglib` declara que a página do JSP utiliza tags personalizadas (por exemplo, JSTL), nomeia a biblioteca de tags que as define e especifica seu prefixo de tag.

1. Execute o projeto para ver como ele é exibido no browser. Clique com o botão direito do mouse no nó na janela Projetos e selecione Executar.

Quando você seleciona Executar, o IDE implanta o projeto no GlassFish Server, a página de índice é compilada em um servlet e a página de boas-vindas é aberta no browser default. O código gerado do item Relatório de BD cria a tabela a seguir na página de boas-vindas.

subject_id	name
1	Financial Consultancy
2	Existential Psychotherapy
3	Temper Management
4	Past-Life Regression
5	Marriage Guidance
6	Crisis Management
7	Dream Analysis
8	Hypnosis
9	Reiki

Figure 21. Utilize o Relatório de BD para criar protótipos dos dados da tabela do banco de dados. Como podemos ver, o item Relatório de BD permite testar rapidamente a conexão do banco de dados e permite, também, exibir os dados da tabela do banco de dados no browser. Isso pode ser particularmente útil ao se fazer protótipos.

As etapas a seguir demonstram como integrar o código gerado à lista drop-down HTML criada anteriormente no tutorial.

1. Examine os dados da coluna no código gerado. Duas tags `<c:forEach>` são utilizadas, uma está aninhada dentro da outra. Isso faz com que o contêiner JSP (por exemplo, o GlassFish Server) execute um loop em todas as linhas da tabela e, para cada linha, ele executa um loop em todas as colunas. Dessa maneira, os dados da tabela inteira são exibidos.
2. Integre as tags `<c:forEach>` no form HTML, conforme a seguir. O valor de cada item se torna `subject_id` e o texto de saída se torna o `name`, conforme registrado no banco de dados. (As alterações são exibidas em **negrito**).

```
<form action="response.jsp">
  <strong>Select a subject:</strong>
  <select name="subject_id">
    *<c:forEach var="row" items="${subjects.rowsByIndex}">
      <c:forEach var="column" items="${row}">*
        <option *value="<c:out value="${column}"/>"*><c:out value="${column}"/>*</option>
      *</c:forEach>
    </c:forEach>*
  </select>
  <input type="submit" value="submit" name="submit" />
</form>
```

Uma maneira alternativa e mais simples de integrar as tags `<c:forEach>` ao form HTML seria a seguinte:

```
<form action="response.jsp">
  <strong>Select a subject:</strong>
  <select name="subject_id">
    *<c:forEach var="row" items="${subjects.rows}">*
      <option *value="${row.subject_id}"*>${row.name}</option>
    *</c:forEach>*
  </select>
  <input type="submit" value="submit" name="submit" />
</form>
```

Em qualquer um dos casos, as tags `<c:forEach>` executam loop em todos os valores de `subject_id` e `name` da consulta SQL e inserem cada par nas tags `<option>` HTML. Dessa maneira, a lista drop-down do form é preenchida com dados.

1. Delete a tabela que foi gerada a partir do item Relatório de BD. (A exclusão é mostrada a seguir como **texto tachado**.)

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
<!--
  Document      : index
  Created on    : Dec 22, 2009, 7:39:49 PM
  Author       : nbuser
--%>

<sql:query var="subjects" dataSource="jdbc/IFPWAFCAD">
  SELECT subject_id, name FROM Subject
</sql:query>

*[.line-through]#<table border="1">
  <!-- column headers -->
  <tr>
    <c:forEach var="columnName" items="${subjects.columnNames}">
      <th><c:out value="${columnName}" /></th>
    </c:forEach>
  </tr>
  <!-- column data -->
  <c:forEach var="row" items="${subjects.rowsByIndex}">
    <tr>
      <c:forEach var="column" items="${row}">
        <td><c:out value="${column}" /></td>
      </c:forEach>
    </tr>
  </c:forEach>
</table>#*

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

1. Salve as alterações (Ctrl-S; ⌘-S no Mac).
2. Atualize a página de boas-vindas do projeto no browser.

Note que a lista drop-down do browser contém, agora, nomes de assuntos que foram recuperados do banco de dados.

Você não precisa reimplantar seu projeto porque a opção de compilação ao salvar está ativada para seu projeto, por default. Isso significa que, quando você modifica e salva um arquivo, ele é automaticamente compilado e implantado, e não é necessário recompilar o projeto inteiro. É possível ativar e desativar compilar ao salvar para seu projeto na categoria Compilar da janela Propriedades do projeto.

response.jsp

A página de resposta fornece detalhes para o conselheiro que corresponde ao assunto escolhido na página de boas-vindas. A consulta criada deverá selecionar o registro do conselheiro cujo `counselor_id` corresponda ao `counselor_idfk` do registro do assunto selecionado.

1. Coloque seu cursor acima da declaração `<%@page ... %>` (linha 7) e clique duas vezes na Consulta do DB na Paleta para abrir a caixa de diálogo Inserir Consulta do BD.
2. Informe os seguintes detalhes na caixa de diálogo Inserir Consulta DB.

- **Nome da Variável:** `counselorQuery`
- **Escopo:** `page`
- **Fonte de Dados:** `jdbc/IFPWAFCAD`
- **Instrução de Consulta:** `SELECT * FROM Subject, Counselor WHERE Counselor.counselor_id = Subject.counselor_idfk AND Subject.subject_id = ? <sql:param value="${param.subject_id}"/>`

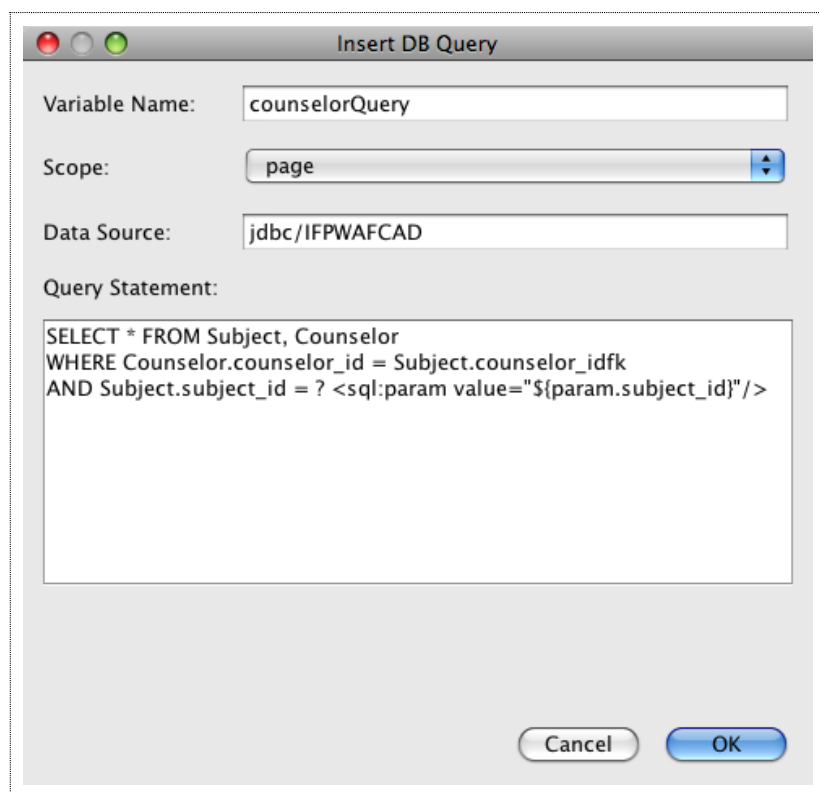


Figure 22. Use a caixa de diálogo Inserir Consulta do BD para especificar detalhes específicos da consulta

1. Clique em OK. O conteúdo a seguir é gerado no arquivo `response.jsp`. (Novo conteúdo mostrado em **negrito**.)

```
*<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>*
<%-
    Document    : response
    Created on  : Dec 22, 2009, 8:52:57 PM
    Author      : nbuser
--%>

*<sql:query var="counselorQuery" dataSource="jdbc/IFPWAFCAD">
    SELECT * FROM Subject, Counselor
    WHERE Counselor.counselor_id = Subject.counselor_idfk
    AND Subject.subject_id = ? <sql:param value="${param.subject_id}"/>
</sql:query>*

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
```

Observe que o IDE adicionou automaticamente a diretiva `taglib` necessária para a tag `<sql:query>`. Além disso, observe que você utilizou uma tag `<sql:param>` diretamente na consulta. Como essa consulta conta com o valor `subject_id` que foi submetido de `index.jsp`, é possível extrair o valor utilizando uma instrução EL (Linguagem de Expressão) na forma de `${param.subject_id}` e, em seguida, passá-lo para a tag `<sql:param>`, para que ele possa ser utilizado no local do ponto de interrogação da SQL (?) durante o runtime.

1. Utilize uma tag `<c:set>` para definir uma variável que corresponda ao primeiro registro (ou seja, linha) do `resultset` retornado da consulta. (Novo conteúdo mostrado em **negrito**.)

```
<sql:query var="counselorQuery" dataSource="jdbc/IFPWAFCAD">
  SELECT * FROM Subject, Counselor
  WHERE Counselor.counselor_id = Subject.counselor_idfk
  AND Subject.subject_id = ? <sql:param value="${param.subject_id}"/>
</sql:query>

*<c:set var="counselorDetails" value="${counselorQuery.rows[0]}" />*
```

Embora o `resultset` retornado da consulta deva conter apenas um único registro, está é uma etapa necessária, pois a página precisa acessar valores do registro utilizando instruções EL (Linguagem de Expressão). Lembre-se que no `index.jsp` você podia acessar valores do `resultset` simplesmente utilizando uma tag `<c:forEach>`. No entanto, a tag `<c:forEach>` opera definindo uma variável para linhas contidas na consulta, permitindo, assim, extrair valores incluindo a variável da linha nas instruções EL.

1. Adicione a diretiva `taglib` para a biblioteca `core` do JSTL na parte superior do arquivo, para que a tag `<c:set>` seja entendida. (Novo conteúdo mostrado em **negrito**.)

```
*<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>*
<%@taglib prefix="sql" uri="http://java.sun.com/jsp/jstl/sql"%>
```

1. Na marcação HTML, substitua todos os espaços reservados por código de instruções EL que exibem os dados contidos na variável `counselorDetails`. (Alterações mostradas a seguir em **negrito**):

```


<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>*${counselorDetails.name}*</title>
  </head>

  <body>
    <table>
      <tr>
        <th colspan="2">*${counselorDetails.name}*</th>
      </tr>
      <tr>
        <td><strong>Description: </strong></td>
        <td><span style="font-size:smaller; font-
style:italic;">*${counselorDetails.description}*</span></td>
      </tr>
      <tr>
        <td><strong>Counselor: </strong></td>
        <td><strong>*${counselorDetails.first_name} ${counselorDetails.nick_name}
${counselorDetails.last_name}*</strong>
        <br><span style="font-size:smaller; font-style:italic;">
        <em>member since: *${counselorDetails.member_since}*</em></span></td>
      </tr>
      <tr>
        <td><strong>Contact Details: </strong></td>
        <td><strong>email: </strong>
        <a href="mailto:*${counselorDetails.email}*">*${counselorDetails.email}*</a>
        <br><strong>phone: </strong>*${counselorDetails.telephone}*</td>
      </tr>
    </table>
  </body>
</html>

```

Executando a Aplicação Pronta

Você, agora, concluiu a aplicação. Tente executá-lo novamente para ver como ele é exibido no browser. Observe que, graças à funcionalidade Compilar ao Salvar do NetBeans, você não precisa se preocupar com a compilação ou reimplantação da aplicação. Assim, ao executar um projeto, você terá certeza de que a implantação contém as alterações mais recentes.

Clique no botão () Executar Projeto na barra de ferramentas principal. A página `index.jsp` será aberta no browser default do IDE.

Quando `index.jsp` for exibido no browser, selecione um assunto na lista drop-down e clique em `submit`. Você será direcionado à página `response.jsp`, mostrando os detalhes correspondentes à sua seleção.

Past-Life Regression	
Description:	<i>Past-Life Regression is a journey of the soul, backward and forward through time, like a yo-yo.</i>
Counselor:	Andre "The Giant" <i>member since: 2001-01-12</i>
Contact Details:	email: bobo@ifpwafcad.com phone: 606 443-4567

Figure 23. `response.jsp` exibido em um browser, mostrando os dados recuperados do banco de dados. Isso conclui o tutorial Criando uma Aplicação Web Simples Usando um Banco de Dados MySQL. Este documento demonstra como criar uma aplicação Web simples que se conecta a um banco de dados MySQL. Este tutorial

também demonstrou como construir uma aplicação utilizando uma arquitetura básica de duas camadas e utilizou várias tecnologias, incluindo JSP, JSTL, JDBC e JNDI, como uma forma de acessar e exibir dinamicamente os dados.



Solução de Problemas

A maioria dos problemas que ocorrem com a aplicação do tutorial são devidos a dificuldades de comunicação entre o GlassFish Server Open Source Edition e o servidor de banco de dados MySQL. Se a sua aplicação não for exibida corretamente, ou se você estiver recebendo um erro de servidor, os testes a seguir poderão ser úteis.

- [Existem recursos do banco de dados?](#)
- [O pool de conexões e a fonte de dados existem no servidor?](#)
- [O driver MySQL Connector/J está acessível ao GlassFish Server?](#)
- [O banco de dados é protegido por senha?](#)
- [As propriedades do pool de conexões estão definidas corretamente?](#)

Existem recursos do banco de dados?

Utilize a janela Serviços do IDE (Ctrl-5; ⌘-5 no Mac) para garantir que o servidor MySQL esteja em execução e que o MyNewDatabase esteja acessível e contenha os dados da tabela adequados.

- Para conectar-se ao servidor de banco de dados MySQL, clique com o botão direito do mouse no nó Servidor MySQL e selecione Conectar.
- Se um nó de conexão () para o MyNewDatabase não for exibido na janela Serviços, você poderá criar uma conexão clicando com o botão direito do mouse no nó do driver MySQL () e selecionando a opção Conectar Usando. Insira os detalhes necessários na caixa de diálogo exibida.

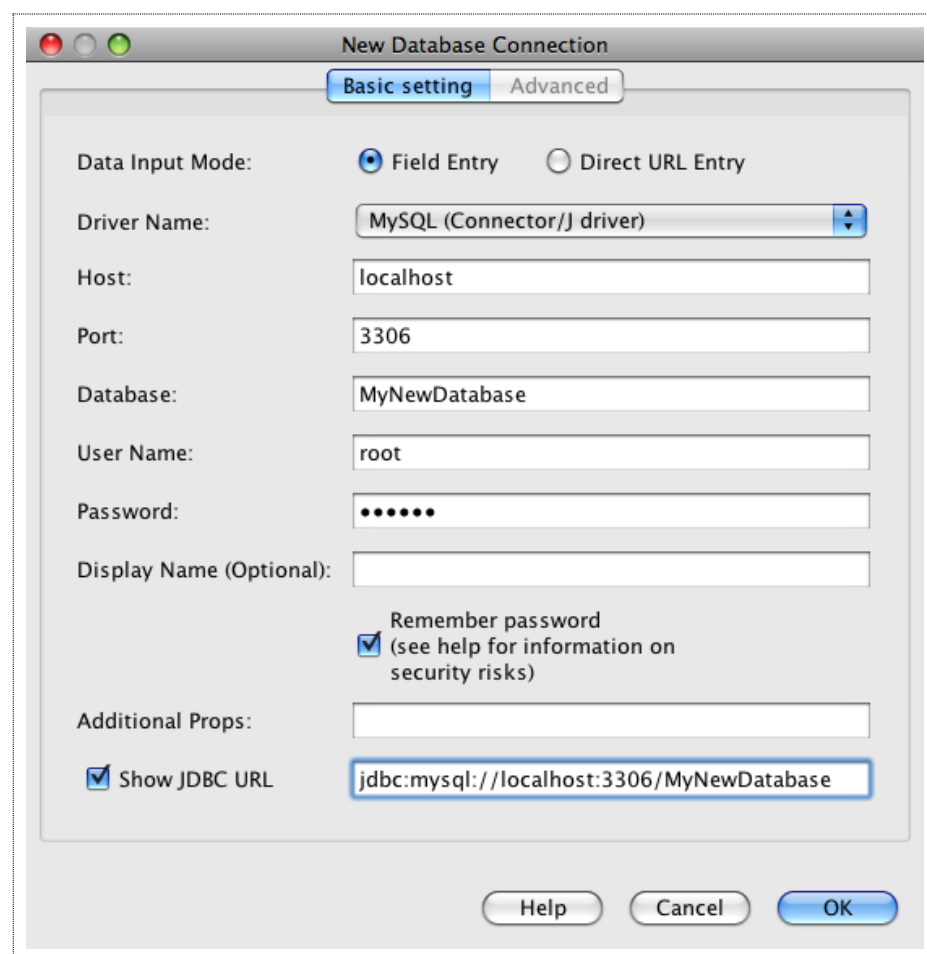




Figure 24. Estabeleça uma conexão de banco de dados no IDE utilizando a caixa de diálogo Nova Conexão de Banco de Dados

Os campos fornecidos na caixa de diálogo Nova Conexão de Banco de Dados espelham a string de URL inserida

Criando uma aplicação Web Simples Usando um Banco de Dad... https://netbeans.apache.org/kb/docs/web/mysql-webapp_pt_BR...

na opção Mostrar URL do JDBC. Portanto, se você souber o URL (por exemplo, `jdbc:mysql://localhost:3306/MyNewDatabase`) você poderá colá-lo no campo Mostrar URL do JDBC e os campos remanescentes da caixa de diálogo serão automaticamente preenchidos. * Para garantir que as tabelas `Subject` e `Counselor` existam e que contenham dados de amostra, expanda o nó de conexão `MyNewDatabase` () e localize o nó do catálogo `MyNewDatabase` (). Expanda o nó de catálogo para exibir as tabelas existentes. Você pode exibir os dados da tabela clicando com o botão direito do mouse em um nó da tabela e selecionando Exibir Dados.

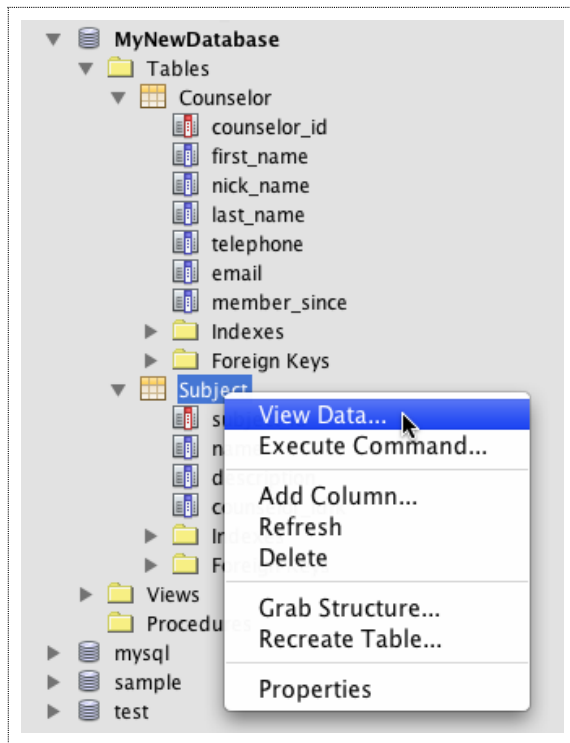


Figure 25. Exibir dados da tabela selecionando a opção Exibir Dados no menu contextual de um nó da tabela de banco de dados

O pool de conexões e a fonte de dados existem no servidor?

Após a implantação da aplicação no GlassFish Server, `sun-resources.xml` contido no projeto deverá instruir o servidor a criar um recurso e um pool de conexões JDBC. É possível determinar se eles existem no nó Servidores na janela Serviços.

- Expanda o nó Servidores > GlassFish Server > Recursos. Expanda Recursos JDBC para exibir a fonte de dados `jdbc/IFPWAFCAD` criada de `glassfish-resources.xml`. Expanda o nó Pools de Conexões para exibir o pool de conexões `IfpwafcadPool` criado a partir de `glassfish-resources.xml`. (Isso está [demonstrado acima](#).)

O driver MySQL Connector/J está acessível para o GlassFish Server?

Certifique-se de que o driver `MySQLConnector/J` foi implantado no GlassFish Server. (Isso é discutido em [Adicionando o arquivo JAR do driver do banco de dados ao servidor](#).)

- Localize a pasta de instalação do GlassFish Server no computador e faça uma análise detalhada da subpasta `GlassFish domains/domain1/lib`. Aqui você encontrará o arquivo `mysql-connector-java-5.1.6-bin.jar`.

O banco de dados é protegido por senha?

O banco de dados precisa ser protegido por senha para que a fonte de dados do GlassFish Server funcione adequadamente neste tutorial. Se você estiver utilizando a conta MySQL `root` sem senha, você poderá definir a senha usando um prompt de linha de comandos.

- Para definir sua senha como `nbuser`, navegue para o diretório `bin` de instalação do MySQL no prompt de linha de comandos e insira o seguinte:

```

shell> mysql -u root
mysql> UPDATE mysql.user SET Password = PASSWORD('_nuser_')
-> WHERE User = 'root';
mysql> FLUSH PRIVILEGES;

```

Para mais informações, consulte o Manual de Referência MySQL oficial: [Proteção das Contas MySQL Iniciais \(http://dev.mysql.com/doc/refman/5.1/en/default-privileges.html\)](http://dev.mysql.com/doc/refman/5.1/en/default-privileges.html).

As propriedades do pool de conexões estão definidas corretamente?

Certifique-se de que o pool de conexões esteja funcionando corretamente para o servidor.

1. Abra a janela Serviços (Ctrl-5; ⌘-5 no Mac) e expanda o nó Servidores.
2. Clique com o botão direito do mouse no nó do GlassFish Server e selecione Exibir Console de Administração
3. Informe o nome do usuário e a senha se for solicitado. Você pode exibir o nome de usuário e senha no gerenciador de Servidores.
4. Na árvore à esquerda da console, expanda o nó Recursos > JDBC > Pools de Conexões de JDBC > IfpwafcadPool . OS detalhes do pool de conexões IfpwafcadPool são exibidos na janela principal.
5. Clique no botão Ping. Se o pool de conexões estiver configurado adequadamente, você verá uma mensagem “Ping Bem-sucedido”.



Figure 26. Teste o seu pool de conexões clicando em Ping na Console de Administração do GlassFish Server

1. Se o ping falhar, clique na guia Propriedades Adicionais e verifique se os valores da propriedade listados estão corretamente definidos.

[Envie-nos Seu Feedback \(/about/contact_form.html?to=3&subject=Feedback:%20Creating%20Web%20App%20MySQL\)](#)

Consulte Também

Para obter mais informações sobre o desenvolvimento Java Web, consulte os recursos a seguir.

- **Artigos e Tutoriais do NetBeans**
- [Conectando-se a um Banco de Dados MySQL no NetBeans IDE \(./ide/mysql.html\)](#) Aborda os conceitos básicos sobre como trabalhar com um banco de dados MySQL no IDE.
- [Introdução ao JavaServer Faces 2.x \(jsf20-intro.html\)](#). Um tutorial introdutório descrevendo como utilizar o framework JSF em um projeto Java Web.
- [Introdução à Framework do Spring \(quickstart-webapps-spring.html\)](#). Um tutorial introdutório que descreve como criar uma aplicação Web MVC utilizando o Framework do Spring.
- **Java Database Connectivity (JDBC)**

- [Visão Geral do JDBC \(http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html\)](http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html)
- [Introdução à API JDBC \(http://download.oracle.com/javase/6/docs/technotes/guides/jdbc/getstart/GettingStartedTOC.fm.html\)](http://download.oracle.com/javase/6/docs/technotes/guides/jdbc/getstart/GettingStartedTOC.fm.html)
- [Tutoriais do Java: Conceitos Básicos do JDBC \(http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html\)](http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html)
- **JavaServer Pages Standard Tag Library (JSTL)**
- [JavaServer Pages Standard Tag Library \(http://www.oracle.com/technetwork/java/index-jsp-135995.html\)](http://www.oracle.com/technetwork/java/index-jsp-135995.html)
(página oficial do produto)
- **Java Naming and Directory Interface (JNDI)**
- [Tecnologias Java SE Core - Java Naming and Directory Interface \(http://www.oracle.com/technetwork/java/jndi-136720.html\)](http://www.oracle.com/technetwork/java/jndi-136720.html)
- [Tutorial do JNDI \(http://docs.oracle.com/javase/jndi/tutorial/\)](http://docs.oracle.com/javase/jndi/tutorial/)
- [Tutoriais do Java: Java Naming and Directory Interface \(http://docs.oracle.com/javase/tutorial/jndi/index.html\)](http://docs.oracle.com/javase/tutorial/jndi/index.html)

[f \(https://www.facebook.com/NetBeans\)](https://www.facebook.com/NetBeans) [t \(https://twitter.com/netbeans\)](https://twitter.com/netbeans)

[G \(https://github.com/apache/netbeans\)](https://github.com/apache/netbeans) [You Tube \(https://www.youtube.com/user/netbeansvideos\)](https://www.youtube.com/user/netbeansvideos)

[🔗 \(https://tinyurl.com/netbeans-slack-signup/\)](https://tinyurl.com/netbeans-slack-signup/)

[🔗 \(https://issues.apache.org/jira/projects/NETBEANS/summary\)](https://issues.apache.org/jira/projects/NETBEANS/summary)

[🔗 See this page in GitHub. \(https://github.com/apache/netbeans-website/blob/master/netbeans.apache.org/src/content/kb/docs/web/mysql-webapp_pt_BR.asciidoc\)](https://github.com/apache/netbeans-website/blob/master/netbeans.apache.org/src/content/kb/docs/web/mysql-webapp_pt_BR.asciidoc)



[\(http://www.apache.org/\)](http://www.apache.org/)



[September 21-23 2021 \(https://www.apache.org/events/current-event.html\)](https://www.apache.org/events/current-event.html)

[ABOUT](#) [\(/ABOUT](#) [/INDEX.HTML\)](#)

[Who's Who](#)

[\(/https://netbeans.apache.org/community/who.html\)](https://netbeans.apache.org/community/who.html)

[Thanks](#)

[\(/https://www.apache.org/foundation/thanks.html\)](https://www.apache.org/foundation/thanks.html)

[Sponsorship](#)

[\(/https://www.apache.org/foundation/sponsorship.html\)](https://www.apache.org/foundation/sponsorship.html)

[Security](#)

[\(/https://www.apache.org/events/current-event.html\)](https://www.apache.org/events/current-event.html)

[COMMUNITY](#) [\(/COMMUNITY](#) [/INDEX.HTML\)](#)

[Mailing lists](#)

[\(/https://netbeans.apache.org/community/ mailing-lists.html\)](https://netbeans.apache.org/community/ mailing-lists.html)

[Becoming a committer](#)

[\(/https://netbeans.apache.org/community/ committer.html\)](https://netbeans.apache.org/community/ committer.html)

[NetBeans Events](#)

[\(/https://netbeans.apache.org/ community/ events.html\)](https://netbeans.apache.org/ community/ events.html)

[Apache Events](#)

[\(/https://www.apache.org/ events/current-event.html\)](https://www.apache.org/ events/current-event.html)

[PARTICIPATE](#) [\(/PARTICIPATE](#) [/INDEX.HTML\)](#)

[Submitting Pull Requests](#)

[\(/https://netbeans.apache.org/participate/submit-pr.html\)](https://netbeans.apache.org/participate/submit-pr.html)

[Reporting Issues](#)

[\(/https://netbeans.apache.org/participate/ report-issue.html\)](https://netbeans.apache.org/participate/ report-issue.html)

[Improving the documentation](#)

[\(/https://netbeans.apache.org/participate/ index.html#documentation\)](https://netbeans.apache.org/participate/ index.html#documentation)

[Improving the documentation](#)

[\(/https://netbeans.apache.org/participate/ index.html#documentation\)](https://netbeans.apache.org/participate/ index.html#documentation)

[GET HELP](#) [\(/HELP](#) [/INDEX.HTML\)](#)

[Documentation](#)

[\(/https://netbeans.apache.org/help/ index.html#documentation\)](https://netbeans.apache.org/help/ index.html#documentation)

[Wiki \(/wiki](#)

[/index.asciidoc\)](https://netbeans.apache.org/wiki/ index.asciidoc)

[Community](#)

[\(/https://netbeans.apache.org/help/ support/ index.html#support\)](https://netbeans.apache.org/help/ support/ index.html#support)

[Commercial](#)

[\(/https://netbeans.apache.org/help/ commercial- support.html\)](https://netbeans.apache.org/help/ commercial- support.html)

[Commercial](#)

[\(/https://netbeans.apache.org/help/ commercial- support.html\)](https://netbeans.apache.org/help/ commercial- support.html)

[DOWNLOAD](#) [\(/DOWNLOAD](#) [/NB110](#) [/NB110.HTML\)](#)

[Releases](#)

[\(/https://netbeans.apache.org/download/ index.html\)](https://netbeans.apache.org/download/ index.html)

[Plugins](#)

[\(/https://plugins.netbeans.apache.org/\)](https://plugins.netbeans.apache.org/)

[Building from source \(/download](#)

[/index.html#source\)](https://netbeans.apache.org/download/ index.html#source)

[Previous releases](#)

[\(/https://netbeans.apache.org/download/ index.html#previous\)](https://netbeans.apache.org/download/ index.html#previous)

[Previous releases](#)

[\(/https://netbeans.apache.org/download/ index.html#previous\)](https://netbeans.apache.org/download/ index.html#previous)

Copyright © 2017-2020 [The Apache Software Foundation \(https://www.apache.org\)](https://www.apache.org).

Licensed under the Apache [license \(https://www.apache.org/licenses/\)](https://www.apache.org/licenses/), version 2.0

Apache, Apache NetBeans, NetBeans, the Apache feather logo and the Apache NetBeans logo are trademarks of [The Apache Software Foundation \(https://www.apache.org\)](https://www.apache.org).

Oracle and Java are registered trademarks of Oracle and/or its affiliates.