

TUTORIAIS

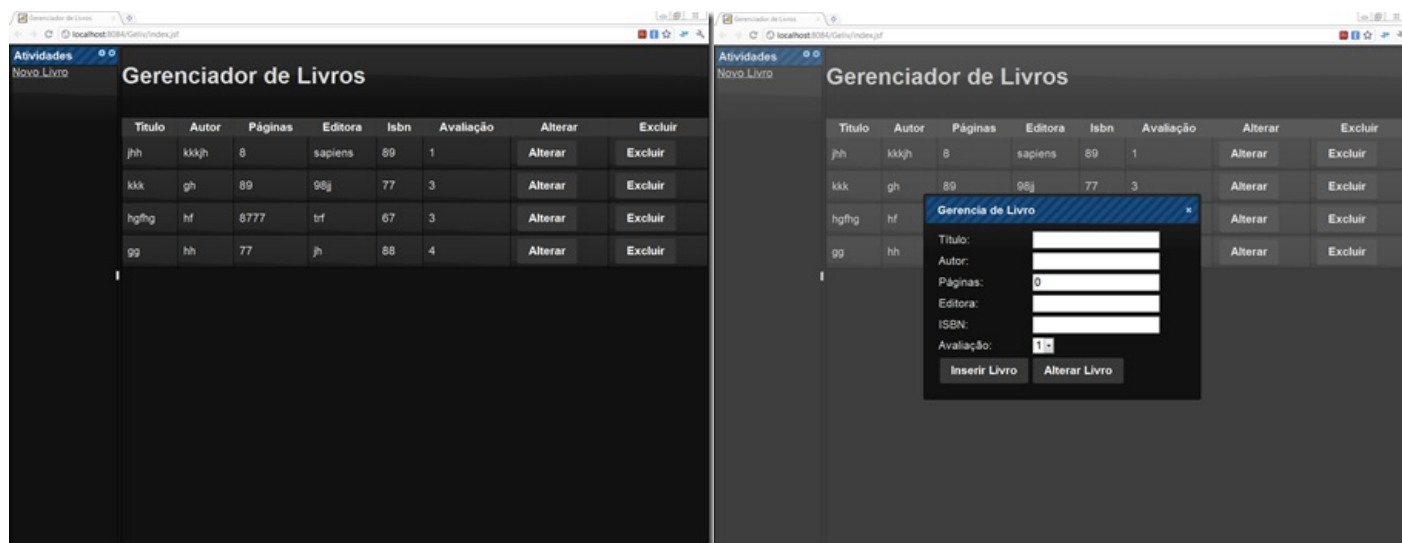
CRUD JSF 2.0 + Hibernate: Parte 2– Utilizando Primefaces

by [José Alexandre Macedo](#) on 13 de Setembro de 2010 • [88 Comments](#)

Depois de algum tempo sem postar nada devido a adaptação ao Rio e a empresa que comecei a trabalhar, [Concrete Solutions](#), volto com muitas ideias novas e tecnologias diferentes para compartilhar aqui no blog. Continuarei escrevendo sobre desenvolvimento java para web mas tratarei também de assuntos como arquitetura orientada a serviço (SOA), projeto e análise de serviços, padrões de projeto empresariais, bibliotecas java script e assuntos referentes ao mercado de trabalho de tecnologia.

Vamos descrever neste artigo algumas mudanças no [nosso CRUD JSF 2](#) para que ele passe a interagir com o usuário utilizando ajax, também vamos melhorar a interface do CRUD com a biblioteca de componentes JSF [PrimeFaces](#). Esta biblioteca de componentes foi uma das primeiras a dar suporte a nova versão do JSF, além de estar sendo constantemente melhorada pelo seu autor Cagatay Civici. Uma das últimas mudanças foi a possibilidade da escolha de [diversos skins para os componentes](#), assim como a possibilidade de utilização do suporte de criação de skin do jquery para gerar skins personalizados para os componentes do PrimeFaces através do [ThemeRoller](#). Utilizaremos skin também neste post.

O CRUD passará a ter a seguinte interface:



Pré Requisitos:

Este post é uma continuação do [CRUD JSF 2.0 + Hibernate – Exemplo Gerenciando Livros](#),

portanto é interessante que você acompanhe os passos iniciais no CRUD para que entenda o que será alterado neste.

- Biblioteca de componentes PrimeFaces disponível [aqui](#)
- Os demais pré requisitos estão descritos na [primeira parte](#) do CRUD JSF 2

Avisos:

- O desenvolvimento dos códigos foi realizado no Netbeans 6.9, porém você pode utilizar a IDE de sua preferência para construir o CRUD. Os projetos disponíveis para download no fim do tutorial é executável no Netbeans.
- A versão do PrimeFaces utilizada foi a 2.2, utilize esta ou uma superior.

Adicionando e configurando a biblioteca PrimeFaces

Após [baixar](#) a biblioteca PrimeFaces você deve adicioná-la ao projeto do CRUD de Livros. Caso não tenha ainda o projeto, [baixe-o por aqui](#) ou acompanhe a primeira parte do CRUD [aqui](#).

A configuração necessária para que o PrimeFaces funcione é ZERO, isso mesmo, não é necessário nenhuma configuração xml para começar a utilizá-lo, apenas a sua declaração na página que o utilizar. No nosso caso a página index.xhtml deverá conter a seguinte declaração:

```
1 | xmlns:p="http://primefaces.prime.com.tr/ui"
```

Para que possa ser utilizado um skin diferente na página, é necessário a inserção do seguinte código no arquivo web.xml

```
1 | <context-param>
2 | <param-name>primefaces.skin</param-name>
3 | <param-value>none</param-value>
4 | </context-param>
```

Esta configuração remove o skin padrão do PrimeFaces e permite que outros skins possam ser usados, sendo que, a especificação ocorre na página através da adição da tag . Você deve [clique aqui](#) para baixar o skin que é utilizado nas modificações deste CRUD. A pasta do skin deve ser colocada na pasta web do seu projeto. E as páginas que forem utilizar o skin devem conter a seguinte chamada dentro da tag :

```
1 | <link type="text/css" rel="stylesheet" href="dot-luv/skin.css" />
```

Porque utilizar ajax no nosso CRUD de Livros?

O ajax é a maneira que foi encontrada para que não fosse necessário recarregar toda uma página quando houvesse algumas mudanças apenas em parte dela. O objetivo do uso de ajax é tornar as páginas web mais próximas de softwares desktop e com isso poder ter uma aplicação que não depende de uma máquina específica e que pode ser acessada de qualquer lugar. Isso implica também em um novo modelo de negócio que começa a se difundir cada vez mais chamado Software as a Service (SaaS), ou, Software como um Serviço, este modelo se baseia em oferecer um serviço a diversos usuários pela web. Para o usuário a diferença está em não pagar um alto custo inicial pelo software como no modelo tradicional, além de poder acessar seu aplicativo de qualquer lugar. Para a empresa desenvolvedora a diferença está no lucro que não virá de uma vez como no modelo tradicional e também na facilidade de manutenção e updates para os usuários, pois, precisará realizar apenas uma vez. Acho que comecei a me desviar um pouco do foco então vamos voltar. Em outro momento oportuno escreverei sobre SaaS.

Utilizando os componentes PrimeFaces no CRUD

Para melhorar a aparência do CRUD e adicionar a chamada assíncrona por ajax aos métodos nós vamos adicionar alguns componentes da biblioteca PrimeFaces. Entre os componentes utilizados estão o layout, dataTable, commandLink, commandButton e o dialog.

Além de algumas mudanças nos códigos, nos descartaremos a página gerenciarLivro.xhtml e seu conteúdo irá para a página index.xhtml. Este conteúdo ficará dentro do componente dialog do primefaces e será chamado quando for necessário incluir ou alterar um livro.

Vamos ver como a página index.xhtml fica e depois vamos entender um pouco das mudanças realizadas.

```
1 <?xml version='1.0' encoding='UTF-8' ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
3 <html xmlns="http://www.w3.org/1999/xhtml"
4     xmlns:h="http://java.sun.com/jsf/html"
5     xmlns:f="http://java.sun.com/jsf/core"
6     xmlns:p="http://primefaces.prime.com.tr/ui">
7     <h:head>
8         <title>Gerenciador de Livros</title>
9         <link type="text/css" rel="stylesheet" href="dot-luv/skir
10     </h:head>
11     <h:body>
12         <p:layout fullPage="true">
13             <p:layoutUnit position="left" width="200" header="Ati
14             closable="true" collapsible="true">
15                 <h:form prependId="false">
16                     <p:commandLink value="Novo Livro" actionListe
17                     update="infosLivro" oncomplete="dialogGerLivro.show()"/>
```

```

18         </h:form>
19     </p:layoutUnit>
20     <p:layoutUnit position="center">
21         <h1>Gerenciador de Livros</h1>
22         <br/>
23         <h:form prependId="false">
24             <p:dataTable id="tabela" var="livro" value="#{
25                 <p:column>
26                     <f:facet name="header">
27                         <h:outputText value="Título"/>
28                     </f:facet>
29                     <h:outputText value="#{livro.titulo}"
30                 </p:column>
31                 <p:column>
32                     <f:facet name="header">
33                         <h:outputText value="Autor"/>
34                     </f:facet>
35                     <h:outputText value="#{livro.autor}"/
36                 </p:column>
37                 <p:column>
38                     <f:facet name="header">
39                         <h:outputText value="Páginas"/>
40                     </f:facet>
41                     <h:outputText value="#{livro.paginas}
42                 </p:column>
43                 <p:column>
44                     <f:facet name="header">
45                         <h:outputText value="Editora"/>
46                     </f:facet>
47                     <h:outputText value="#{livro.editora}
48                 </p:column>
49                 <p:column>
50                     <f:facet name="header">
51                         <h:outputText value="Isbn"/>
52                     </f:facet>
53                     <h:outputText value="#{livro.isbn}"/>
54                 </p:column>
55                 <p:column>
56                     <f:facet name="header">
57                         <h:outputText value="Avaliação"/>
58                     </f:facet>
59                     <h:outputText value="#{livro.avaliaca
60                 </p:column>
61                 <p:column>
62                     <f:facet name="header">
63                         <h:outputText value="Alterar"/>
64                     </f:facet>
65                     <p:commandButton actionListener="#{li
66 value="Alterar" update="infosLivro" oncomplete="dialogGerLivro.s
67                 </p:column>
68                 <p:column>
69                     <f:facet name="header">

```

```

70         <h:outputText value="Excluir"/>
71     </f:facet>
72     <h:commandLink action="#{livroControl
73     </p:column>
74 </p:dataTable>
75 </h:form>
76 </p:layoutUnit>
77 </p:layout>
78
79 <p:dialog header="Gerencia de Livro" widgetVar="dialogGer
80 modal="true" showEffect="slide" width="500">
81     <h:form prependId="false">
82         <h:panelGrid id="infosLivro" columns="2" style="n
83
84         <h:outputLabel for="titulo" value="Título:" /
85         <h:inputText id="titulo" value="#{livroContr
86
87         <h:outputLabel for="autor" value="Autor:" />
88         <h:inputText id="autor" value="#{livroControl
89
90         <h:outputLabel for="paginas" value="Páginas:"
91         <h:inputText id="paginas" value="#{livroContr
92
93         <h:outputLabel for="editora" value="Editora:"
94         <h:inputText id="editora" value="#{livroContr
95
96         <h:outputLabel for="isbn" value="ISBN:" />
97         <h:inputText id="isbn" value="#{livroControl
98
99         <h:outputLabel for="avaliacao" value="Avaliaç
100         <h:selectOneMenu id="avaliacao" value="#{livr
101             <f:selectItem itemLabel="1" itemValue="1"
102             <f:selectItem itemLabel="2" itemValue="2"
103             <f:selectItem itemLabel="3" itemValue="3"
104             <f:selectItem itemLabel="4" itemValue="4"
105             <f:selectItem itemLabel="5" itemValue="5"
106         </h:selectOneMenu>
107
108         <p:commandButton update="tabela" oncomplete="
109         actionListener="#{livroController.adicionarLivro}" value="Inseri
110         <p:commandButton update="tabela" oncomplete="
111         actionListener="#{livroController.alterarLivro}" value="Alterar
112
113     </h:panelGrid>
114     </h:form>
115 </p:dialog>
116
117 </h:body>
118 </html>

```

O componente layout contém um subcomponente chamado layoutUnit. Nas configurações

deste subcomponente nós informamos onde o código que estiver dentro dele vai se localizar na página. Por exemplo utilizando o `position="left"` estamos falando que todo código dentro deste componente ficará do lado esquerdo da tela. Além disso, definimos a configuração `fullPage` do layout como `true`, o que faz o layout preencher toda a tela do browser.

A principal mudança do `dataTable` está na troca dos componentes padrões do jsf pelo `dataTable` do primefaces. Isso permite que a tabela seja renderizada e seja alterada com o uso do skin, além de muitas outras funcionalidades que não serão abordadas neste post. O `dataTable` passou a ter um id que serve para identifica-lo para que fosse possível atualizar a tabela por ajax.

Os `commandLinks` e os `commandButtons` tiveram grandes alterações, pois é através deles que as chamadas ajax são realizadas. Vamos ver as principais mudanças:

- Troca do `action` pelo `actionListener`: esta troca é necessária porque não vamos mais carregar outra página após a execução de um método no servidor. Nós vamos executar os métodos no servidor e depois apenas atualizar os componentes que sofreram mudanças com a execução do método.
- O uso do atributo `update` informa quais componentes deverão ser renderizados novamente após a execução do `actionListener`, com isso o usuário passa a ter a sensação que está utilizando um programa desktop e não simplesmente acessando um site.
- O atributo `onComplete` serve para ativar o dialog que apresenta as informações de um livro para alterações ou inclusões. A ativação do dialog só ocorre após a execução do método que estiver definido no `actionListener` e também após a renderização dos componentes que estão na variável `update`.

O componente `dialog` permite a criação de uma janela modal que pode ser arrastada pela tela e torna a aplicação mais próxima ainda de aplicações desktop, pelo fato das janelas serem comuns nas aplicações desktop. Uma variável importante deste componente é o `widgetVar` que serve para definir um nome para o dialog. Com este nome o dialog pode ser apresentado ou escondido na tela através das funções javascript `nomeDoDialog.show()` e `nomeDoDialog.hide()`, respectivamente.

Estas são as informações importantes para compreender as mudanças realizadas nas páginas xhtml. Agora vamos ver como a classe `LivroController.java` deve ficar para que tudo funcione corretamente.

```
1 package Controller;  
2  
3 import Dao.LivroDao;
```



```

4  import Dao.LivroDaoImp;
5  import Model.Livro;
6  import java.util.List;
7  import javax.faces.bean.ManagedBean;
8  import javax.faces.bean.SessionScoped;
9  import javax.faces.event.ActionEvent;
10 import javax.faces.model.DataModel;
11 import javax.faces.model.ListDataModel;
12
13 /**
14  * @author José Alexandre
15  */
16 @ManagedBean
17 @SessionScoped
18 public class LivroController {
19
20     private Livro livro;
21     private DataModel listaLivros;
22
23     public DataModel getListarLivros() {
24         List<livro> lista = new LivroDaoImp().list();
25         listaLivros = new ListDataModel(lista);
26         return listaLivros;
27     }
28
29     public Livro getLivro() {
30         return livro;
31     }
32
33     public void setLivro(Livro livro) {
34         this.livro = livro;
35     }
36
37     public void prepararAdicionarLivro(ActionEvent actionEvent){
38         livro = new Livro();
39     }
40
41     public void prepararAlterarLivro(ActionEvent actionEvent){
42         livro = (Livro)(listaLivros.getRowData());
43     }
44
45     public String excluirLivro(){
46
47         Livro livroTemp = (Livro)(listaLivros.getRowData());
48         LivroDao dao = new LivroDaoImp();
49         dao.remove(livroTemp);
50         return "index";
51     }
52 }
53
54 public void adicionarLivro(ActionEvent actionEvent){
55
56     LivroDao dao = new LivroDaoImp();

```

```

56 LivroDao dao = new LivroDaoImp();
57 dao.save(livro);
58
59 }
60
61 public void alterarLivro(ActionEvent actionEvent){
62
63 LivroDao dao = new LivroDaoImp();
64 dao.update(livro);
65
66 }
67
68 }

```

A mudança realizada nesta classe está nos métodos deixarem de retornar uma String com a página de destino. Eles passam a não retornar nada e também passam a receber como parâmetro um objeto ActionEvent. Isso acontece porque o objetivo agora é apenas realizar algumas mudanças no servidor e depois atualizar somente algumas partes da página através de ajax.

Essas foram as mudanças necessárias no CRUD para que ele passasse a utilizar ajax e alguns dos componentes da biblioteca primefaces.

Conclusão

Podemos notar que com poucas mudanças tivemos um ganho enorme com relação a usabilidade e aparência do CRUD. A partir deste CRUD o usuário pode começar a explorar os diversos outros componentes disponibilizados pelo primefaces, [veja aqui os demos de todos eles](#), lembrando que existem também outras bibliotecas de componentes que o usuário pode conhecer e que podem atender melhor as necessidades de componentes de acordo com o caso.

Espero que tenham aproveitado e entendido o funcionamento do uso dos componentes primefaces e do ajax!

Por hoje é só e até a próxima! Qualquer dúvida estou a disposição!

Download

Para baixar o código fonte compatível com Netbeans [clique aqui](#).



Tags: [ajax](#) [biblioteca](#) [crud](#) [hibernate](#) [Java](#) [jsf 2](#) [primefaces](#)

88 comments for “CRUD JSF 2.0 + Hibernate: Parte 2– Utilizando Primefaces”

Pingback: [CRUD JSF 2.0 + Hibernate – Exemplo Gerenciando Livros « José Alexandre Macedo Blog](#)

Pingback: [Tweets that mention CRUD JSF 2.0 + Hibernate: Parte 2– Utilizando a biblioteca Primefaces no Gerenciador de Livros « José Alexandre Macedo Blog -- Topsy.com](#)



Marcelo

14 de Setembro de 2010 at 9:00

Muito Bom mesmo.!

Vou da uma lida neste novo post, para estudar.

Existe algumas duvidas em relação ao código java da primeira parte. Seria interessante se todo o código fosse comentado, assim ficaria mais fácil. Tem algumas coisas do hibernate que não abstrai.

Claro, o que você já fez, foi o bastante em desenvolver e publicar essa aplicação, esclareceu muitas duvidas que eu tinha em relação ao desenvolvimento JSF.

Estou desenvolvendo um sisteminha para meu tcc e irei utilizar JSF. Espero conseguir. kkkk

Agradeço em nome de todos que visualizarão seu Post.

Ótimo trabalho e Obrigado.

Abraços,

Marcelo Wanderley

Responder



José Alexandre Macedo

14 de Setembro de 2010 at 9:12

Olá Marcelo,

Caso queira me falar melhor quais são suas dúvidas em relação ao código java fique a vontade que te explico!

Espero que ajude no seu tcc... qualquer dúvida estou ai!

Abraços,

José Alexandre

Responder



Marcelo

14 de Setembro de 2010 at 9:29

Alexandre,

Eu tenho conhecimento em php.

Como não entendo muito “ainda” a sintaxe do Java, seria possível você fazer alguns comentários nas classes que foram criada?

Tipo:

– Class

Livro

LivroDaoImp

HibernateUtil

LivroController

– Interface

LivroDao

Em relação a configurações e aos Views já deu para abstrair.

Muito Obrigado, Alexandre,

Marcelo Wanderley

Responder



José Alexandre Macedo

14 de Setembro de 2010 at 9:39

Marcelo,

Assim que possível comentarei algumas classes para torna-las mais claras!

Abraços,

José Alexandre

Responder



Marcelo

14 de Setembro de 2010 at 9:47

Alexandre,

Espero não ser incomodo.

Agradeço pela atenção.

Obrigado.

Abraços,

Marcelo Wanderley

Responder



Guto

14 de Setembro de 2010 at 9:52

Alguém conseguiu rodar o exemplo no Eclipse?

Estou levando uma surra aqui... se alguém gostaria de disponibilizar o código/jar's/configs???

Ótimo artigo!

Responder



José Alexandre Macedo

14 de Setembro de 2010 at 9:58

Guto,

Você pode se basear na versão que o Edson Lopes fez da primeira parte do CRUD, [disponível aqui](#), e então fazer as modificações explicadas na parte 2!

Responder



franklin Santos

27 de Setembro de 2012 at 18:47

oi jose talvez já tem um exemplo cuandose chamadas de três mesas, onde você pode atualizar e excluir

hola jose talvez ya tengas un ejemplo cuandose llama desde tres tablas donde se pueda actualizar y eliminar

Responder



Gilliard

16 de Julho de 2013 at 23:12

José,

Cara realmente seu artigo é ótimo. Estou querendo desenvolver meu TCC e JSF 2.0. Mas estou apanhando muito pois quero usar o eclipse. O linque que você deixou aqui não está mais disponível para Dowload. Se não for abuso pode enviar por e-mail ou postar para dowload.

Desde já agradeço e parabéns.

Responder



Marcelo

15 de Setembro de 2010 at 9:29

Olá Alexandre bom dia,

Olha aqui novamente. rrsrrs

Alexandre, existe muitas diferenças entre as versões jsf 1.2 para 2.0 ?

Foi utilizado spring neste pequeno sisteminha que você fez?

O que seria Spring?

Obrigado pela força.!

Abraços,
Marcelo Wanderley

Responder



José Alexandre Macedo

15 de Setembro de 2010 at 9:52

Olá Marcelo!

Bom dia!

Existem muitas diferenças sim entre as versões 1.2 e 2 do JSF! Essa nova versão amadureceu bastante, passando a contar com muito menos configurações xml, novos tipos de escopo, criação de componentes personalizados de forma mais fácil entre muitas outras coisas, aumentando assim a produtividade no desenvolvimento!

Não utilizei spring no CRUD pois a intenção foi focar no jsf e um pouco no hibernate... poderia pensar no uso de spring no CRUD em uma 3ª ou 4ª parte...

Spring é um framework muito robusto para desenvolvimento web com java. Ele pode melhorar diversos aspectos da aplicação, como, segurança(spring security), diminuição do acoplamento (IoC). Spring também dá suporte a programação orientada a aspecto, a banco de dados, além de poder ser utilizado como um framework mvc.

É importante aprender a utiliza-lo uma hora pois sua aplicação pode ganhar muito com ele. Existem também frameworks de desenvolvimento java para web que utilizam o spring por trás, como por exemplo o vraptor.

Qualquer dúvida estou ai!

Abraços,

José Alexandre

Responder



Najib El Alam

23 de Setembro de 2010 at 17:20

José, achei muito bom o tutorial, não tive problema algum para importar o projeto e fazer funcionar. Parabéns pela boa vontade.

Alterei a dataTable para incluir paginação e ordenação. Estou com problema na ordenação. Simplesmente quando seleciono uma coluna, parece que ocorre o evento via ajax, pois a table dá uma piscada, mas a ordem dos dados permanecem.

já mudei o a dataTable pra dynamic="true" forçando a exec do ajax no servidor mas mesmo assim não tá funcionando.

you could please try to help me with this question ?

obrigado.

Responder



José Alexandre Macedo

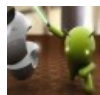
23 de Setembro de 2010 at 21:41

Olá Najib,

obrigado pelos elogios...

posso tentar te ajudar sim, você chegou a ver se apareceu algum erro no tomcat?!

Responder

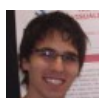


Najib El Alam

24 de Setembro de 2010 at 10:09

Não aparece nada na log do tomcat.

Responder



José Alexandre Macedo

24 de Setembro de 2010 at 17:17

Najib,

envie então o datatable pra que eu de uma olhada...

Responder



Andre

28 de Setembro de 2010 at 10:28

Opa, primeiramente parabens pelo POST, me ajudou muito. Muito bom mesmo. Continue postando amigo.

Responder



André

28 de Setembro de 2010 at 16:06

Amigo, fiz o teu exemplo junto com facelets(Template), so que na hora de abrir o formulário, o formulário fica em segundo plano.

Sabe me dizer o porque disso estar acontecendo.

Helps.



Responder



José Alexandre Macedo

28 de Setembro de 2010 at 16:43

Olá André,

Como assim fica em segundo plano?

Poderia me explicar melhor?!



Marcelo Wanderley

5 de Outubro de 2010 at 23:11

Olá Alexandre tudo bem?

Eu tentei fazer um exemplo apartir de um novo projeto. Fiz o mesmo procedimento, porém utilizei outra tabela.

Gerou este erro quando compilo:

```
org.hibernate.hql.ast.QuerySyntaxException: Pessoa is not mapped [from Pessoa]
at
org.hibernate.hql.ast.util.SessionFactoryHelper.requireClassPersister(SessionFactoryHelper.java:
158)
at
org.hibernate.hql.ast.tree.FromElementFactory.addFromElement(FromElementFactory.java:87)
at org.hibernate.hql.ast.tree.FromClause.addFromElement(FromClause.java:70)
at org.hibernate.hql.ast.HqlSqlWalker.createFromElement(HqlSqlWalker.java:255)
at org.hibernate.hql.antlr.HqlSqlBaseWalker.fromElement(HqlSqlBaseWalker.java:3056)
at org.hibernate.hql.antlr.HqlSqlBaseWalker.fromElementList(HqlSqlBaseWalker.java:2945)
at org.hibernate.hql.antlr.HqlSqlBaseWalker.fromClause(HqlSqlBaseWalker.java:688)
at org.hibernate.hql.antlr.HqlSqlBaseWalker.query(HqlSqlBaseWalker.java:544)
at org.hibernate.hql.antlr.HqlSqlBaseWalker.selectStatement(HqlSqlBaseWalker.java:281)
at org.hibernate.hql.antlr.HqlSqlBaseWalker.statement(HqlSqlBaseWalker.java:229)
at org.hibernate.hql.ast.QueryTranslatorImpl.analyze(QueryTranslatorImpl.java:228)
at org.hibernate.hql.ast.QueryTranslatorImpl.doCompile(QueryTranslatorImpl.java:160)
at org.hibernate.hql.ast.QueryTranslatorImpl.compile(QueryTranslatorImpl.java:111)
at org.hibernate.engine.query.HQLQueryPlan.(HQLQueryPlan.java:77)
at org.hibernate.engine.query.HQLQueryPlan.(HQLQueryPlan.java:56)
at org.hibernate.engine.query.QueryPlanCache.getHQLQueryPlan(QueryPlanCache.java:72)
at org.hibernate.impl.AbstractSessionImpl.getHQLQueryPlan(AbstractSessionImpl.java:133)
at org.hibernate.impl.AbstractSessionImpl.createQuery(AbstractSessionImpl.java:112)
at org.hibernate.impl.SessionImpl.createQuery(SessionImpl.java:1623)
at br.com.sadeweb.dao.PessoaDaoImp.list(PessoaDaoImp.java:36)
at br.com.sadeweb.controller.PessoaController.getListarPessoas(PessoaController.java:30)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at javax.el.BeanELResolver.getValue(BeanELResolver.java:302)
at javax.el.CompositeELResolver.getValue(CompositeELResolver.java:175)
at com.sun.faces.el.FacesCompositeELResolver.getValue(FacesCompositeELResolver.java:72)
at com.sun.el.parser.AstValue.getValue(AstValue.java:116)
at com.sun.el.parser.AstValue.getValue(AstValue.java:163)
```


at com.sun.el.ValueExpressionImpl.getValue(ValueExpressionImpl.java:219)
at com.sun.faces.facelets.el.TagValueExpression.getValue(TagValueExpression.java:102)
at javax.faces.component.ComponentStateHelper.eval(ComponentStateHelper.java:190)
at javax.faces.component.ComponentStateHelper.eval(ComponentStateHelper.java:178)
at javax.faces.component.UIData.getValue(UIData.java:554)
at javax.faces.component.UIData.getDataModel(UIData.java:1248)
at javax.faces.component.UIData.getRowCount(UIData.java:339)
at
org.primefaces.component.datatable.DataTableRenderer.encodeTbody(DataTableRenderer.java:387)
at
org.primefaces.component.datatable.DataTableRenderer.encodeMarkup(DataTableRenderer.java:152)
at
org.primefaces.component.datatable.DataTableRenderer.encodeEnd(DataTableRenderer.java:78)
at javax.faces.component.UIComponentBase.encodeEnd(UIComponentBase.java:878)
at javax.faces.component.UIComponent.encodeAll(UIComponent.java:1620)
at javax.faces.render.Renderer.encodeChildren(Renderer.java:168)
at javax.faces.component.UIComponentBase.encodeChildren(UIComponentBase.java:848)
at javax.faces.component.UIComponent.encodeAll(UIComponent.java:1613)
at javax.faces.component.UIComponent.encodeAll(UIComponent.java:1616)
at javax.faces.component.UIComponent.encodeAll(UIComponent.java:1616)
at javax.faces.component.UIComponent.encodeAll(UIComponent.java:1616)
at javax.faces.component.UIComponent.encodeAll(UIComponent.java:1616)
at
com.sun.faces.application.view.FaceletViewHandlingStrategy.renderView(FaceletViewHandlingStrategy.java:380)
at com.sun.faces.application.view.MultiViewHandler.renderView(MultiViewHandler.java:126)
at com.sun.faces.lifecycle.RenderResponsePhase.execute(RenderResponsePhase.java:127)
at com.sun.faces.lifecycle.Phase.doPhase(Phase.java:101)
at com.sun.faces.lifecycle.LifecycleImpl.render(LifecycleImpl.java:139)
at javax.faces.webapp.FacesServlet.service(FacesServlet.java:313)
at org.apache.catalina.core.StandardWrapper.service(StandardWrapper.java:1523)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:279)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:188)
at org.apache.catalina.core.StandardPipeline.invoke(StandardPipeline.java:641)
at com.sun.enterprise.web.WebPipeline.invoke(WebPipeline.java:97)
at
com.sun.enterprise.web.PESessionLockingStandardPipeline.invoke(PESessionLockingStandardPipeline.java:85)
at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:185)
at org.apache.catalina.connector.CoyoteAdapter.doService(CoyoteAdapter.java:325)
at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:226)
at com.sun.enterprise.v3.services.impl.ContainerMapper.service(ContainerMapper.java:165)

```
at com.sun.grizzly.http.ProcessorTask.invokeAdapter(ProcessorTask.java:791)
at com.sun.grizzly.http.ProcessorTask.doProcess(ProcessorTask.java:693)
at com.sun.grizzly.http.ProcessorTask.process(ProcessorTask.java:954)
at com.sun.grizzly.http.DefaultProtocolFilter.execute(DefaultProtocolFilter.java:170)
at com.sun.grizzly.DefaultProtocolChain.executeProtocolFilter(DefaultProtocolChain.java:135)
at com.sun.grizzly.DefaultProtocolChain.execute(DefaultProtocolChain.java:102)
at com.sun.grizzly.DefaultProtocolChain.execute(DefaultProtocolChain.java:88)
at com.sun.grizzly.http.HttpProtocolChain.execute(HttpProtocolChain.java:76)
at com.sun.grizzly.ProtocolChainContextTask.doCall(ProtocolChainContextTask.java:53)
at com.sun.grizzly.SelectionKeyContextTask.call(SelectionKeyContextTask.java:57)
at com.sun.grizzly.ContextTask.run(ContextTask.java:69)
at com.sun.grizzly.util.AbstractThreadPool$Worker.doWork(AbstractThreadPool.java:330)
at com.sun.grizzly.util.AbstractThreadPool$Worker.run(AbstractThreadPool.java:309)
at java.lang.Thread.run(Thread.java:619)
```

Ainda não sei ler este tipo de erro.

Poderia ajudar?

Abraços,

Marcelo Wanderley

Responder



José Alexandre Macedo

6 de Outubro de 2010 at 10:18

Olá Marcelo,

É um problema que pode estar relacionado a três coisas...

¹ Veja se a classe Pessoa está com as annotations do hibernate

² Na hora de utilizar os métodos do hibernate veja se o nome da tabela está correto, se eu não me engano, a primeira letra deve estar em maiúsculo...

³ Confira as importações para ver se está importando dos lugares corretos

Caso faça isso e não de certo você me dá um toque!

Abraços!

Responder



Marcelo

6 de Outubro de 2010 at 9:43

Olá Alexandre,

Eu tinha colocado um código ontem a noite.

Você chegou a visualizar?

abraços,

Marcelo Wanderley

Responder



José Alexandre Macedo

6 de Outubro de 2010 at 10:06

Olá Marcelo!

Não tive tempo ontem... vou dar uma olhada e respondo lá msmo!

Abraços!

Responder



Marcelo Wanderley

6 de Outubro de 2010 at 22:50

Olá Alexandre,

Obrigado pela paciência.

Eu coloquei os annotations nos métodos, verifiquei as importações. E até agora nada.

estou usando uma tabela pessoa, cujo essa tabela esta tabela usa os campos com os nomes em maiúsculos, isso influencia?

-No DAO coloquei da mesma forma, tudo em maiúsculo.

– Na index, não utilizei todos os campos da tabela, tentei manipular apenas alguns campos.

Logo abaixo vai o link do projeto, seria possível verificar onde estaria o erro, assim tento entender o que esta acontecendo.

http://rapidshare.com/files/423555756/JAVA_JSF.war

– Tenho algumas tabelas que são relacionadas. O hibernate já prepara todo o relacionamento entre as tabelas?

Muito obrigado Alexandre,

Abraços,

Marcelo Wanderley

Responder



Marcelo Wanderley

7 de Outubro de 2010 at 23:58

Oi Alexandre,

Fico no aguardo ?

Abraços,

Marcelo Wanderley

Responder



José Alexandre Macedo

8 de Outubro de 2010 at 13:39

Olá Marcelo,

Estou na correria esta semana aqui na empresa e assim que der dou uma olhada pra você OK?!
Acho melhor vc colocar o nome em minusculo na tabela... de uma olhada nisso pra ver...

Logo te respondo se demorar pode me cobrar...

Abraços!

Responder



Marcelo Wanderley

9 de Outubro de 2010 at 13:55

Ok Alexandre,

Ficarei no aguardo.

Os nomes estão em maiúsculo e mesmo assim nada =[

Abraços e Obrigado,

Marcelo Wadnerley

Responder



José Alexandre Macedo

13 de Outubro de 2010 at 9:49

Marcelo o Pedro disse que teve o mesmo problema e no caso dele estavam faltando algumas libs, veja se não é o seu caso tb e dps me avise!

Abraços!

Responder



Pedro Henrique Castro

23 de Outubro de 2010 at 17:27

Bom tarde José Alexandre! Estou fazendo meu projeto TCC como mencionado a você, porém me deparei com um problema agora, estava fazendo e rodando no Mozilla, até aí beleza roda tranquilo, fui fazer um teste no IE e não abre por nada. Sabe o que possa ser? Tem algum contato e-mail pra entrar em contato contigo? Meu e-mail: phcastro03@yahoo.com.br

Responder



Eugenio

26 de Outubro de 2010 at 1:48

Primeiramente, parabens, seu material é otimo.

To com um problema aqui, to usando seu crud como modelo. Tava tudo funcioando correntamente mas quando eu comecei a usar o primefaces na pagina ta dando esse erro.
`/view/formProduto.xhtml @19,87 value="#{produtoController.produto.nome}": Target`

Unreachable, 'produto' returned null

e nao mudei nada com relação a isso para que pudesse ocorrer o erro

Responder



José Alexandre Macedo

26 de Outubro de 2010 at 12:37

E ai Eugênio não vi que era você nas duas mensagem senão tinha respondido só em uma! Confira a classe produtoController para ver se ela está de acordo com a produtoController da 2ª parte do CRUD.. em algum lugar o objeto produto está ficando null sem poder mais veja essa classe e dps me diga se achou algo!

Abraços!

Responder



Eugênio

26 de Outubro de 2010 at 10:47

Qual a diferença do action pro actionListener? E no commandButton o que o update faz?

Responder



José Alexandre Macedo

26 de Outubro de 2010 at 12:34

Olá Eugenio,

O actionListener é um ouvinte que apenas executa alguns códigos no servidor e não precisa retornar nenhuma página (não redireciona após executar como é o caso do action). O update informa quais componentes da páginas deverão ser renderizados novamente após a execução do actionListener, estes dois atributos foram utilizados para que o ajax funcionasse!

Espero ter esclarecido, qualquer coisa vc me fala!

Abraços,

José Alexandre Macedo

Responder



Eugenio

26 de Outubro de 2010 at 19:41

Esclareceu sim, muito obrigado.

Sobre a primeira pergunta que eu fiz, o problema é que eu estava usando um menu item com um action(que deveria retornar uma pagina) so que nao estava retornando(apesar de chamar o metodo no bean) ai entao eu tinha colocado url="" no final pra ir direto pra url, mas ai nao passava pelo metodo ai eu teria um produto nulo. Ai eu resolvi da seguinte forma, tirei o "url" e coloquei ajax="false" ai agora ta funcionando direitinho. 😊

Responder



Andre Luiz

10 de Novembro de 2010 at 3:34

Ola jose, exemplo excelente esse q vc deu. Bem q vc poderia implementar um relatorio agora, nao vi ainda nenhum tutorial de relatorio, e sei q vc e o cara mais apropriado pra implementar isso. obrigado pelos tutoriais e por tudo

Responder



José Alexandre Macedo

10 de Novembro de 2010 at 10:48

Olá André,

Achei sua ideia interessante e diferente!

Vou ver se implemento ela em um dos próximos posts!

Agradeço a sugestão e sempre que tiver pode me falar!

Abraços,

José A Macedo

Responder



juniorsatanas

30 de Novembro de 2010 at 17:21

Gostei do CRUD, gostaria de por Spring Security nele, e usalo para aprender... juntando com um pouco de Relacionamento e um relatorio básico !

Poderia dizer se é possível usar Spring Security nele ?

junior

Responder



José Alexandre Macedo

30 de Novembro de 2010 at 17:29

Olá Junior!

É possível sim utilizar Spring Security nele...

Estou querendo colocar logo mas o tempo ta mto curto nos últimos dias, estou terminando um trabalho para entregar na sexta daí poderei escrever o post!

Abraços!

Responder



juniorsatanas

30 de Novembro de 2010 at 17:22

relatorio jsf 2

<http://pobrecomputeiro.wordpress.com/2010/08/14/jasperreports-para-quem-tem-jsf-2-e-o-primefaces/>

mas não conseguir por para rodar !

abraço.

Responder



José Alexandre Macedo

30 de Novembro de 2010 at 17:31

Interessante Junior, qual o problema que ocorre?!

Responder



medical Technologist

3 de Janeiro de 2011 at 4:16

this post is very usefull thx!

Responder

Pingback: [CRUD JSF 2 – Parte 3: Segurança com Spring Security 3 | José Alexandre Macedo Blog](#)



Vic Bormolini

19 de Janeiro de 2011 at 19:36

Estou alterando o bco de dados Livro para o meu específico porém estou com um erro de mapeamento.. An Error Occurred: ordemdeservico is not mapped [from ordemdeservico]
Eu mapeei em Context.xml, persistente.xml, hibernate.cfg.xml e no livroDaoImp.java.. mas o erro persiste. Será que tenho alterar em mais outro lugar?

desde já,

Responder



José Alexandre Macedo

24 de Janeiro de 2011 at 10:17

Vic,

Da uma olhada na classe HibernateUtil, veja se nela você colocou todas as classes que devem ser mapeadas pelo hibernate da seguinte forma:

```
ac.addAnnotatedClass(NOMEDACLASSE.class);
```

Se isso não resolver me avise que pensamos em outra coisa!

Abraços!

Responder



Jefferson

20 de Janeiro de 2011 at 11:32

Alexandre,

Obrigado pela resposta rápida.

Eu fiz algumas alterações pro que eu preciso.

Estou “tentando” usar JPA com EclipseLink e Majorra.

Modifiquei algumas das classes, por exemplo, a classe Livro, a LivroController e a LivroDaoImp.

Quando tento fazer como no seu exemplo, todas as informações na classe Livro, funciona legal, mas quando crio uma nova classe Autor, modifico o atributo autor na classe livro apontando para a classe Autor, e coloco um atributo na classe Autor apontando para a classe Livro (relacionamento 1 para muitos – livro-autor), e tento salvar, aparece o erro:

WARNING: /index.xhtml @86,88 value="{livroController.livro.autor.id}": Target Unreachable, 'autor' returned null

javax.el.PropertyNotFoundException: /index.xhtml @86,88 value="{#

{livroController.livro.autor.id}": Target Unreachable, 'autor' returned null

Ví em vários foruns que esse erro é básico, mas não consegui resolve-lo ainda.

Se puder ajudar, agradeço.

Classe Livro:

```
package Model;

import java.io.Serializable;
import javax.persistence.*;

/**
 *
 * @author José Alexandre
 */
@Entity
@Table(schema="livraria", name="livro")
public class Livro implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @SequenceGenerator(name="Livro_id_GENERATOR", sequenceName="Livro_id")
    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="Livro_id_GENERATOR")
    @Column(name="id")
    private long id;

    @Column(name="titulo")
    private String titulo;

    @Column(name="paginas")
    private int paginas;

    @Column(name="editora")
    private String editora;

    @Column(name="isbn")
    private String isbn;
```

```
@Column(name="avaliacao")
private int avaliacao;

//bi-directional many-to-one association to SegCadmod
@ManyToOne
@JoinColumn(name="autor")
private Autor autor;

public Livro() {
}

public Livro(long id, String titulo, Autor autor, int paginas, String editora, String isbn, int avaliacao) {
    this.id = id;
    this.titulo = titulo;
    this.autor = autor;
    this.paginas = paginas;
    this.editora = editora;
    this.isbn = isbn;
    this.avaliacao = avaliacao;
}

public Autor getAutor() {
    return autor;
}

public void setAutor(Autor autor) {
    this.autor = autor;
}

public int getAvaliacao() {
    return avaliacao;
}

public void setAvaliacao(int avaliacao) {
    this.avaliacao = avaliacao;
}

public String getEditora() {
    return editora;
}

public void setEditora(String editora) {
    this.editora = editora;
}

public long getId() {
    return id;
}

public void setId(long id) {
    this.id = id;
}

public String getIsbn() {
    return isbn;
}
}
```

```

public void setIsbn(String isbn) {
    this.isbn = isbn;
}

public int getPaginas() {
    return paginas;
}

public void setPaginas(int paginas) {
    this.paginas = paginas;
}

public String getTitulo() {
    return titulo;
}

public void setTitulo(String titulo) {
    this.titulo = titulo;
}
}

```

Classe Autor:

```

package Model;

import java.io.Serializable;
import java.util.List;
import javax.persistence.*;

@Entity
@Table(schema="livraria", name="autor")
public class Autor implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @SequenceGenerator(name="Autor_id_GENERATOR", sequenceName="Autor_id")
    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="Autor_id_GENERATOR")
    @Column(name="id")
    private long id;

    @Column(name="nome")
    private String nome;

    @Column(name="telefone")
    private String telefone;

    //bi-directional many-to-one association to livro
    @OneToMany(mappedBy="autor")
    private List livros;

    public Autor() {
    }

    public Autor(long id, String nome, String telefone) {
        this.id = id;
    }
}

```

```

this.nome = nome;
this.telefone = telefone;
}

public String getNome() {
return nome;
}

public void setNome(String nome) {
this.nome = nome;
}

public String getTelefone() {
return telefone;
}

public void setTelefone(String telefone) {
this.telefone = telefone;
}

public long getId() {
return id;
}

public void setId(long id) {
this.id = id;
}

}

public List getLivros(){
return this.livros;
}

public void setLivros(List livros){
this.livros = livros;
}
}

```

LivroController:

```

package Controller;

import Dao.LivroDao;
import Dao.LivroDaoImp;
import Model.Livro;
import java.util.List;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.SessionScoped;
import javax.faces.event.ActionEvent;
import javax.faces.model.DataModel;
import javax.faces.model.ListDataModel;

/**
 * @author José Alexandre
 */
@ManagedBean

```

```

@SessionScoped
public class LivroController {

    private Livro livro;
    @SuppressWarnings("rawtypes")
    private DataModel listaLivros;

    @SuppressWarnings({ "rawtypes", "unchecked" })
    public DataModel getListarLivros() {
        List lista = new LivroDaoImp().list();
        listaLivros = new ListDataModel(lista);
        return listaLivros;
    }

    public Livro getLivro() {
        return livro;
    }

    public void setLivro(Livro livro) {
        this.livro = livro;
    }

    public void prepararAdicionarLivro(ActionEvent actionEvent){
        livro = new Livro();
    }

    public void prepararAlterarLivro(ActionEvent actionEvent){
        livro = (Livro)(listaLivros.getRowData());
    }

    public String excluirLivro(){
        Livro livroTemp = (Livro)(listaLivros.getRowData());
        LivroDao dao = new LivroDaoImp();
        dao.remove(livroTemp);
        return "index";
    }

    public void adicionarLivro(ActionEvent actionEvent){
        LivroDao dao = new LivroDaoImp();
        dao.save(livro);
    }

    public void alterarLivro(ActionEvent actionEvent){
        LivroDao dao = new LivroDaoImp();
        dao.update(livro);
    }
}

```

Responder



José Alexandre Macedo

24 de Janeiro de 2011 at 10:30

Olá Jefferson,

Quando o tomcat retorna aquele erro que você me mandou? Nele, da pra perceber que o autor do livro não foi setado, você adicionou dados de autores e seus respectivos livros no banco?

Como você esta fazendo para adicionar novos autores, ou esta usando apenas os que estão no banco?

Como você deixa o usuário escolher o autor para o livro na hora de adicioná-lo?

Talvez você precise de um Controller para o Autor.

Abraços.

Responder



Vic Bormolini

28 de Fevereiro de 2011 at 13:03

Olá, Jose Alexandre, por um acaso, vc tem a continuidade deste exemplo? Estou com dificuldade para aplicar isso tbem..

obrigado e um abraço

Vic

Responder



José Alexandre Macedo

28 de Fevereiro de 2011 at 16:56

Olá Vic,

Tudo jóia?!

Com relação a estas modificações do Jefferson eu não cheguei a implementar não...

Ele não nos respondeu também sobre as perguntas que eu tinha feito a ele...

Abraços,

José Alexandre

Responder



Felipe Souza

31 de Janeiro de 2011 at 16:08

Ola amigo muito bom o tutorial!!! Gostaria de saber como vc fez pra colocar a imagem de fundo ou prar alterar a cor de fundo da aplicação no primefaces.

Grato

Responder



José Alexandre Macedo

31 de Janeiro de 2011 at 18:34

Olá Felipe,

Que bom que gostou!

Com relação a qual imagem você se refere?!

As cores utilizadas estão definidas no css dot-luv/skin.css que é a folha de estilos que é utilizada pelo primefaces...

Abraços,

José Alexandre.

Responder



Rafael Programando

19 de Abril de 2011 at 10:39

Olá quero que vc me ajuda no projeto JSF!...

Responder



José Alexandre Macedo

19 de Abril de 2011 at 12:57

Olá Rafael,

No que eu poderia te ajudar?!

Responder



vicente

23 de Abril de 2011 at 23:11

Parabéns e muito obrigado compartilhar seu conhecimento com todos!.

José , muito bacana esse seu blog, estava precisando de um lugar assim, sinto-me a vontade, vc não imagina o quanto já me ajudou com seus tutoriais.

Tenho uma duvida.

Existe uma forma de gerar as tabelas automaticamente?

Se sim, será que vc poderia dar uma dica, para este seu exemplo?

mais uma vez, muito obrigado!

Responder



Vicente

24 de Abril de 2011 at 21:57

Olá José, Já tenho a resposta pra esta minha duvida, vc já havia respondido esta mesma

duvida no tutorial parte 3. Consegui gerar as tabelas automaticamente! mais uma vez muito obrigado!

Responder



José Alexandre Macedo

25 de Abril de 2011 at 8:40

Olá Vicente,

Qualquer outro problema estou aí!

Abraços,

José Alexandre

Responder



Levy Carvalho

17 de Julho de 2011 at 16:36

Ola José,

gostaria de saber porque você utilizou commandLink no item EXCLUIR da tabela e não utilizou o commandButton como no item ALTERAR?

Obrigado

Responder



José Alexandre Macedo

27 de Julho de 2011 at 14:56

Olá Levy,

Na época eu estava tendo alguns problemas com o commandButton no Excluir daí optei pelo commandLink!

Abraços,

José Alexandre

Responder



Daniel

19 de Agosto de 2011 at 12:44

Olá Alexandre,

parabéns pelo post, muito bom. Referente a utilização do “CommandLink” na opção excluir, qual o procedimento para alterar o mesmo para “CommandButton”? Fiz está alteração aqui, baseado no código do do botão “Alterar” e, o item é excluído, entretanto não ocorre a atualização “automática” da tabela. Utilizando o CommandLink a atualização da tabela ocorre sem problemas.

Você poderia dar uma orientação a este respeito?

Um grande abraço,

Daniel

Responder



Robson de Araújo

30 de Setembro de 2011 at 19:49

Olá Alexandre, primeiramente parabéns pelo artigo, está excelente. Só tenho uma dúvida. Estou tentando fazer um mestre detalhe, sendo que os detalhes são inseridos através de modal no mestre. O problema é que quando eu coloco o botão que abre o Modal do detalhe dentro do form principal, aí não funciona. Você sabe se isso é um bug? Exista uma forma de eu usar formulários aninhados? Valeu.

Responder



Daniel Hideki

20 de Outubro de 2011 at 10:02

Cara, parabéns pelo blog, neste seu tutorial aprendi a facilidade de implementar temas em uma aplicação!!!!

Valew mesmo!!!!

Responder



Daniel Hideki

20 de Outubro de 2011 at 10:13

Cara, tive uns pequenos probleminhas pra poder colocar os temas pra funcionar no eclipse e acabei esquecendo de deixar uma dica pro pessoal que estiver baixando os jars de temas do primefaces.

É preciso fazer uma pequena mudança nesta configuração.

1- importar o jar do tema desejado pra pasta lib;

2- adicionar na web.xml o context-param

primefaces.THEME

nome do tema

Com isso, o tema roda legal!!!

Espero ter contribuído com esta pequena dica!!!

Responder



Paulo Costa

28 de Outubro de 2011 at 17:29

Apenas complementando a dica do Daniel Hideki:

Os temas do Primefaces estão disponíveis em <http://www.primefaces.org/themes.html>

Basta fazer o download do arquivo .jar correspondente ao tema que deseja utilizar em sua

aplicação.

Adicione a seguinte entrada no arquivo web.xml – localizado na pasta WEB-INF\web.xml do seu projeto ...

Por exemplo, se você efetuou o download do arquivo cupertino-1.0.1.jar a entrada no arquivo web.xml ficará da seguinte forma:

```
primefaces.THEME
```

```
cupertino
```

Feito isso, coloque o arquivo JAR (cupertino-1.0.1.jar) dentro do diretório WEB-INF\lib e reinicie seu servidor de aplicação!

Magicamente o tema será aplicado ao seu projeto.

Hope this helps!

– Paulo

Responder



Flavio Andrade

29 de Dezembro de 2011 at 21:55

Parabéns!!!

Muito bom tutorial!

Responder



Luiz Lahr

8 de Fevereiro de 2012 at 18:59

O post é muito bom, tanto esse quanto o anterior. Parabens

Estou recebendo a seguinte mensagem quando tento rodar o projeto... Saberria dizer o que pode ser? Sou novo em Java e estou quebrando a cabeça aqui....

/index.xhtml: Property 'prepararAdicionarLivro' not found on type Controller.LivroController

Valeu

Responder



Ciro Sobral

7 de Dezembro de 2012 at 13:52

Conseguiu resolver seu problema?

Eu to com esse mesmo problema.

Responder



Tony

24 de Março de 2013 at 20:36

Cara você conseguiu resolver esse problema?

Responder



Tony

24 de Março de 2013 at 22:12

Caras acabei de descobrir o erro, é um jar do primeface que ta faltando na pasta lib da tua aplicação. O nome dele é: primefaces-2.2.M1.jar e ta no projeto que o amigo José Alexandre Macêdo disponibilizou para download. Espero ter ajudado.

Responder



Flávio

15 de Fevereiro de 2012 at 12:29

Olá José! Muito bom o teu tutorial cara, parabéns!

Velho, quando coloco pra rodar eu recebo a seguinte mensagem:

“Cannot find component with identifier “infosLivro” in view.”

Mas o componente está lá, é o panelGrid. Na verdade copie e cole na íntegra o teu código.

Por acaso sabes o que poderia ser?

To usando o primefaces 3.1.

Obrigado e abraço! Parabéns!

Responder



Valdi Geraldo

24 de Fevereiro de 2012 at 12:28

Amigo gostei muito do seu tutorial vc está de parabens com essa ação, entretando, não consegui total sucesso, meu projeto fiz no eclipse e sempre da erro no update=”infosLivro” e no update=”tabela”, tambem não aparece o outro layoute com o menu adicionar livros.

tentei tirar os update funcionou so que não cosigo excluir, e continua sem aparecer o menu de adicionar livro, se vc tiver alguma ideia do que possa estar acontecendo por favor retornar esse comentario.

agradeço desde de já

Responder



Tiago Pires

27 de Agosto de 2012 at 21:34

Valdi, vc conseguiu a solução desse problema? Se sim...teria como me explicar qual a nova implementação?

Responder



Raphael

24 de Fevereiro de 2012 at 13:59

Estou tendo o mesmo problema que vc Flávio e estou usando o prime 3.1.1

Responder



Paulo

14 de Março de 2012 at 19:11

Boa tarde desde o Mexico pessoal!

Respondendo a pergunta do Valdi y Raphael. Esse problema eu resolvi cambiando a versao da livraria do Primefaces para a 2.1.1 y no meu caso usei el skin cupertino que también só funcionou usando o theme versao 1.1.

Quando eu tiver um pouco de tempo subo o projeto no eclipse para vcs.

Responder



juniorxdc

30 de Maio de 2012 at 6:35

e se eu quiser por mais um link, como faço pra chamar mais uma tela de cadastro...

Responder



Erick

8 de Junho de 2012 at 2:06

Oye hombre muchas gracias llevo tiempo buscando un ejemplo completo, excelente tu aporte.

Responder



Valdir

21 de Julho de 2012 at 17:35

Cara, impressionante como houve mudanças significativas entre as versões impressas e a versão disponível para download. Fiquei meio perdido mas depois após uma breve análise, consegui me situar. O post está bom, deu para ver o poder do jsf e do primeface. Viva o java.

Responder



Tiago Pires

27 de Agosto de 2012 at 21:08

Amigão...como eu estou usando o primefaces 3.3.1, estou tendo o mesmo erro dos colegas aí acima. Fica dando erro no "infosLivro". Acho que nessa nova implementação do primefaces, houveram algumas modificações e não estou conseguindo solucionar. Alguém teria idéia de como solucionar esse problema do commandLink??

Responder



Alison Rucks

6 de Setembro de 2012 at 0:11

Para quem está com erro no "infosLivro" ou "tabela", basta apenas adicionar id aos forms em que

eles pertencem, removendo o atributo prependId. Depois é só alterar o update="infosLivro" para update=":nomeDoForm:infosLivro" e update=":nomeDoForm:tabela". Lembrando também que o jeito de configurar o Hibernate mostrado nesse tutorial, passou a ser deprecated na versão 4, qualquer dúvida estou a disposição.

Responder



arllenlira

22 de Setembro de 2012 at 9:58

Não consegui...

O que seria assim:

Ficaria assim??

Pode confirmar? Fiz isso em todos colocando id's diferente pr cada form e o erro continuou 😞

Responder



franklin

1 de Outubro de 2012 at 14:55

hola alexandre soy de bolivia tu tutorial esta buenissimo no se si tendras otro con tres tablas de base de datos donde pueda insertar, actualizar y dar de baja

bueno es que en mi sistema tengo llamar datos de tres tablas incluso de cuatro

Responder



Paulo Gomes

24 de Outubro de 2012 at 12:05

Show, muito bem explicado.

Responder



João Batista Ladchuk

5 de Dezembro de 2012 at 23:50

Parabéns cara muito bom!

Responder



berdam

10 de Dezembro de 2012 at 11:43

Ola jose, muito bom o projeto por sinal, ajuda muito quem está iniciando.

O projeto rodou certo comigo, porém um único problema, não está conseguindo gravar no banco, testei o hibernate e está conectando e inserindo certo. Estou utilizando a versão certa do primefaces. porém nada é gravado. Alguma idéia do que pode ser. ??

Obrigado e sucesso.

Responder



Robson Cesar de Siqueira

1 de Maio de 2013 at 11:32

Estava tendo um problema com a versão do primeface que estou utilizando que não estava gravando.

Os métodos que continham o parâmetros "ActionEvent actionEvent" como:

`prepararAdicionarLivro(ActionEvent actionEvent)`

Não estavam sendo encontrados.

Para resolver bastou remover o parâmetro `ActionEvent actionEvent` dos métodos e voltou a funcionar.

Responder



Marcio

8 de Agosto de 2013 at 21:31

Muito bom o tutorial, gostaria de saber como faço para utilizar mapeamentos para duas tabelas que criei, ou se alguém fez esta implementação, para que possa gerar outras telas para este sistema?

Responder



Idail

22 de Agosto de 2013 at 1:14

Boa noite José, parabéns pelo trabalho gostaria de uma ajuda estou rodando o sistema no Eclipse Juno e ocorre o seguinte erro, este erro ocorre também no Netbeans o erro:

Ago 22, 2013 1:12:21 AM org.apache.catalina.core.StandardWrapperValve invoke

SEVERE: Servlet.service() for servlet [Faces Servlet] in context with path [/Gerencia] threw exception [/index.xhtml: Property 'prepararAdicionarLivro' not found on type Controller.LivroController] with root cause

javax.el.ELException: /index.xhtml: Property 'prepararAdicionarLivro' not found on type Controller.LivroController

Obrigado.

Responder



Emerson Santos

26 de Abril de 2014 at 21:49

Fala Alexandre, Boa noite!

Cara, em primeiro lugar quero lhe parabenizar pelos seus posts; são bem intuitivos!

Em segundo lugar queria uma ajuda tua, se possível. Tipo, estou executando esse projeto part 2 que você disponibilizou e não consigo. Toda vez me dá esse erro.

"Cannot find component with expression "infosLivro" referenced from "j_idt8".

"Cannot find component with expression "tabelaDeLivros" referenced from "j_idt49".

Preciso muito da tua ajuda!

Responder



Joathan

16 de Julho de 2014 at 0:52

Alison Rucks

6 de Setembro de 2012 at 0:11

Para quem está com erro no “infosLivro” ou “tabela”, basta apenas adicionar id aos forms em que eles pertencem, removendo o atributo prependId. Depois é só alterar o update=”infosLivro” para update=”:nomeDoForm:infosLivro” e update=”:nomeDoForm:tabela”. Lembrando também que o jeito de configurar o Hibernate mostrado nesse tutorial, passou a ser deprecated na versão 4, qualquer dúvida estou a disposição.

Responder

Deixar uma resposta

Escreva o seu comentário aqui...