

prime



USER GUIDE

5.0

Author

Çağatay Çivici

Table of Contents

About the Author.....	9
1. Introduction.....	10
1.1 What is PrimeFaces.....	10
2. Setup.....	11
2.1 Download.....	11
2.2 Dependencies	1%
2.3 Configuration.....	1)
2.4 Application Structure.....	1)
3. PrimeFaces Components.....	1-
3.1 Accordion.....	1-
3.2 AutoComplete.....	12
3.3 AutoSelectItem.....	10
3.4 AutoStatut.....	11
3.5 Button.....	1-
3.6 Calendar.....	5
3.7 CommandButton.....	9
3.8 CommandLink.....	-%
3.9 CommandPanel.....	--
3.10 CommandScript.....	5-
3.11 CommandTab.....	55
3.12 Editor.....	4)
3.13 Hart.....	4-
3.13.1 InputHart.....	45
3.13.2 InputText.....	45
3.13.3 InputText.....	50
3.13.4 DonutHart.....	5%
3.13.5 InputText.....	5-
3.13.6 InputText.....	54
3.13.7 InputText.....	52
3.13.8 InputText.....	20
3.13.9 InputText.....	21
3.13.10 DateInput.....	2)
3.13.11 InteractiveHart.....	2-
3.13.12 Export.....	25
3.13.13 StaticInput.....	24
3.13.14 S3Input.....	25
3.13.15 E0tender.....	22
3.13.16 InputText.....	29
3.13.17 InputText.....	9-
3.13.18 Editor.....	94
3.13.19 InputText.....	95

).15 *o(u " n.....	100
).12 *o(u " n	10%
).19 *o(u " nGrou&.....	10-
).%0 *o " " and1utton.....	105
).%1 *o " " and6in3.....	110
).%% *on+ir "	11-
).%% *on+ir " Dia(o,.....	115
).%) *ontent#(o '	112
).% - *onteOt 8 enu.....	1%0
).%5 Da hboard.....	1%)
).%4 DataE0&orter.....	1%2
).%5 DataGrid.....	1)1
).%2 Data6i t.....	1)5
).%9 DataScro((er.....	1-1
).)0 Data9ab(e.....	1-5
).)1 De+au(t*o " " and.....	150
).)% Dia(o,.....	15%
).)) Dra, : Dro&.....	155
).)).1 Dra, ,ab(e.....	155
).)).% Dro&&ab(e.....	121
).)- Doc3.....	124
).)5 Editor.....	122
).)4 E+ect.....	19%
).)5 #eedReader.....	195
).)2 #ie(d et.....	194
).)9 #i(eDo ' n(oad.....	199
).-0 #i(eU&(oad.....	%01
).-1 #ocu	%02
).-% #ra, " ent.....	%10
).-) Ga((eria.....	%1%
).-- G 8 a&.....	%15
).-5 G 8 a&In+oWindo '	%%4
).-4 Gra&hicl " a,e.....	%%5
).-5 Gro ' (.....	%)1
).-2 . ot ; ey.....	%)5
).-9 Id(e 8 onitor.....	%)2
).50 I " a,e*o " &are.....	% -0
).51 I " a,e*ro&&er.....	% - %
).5% I " a,eS ' itch.....	% -4
).5) In&(ace.....	% -9
).5- In&ut 8 a 3.....	%5)
).55 In&ut9e0t.....	%55
).54 In&ut9e0tarea.....	%40

).55 ;eyboard.....	%45
).52 6ayout.....	%50
).59 6ayoutUnit.....	%55
).40 6i,ht1o0.....	%55
).41 6in3.....	%20
).4% 6o,.....	%2)
).4) 8edia.....	%25
).4- 8e,a 8enu.....	%25
).45 8enu.....	%90
).44 8enubar.....	%94
).45 8enu1utton.....	%99
).42 8enulte ")01
).49 8e a,e.....)0-
).50 8e a,e)04
).51 8ind " a&.....)09
).5% 8u(tiSe(ect6i tbo0.....)1%
).5) <oti+ication1ar.....)15
).5- 7rder6i t.....)15
).55 7ut&ut6abe(..)%1
).54 7ut&ut !ane(..)%-
).55 7ver(ay !ane(..)%4
).52 !ane(..)%9
).59 !ane(Grid.....)%)
).20 !ane(8enu.....)5
).21 !a 'ord.....)5
).2% !hoto*a ")-%
).2) !ic36i t.....)--
).2- !o(..)50
).25 !rinter.....)5)
).24 !ro,re 1ar.....)5-
).25 Radio1utton.....)52
).22 Ratin,.....)59
).29 Re " ote*o " " and.....)4)
).90 Re etIn&ut.....)45
).91 Re i=ab(e.....)45
).9% Rin,.....)51
).9) Ro ')5-
).9- Ro ' Editor.....)55
).95 Ro ' E0&an ion.....)54
).94 Ro ' Ro,,(er.....)55
).95 Schedu(e.....)52
).92 Scro((!ane(..)25
).99 Se(ect1oo(ean1utton.....)29

).100 Se(ect1oo(ean*hec3bo0.....)91
).101 Se(ect*hec3bo0 8 enu.....)9)
).10% Se(ect 8 any1utton.....)95
).10) Se(ect 8 any*hec3bo0.....)99
).10- Se(ect 8 any 8 enu.....	-01
).105 Se(ect7ne1utton.....	-05
).104 Se(ect7ne6i tbo0.....	-05
).105 Se(ect7ne 8 enu.....	-11
).102 Se(ect7neRadio.....	-14
).109 Se&arator.....	-19
).110 S(ide 8 enu.....	-%1
).111 S(ider.....	-%-
).11% S&ot(i , ht.....	-%9
).11- Soc3et.....	-)1
).115 S&acer.....	-))
).114 S&inner.....	-)-
).115 S&(it1utton.....	-)9
).112 Sub " enu.....	- -)
).119 Stac3.....	- - -
).1%0 Stic3y.....	- -4
).1%1 Sub9ab(e.....	- -2
).1% Su " " aryRo '	- -9
).1%) 9ab.....	- 50
).1%- 9ab 8 enu.....	- 51
).1%5 9ab>ie '	- 5)
).1%4 9a, *(oud.....	- 52
).1%5 9er " ina(.....	- 41
).1%2 9he " eS ' itcher.....	- 4%
).1%9 9iered 8 enu.....	- 45
).1)0 9oo(bar.....	- 42
).1)1 9oo(barGrou& ?De&recated@.....	- 50
).1)% 9oo(ti&.....	- 51
).1)) 9ree.....	- 5-
).1)- 9ree<ode.....	- 25
).1)5 9ree9ab(e.....	- 24
).1)4 Water " ar3.....	- 90
).1)5 Wi=ard.....	- 9%
- . !artia(Renderin, and !roce in,.....	- 92
- .1 !artia(Renderin,.....	- 92
- .1.1 In+ra tructure.....	- 92
- .1.% U in, ID	- 92
- .1.) <oti+yin, U er	500
- .1.- 1it : !iece	500

-..% !artia(!roce in,.....	501
-..%.1 !artia(>a(idation.....	501
-..%.% U in , Id	50%
-..) Search E0&re ion #ra " e ' or3.....	50)
-..).1 ;ey ' ord	50)
-..).% !ri " e#ace Se(ector ?!#S@.....	50-
-..- !artia(Sub " it.....	504
5. Aava cri&t A !I.....	505
5.1 !ri " e#ace <a " e &ace.....	505
5.% A/a0 A !I.....	502
4. Dia(o, #ra " e ' or3.....	510
5. *(ient Side >a(idation.....	51-
5.1 *on+i, uration.....	51-
5.% A/a0 v <onB A/a0.....	51-
5.) Event	515
5.- 8 e a,e	515
5.5 1ean >a(idation.....	514
5.4 E0tendin, *S>.....	514
2. 9he " e	5%1
2.1 A&&(yin, a 9he " e.....	5%%
2.% *reatin, a <e ' 9he " e.....	5%)
2.) . o ' 9he " e Wor3.....	5%-
2.- 9he " in, 9i&	5%5
9. !ri " e#ace !u h.....	5%4
9.1 Setu&.....	5%4
9.% Annotation	5%5
9.) A !I.....	5%9
9.- Soc3et *o "&onent.....	5%9
9.5 !uttin, It A((9o ,ether.....	5)0
9.5.1 *ounter.....	5)0
9.5.% #ace 8 e a,e.....	5)%
9.4 9i& and 9ric3	5)-
10. !ri " e#ace 8 obi(e.....	5)5
10.1 Setu&.....	5)5
10.% !a,e	5)4
10.% <avi,ation	5)5
10.) *o "&onent	5)2
10.).1 *ontent.....	5)2
10.).% #ie(d.....	5)9
10.).) #ooter.....	5-0
10.).- . eader.....	5-1
10.).5 In&utS(ider.....	5-%
10.).4 !a,e.....	5--

10.) .5 Ran , eS(ider.....	5-5
10.) .2 S ' itch.....	5-4
10.- Render ; it.....	5-2
10.-.1 Accordion !ane(.....	5-2
10.-.% Auto *o " &(ete.....	5-2
10.-.) 1utton.....	5-2
10.-.- *a(endar.....	5-9
10.-.5 Data6i t.....	5-9
10.-.4 Dia(o,.....	5-9
10.-.5 #i(eU&(oad.....	5-9
10.-.2 Gro ' (.....	550
10.-.9 In&ut9e0t.....	550
10.-.10 In&ut9e0tarea.....	550
10.-.11 6in3.....	550
10.-.1% 8 enu.....	550
10.-.1) !ane(.....	550
10.-.1- !ane(Grid.....	551
10.-.15 Se(ect1 oo(ean *hec3bo0.....	551
10.-.14 Se(ect*hec3bo0 8 enu.....	551
10.-.15 Se(ect 8 any 1utton.....	551
10.-.12 Se(ect 8 any *hec3bo0.....	551
10.-.19 Se(ect7 ne 1utton.....	551
10.-.%0 Se(ect7 ne 8 enu.....	551
10.-.%1 Se(ect7 neRadio.....	551
10.-.%% 9ab 8 enu.....	551
10.-.%- 9ab>ie '	55%
10.-.%5 9oo(bar.....	55%
10.5 9he " e	55)
10.4 #ra " e ' or3.....	55-
10.4.1 A/a0 U&date	55-
10.4.% !a 9hrou,h E(e " ent	55-
10.4.) 6a=y !a,e	555
10.4.- 9ouch Event	555
11. Ut(i tie	555
11.1 ReCue t*onte0t.....	555
11.% E6 #unction	540
11.) E0ce&tion . and(er.....	541
11.- 1ean>a(idation 9ran +or " ation.....	54)
11.5 !ri " e#ace 6oca(e	54-
1%. !ort(et	545
1). Ri, htB9OB6e+t.....	542
1-. IDE Su&&ort.....	549
1-.1 <et1ean	549
1-.% Ec(i& e.....	550

15. !ro/ect Re ource	551
14. #AD.....	55%

About the Author

Ça atay Çivici i a "e"ber o+ AavaServer #ace E0&ert Grou&E the +ounder o+ !ri "e#ace and ! 8 * "e"ber o+ o&en ource AS# i " &(e"entation A&ache 8 y#ace . .e i a reco,ni=ed &ea3er in internationa(con+erence inc(udin, Aava7neE S&rin, 7neE Aa=oonE AA FE WBAAFE AS#Su " " itE AS#Day E *onB#e and "any (oca(event uch a AUG .

Ça atay i a(o an author and technica(revie 'er o+ a cou&(e boo3 re,ardin, 'eb a&&(ication deve(o& ent ' ith Aava and AS#. A an e0&erienced trainer& he ha trained over)00 deve(o&er on Aava EE techn(o, ie "ain(y AS# S&rin,E EA1).0 and A!A.

Ça atay i a(o 3no 'n a 7&ti "u !ri "e in AavaServer #ace *o" " unity.

1. Introduction

1.1 What is PrimeFaces?

- !ri " e#ace i an o&en source AS# co " &onent uite ' ith variou e0ten ion .
- G Rich et o+ co " &onent ? . t" (Editor Dia(o, Auto*o " &(eteE *hart and " any " ore@.
- G 1ui(tBin A/a0 ba ed on standard AS# %.0 A/a0 A!I .
- G 6i, ht ' ei, htE one /arE =eroBcon+i, uration and no reCuired de&endencie .
- G !u h u&&ort via At " o &here #ra " e ' or3.
- G 8obi(e UI 3it to create " obi(e ' eb a&&(ication .
- G S3innin, #ra " e ' or3 ' ith)5H bui(tBin the " e and u&&ort +or vi ua(the " e de i, ner too(.
- G E0ten ive docu " entation.
- G 6ar, eE vibrant and active u er co " " unity.
- G Deve(o&ed ' ith I&a ionI +ro " a&&(ication deve(o&er to a&&(ication deve(o&er .

2. Setup

2.1 Download

There are three ways to download PrimeFaces. You can either do it manually or you can define it as a dependency.

Download Manually

Three different artifacts are available for each version: binary source and bundle. Bundles contain binary source andavadoc.

<http://www.primefaces.org/downloads>

Download with Maven

Group id is `org.primefaces` and artifact id is `primefaces`.

```
<dependency>
    <groupId>org.primefaces</groupId>
    <artifactId>primefaces</artifactId>
    <version>5.0</version>
</dependency>
```

2.2 Dependencies

!ri " e#ace on(y reCuire a AA>A 5H runti " e and a AS# %.0 i" &(e" entation a " andatory de&endencie . 9here're o " e o&tiona(ibrarie +or certain +eature . 6icen e o+ a((de&endencie and any)rd &art ' or3 incor&orated are co " &atib(e ' ith the !ri " e#ace 6icen e .

Dependency	Version *	Type	Description
AS# runti " e	%.0E %.1 or %.%	ReCuired	A&ache 8 y#ace or 7rac(e 8 o/arra
iteOt	%.1.5	7&tiona(DataE0&orter ?!D#0.
a&ache &oi).5	7&tiona(DataE0&orter ?E0ce@.
ro " e	1.0	7&tiona(#eedReader.
co " " on Bi(i eu&oad	1.)	7&tiona(#i(eU&oad
co " " on Bio	%.%	7&tiona(#i(eU&oad
at " o &here	%.1.)	7&tiona(!ri " e#ace !u h

J 6i ted ver ion are te ted and 3no 'n to be 'or3in, 'ith !ri " e#ace E other ver ion o+ the e de&endencie "ay a(o 'or3 but not te ted.

JSF Runtime

!ri " e#ace 5.0 u&&ort AS# %.0E %.1 and %.% runti " e at the a" e ti " e u in, +eature detection and by not havin, co " &i(e ti " e de&endency to a &eci+ic ver ion. A a re u(t o " e +eature are on(y avai(ab(e de&endin, on the runti " e.

A ,ood e0a " &(e +or runti " e co " &atibi(ity i the &a throu,h attribute E a AS# %.% &eci+ic +eature to di &(ay dyna " ic attribute . In +o(o ' in, &a, eE &a throu,h attribute &(aceho(der on(y ,et rendered i+ the runti " e i AS# %.%.

```
<!DOCTYPE html>
<html xmlns="http://www.w3c.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui"
      xmlns:pt="http://xmlns.jcp.org/jsf/passthrough">

    <h:head>
    </h:head>

    <h:body>
        <p:inputText value="#{bean.value}" pt:placeholder="Watermark here"/>
    </h:body>

</html>
```

2.3 Configuration

!ri " e#ace doe not require any " mandatory configuration and +o((o ' configuration by exception &attern o+ Java EE. . ere i the (i t o+ a((configuration option defined ' ith a conteOB&ara" such a K

```
<context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>bootstrap</param-value>
</context-param>
```

Name	Default	Description
9 . E 8 E	ari to	9he " e o+ the a&&(ication.
" obi(e.9 . E 8 E	nu((9he " e o+ the " obi(e a&&(ication.
!US . LSER>ERLUR6	nu((*u to " erver ur(+or !ri " e#ace !u h.
SU1 8 I9	+u((De+ine a/a0 ub " it " ode full or partial.
DIR	(tr	De+ine orientation: ltr or rtl.
RESE9L>A6UES	0	When enab(ed) a/a0 u&dated in&ut are re et +ir t.
SE*RE9	&ri " e+ace	Secret key to encry&t decry&t va(ue e0&re ion e0&o ed in renderin , Strea " ed*ontent .
*6IE<9LSIDEL>A6IDA9I7<	0	*ontro(c(ient ide va(idation.
U!67ADER	auto	De+ine u&(oader " ode auto native or commons.
9RA<S#7R 8L8 E9ADA9A	+a(e	9ran +or " bean va(idation " etadata to ht " (attribute .

2.4 Hello World

Once you have added the &ri " e+ace /ar to your c(a &athE you need to add the !ri " e#ace na " e &ace to your &a,e to be,in u in , the co " &onent . . ere i a i" &(e &a,e (i3e te t.0ht " (

```
<!DOCTYPE html>
<html xmlns="http://www.w3c.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:p="http://primefaces.org/ui">

    <h:head></h:head>

    <h:body>
        <p:editor />
    </h:body>

</html>
```

When you run thi &a,e throu,h #ace Serv(et " a&&in, e.,. J./ +E you hou(d ee a rich te0t editor ' hen you run the &a,e ' ith te t./ +.

3. Component Suite

3.1 AccordionPanel

AccordionPanel is a container component that displays content in stacked or "at".

Godfather Part I

The story begins as Don Vito Corleone, the head of a New York Mafia family, oversees his daughter's wedding. His beloved son Michael has just come home from the war, but does not intend to become part of his father's business. Through Michael's life the nature of the family business becomes clear. The business of the family is just like the head of the family, kind and benevolent to those who give respect, but given to ruthless violence whenever anything stands against the good of the family.

Godfather Part II

Godfather Part III

Info

9a,	accordionPanel
*o " &onent *(a	org.primefaces.component.accordionpanel.Accordionpanel
*o " &onent 9y&e	org.primefaces.component.AccordionPanel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.AccordionPanelRenderer
Renderer *(a	org.primefaces.component.accordionpanel.AccordionPanelRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier of the component
rendered	true	boolean	Boolean value to specify the rendering of the component.
binding,	null	Object	An E6 expression that adds to a server-side UI component instance in a backing bean.
activeIndex	-1	String,	Index of the active tab or a component separated string, or index when "unselected" mode is on.



```
<p:accordionPanel dynamic="true">
    //..tabs
</p:accordionPanel>
```

Client Side Callbacks

onTabChange is called before a tab is shown and *onTabShow* is called after. Both receive container events sent to the tab to handle the area after.

```
<p:accordionPanel onTabChange="handleChange(panel)">
    //..tabs
</p:accordionPanel>

<script type="text/javascript">
    function handleChange(panel) {
        //panel: new tab content container
    }
</script>
```

Ajax Behavior Events

tabChange is the one and only ajax behavior event of accordion panels that is executed when a tab is selected.

```
<p:accordionPanel>
    <p:ajax event="tabChange" listener="#{bean.onChange}" />
</p:accordionPanel>
```

```
public void onChange(TabChangeEvent event) {
    //Tab activeTab = event.getTab();
    //...
}
```

Mour (tener?) is defined to be invoked with an *org.primefaces.event.TabChangeEvent* instance that contains a reference to the new active tab and the accordion panel it belongs to.

Dynamic Number of Tabs

When the tabs to display are not static use the built-in iteration feature (refer to [ui:repeat](#)).

```
<p:accordionPanel value="#{bean.list}" var="listItem">
    <p:tab title="#{listItem.propertyA}">
        <h:outputText value="#{listItem.propertyB}" />
        ...More content
    </p:tab>
</p:accordionPanel>
```

Disabled Tabs

A tab can be disabled by setting the `disabled` attribute to true.

```
<p:accordionPanel>
    <p:tab title="First Tab Title" disabled="true">
        <h:outputText value= "Lorem"/>
        ...More content for first tab
    </p:tab>
    <p:tab title="Second Tab Title">
        <h:outputText value="Ipsum" />
    </p:tab>
    //any number of tabs
</p:accordionPanel>
```

Multiple Selection

By default only one tab at a time can be active. Enabling `multiple="true"` mode to activate multiple tabs.

```
<p:accordionPanel multiple="true">
    //tabs
</p:accordionPanel>
```

Client Side API

Widget `PrimeFaces.widget.AccordionPanel`

Method	Params	Return Type	Description
<code>select(index)</code>	index Index of tab	void	Activate tab with index, given index.
<code>deselect(index)</code>	index Index of tab	void	Deactivate tab with index, given index.

Skinning

Accordion!ane(re ide in a "ain container e(e "ent ' hich `style` and `styleClass` o&&ly. A 3innin, ty(e c(a e are ,(oba(ee the "ain the "in, ection +or "ore in+or "ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.uiBaccordion	8 ain container e(e "ent
.uiBaccordionBheader	9ab header
.uiBaccordionBcontent	9ab content

3.2 AjaxBehavior

Ajax behavior is an extension to standard Java.

Info

Namespace	ajax
Behavior Id	org.primefaces.component.AjaxBehavior
Behavior Class	org.primefaces.component.behavior.ajax.AjaxBehavior

Attributes

Name	Default	Type	Description
method	null	String	Method to invoke in 'reCue t.
immediate	false	boolean	True (value that determines if the method is executed at the same time as the action or after the value of the method is invoked).
async	false	boolean	When set to true, a/a0 reCue t are not queued.
oncomplete	null	String	*o " &onent? @ to &roce in &artia(reCue t.
update	null	String	*o " &onent? @ to update with a/a0.
onstart	null	String	*a((bac3 to execute before a/a0 reCue t is begin).
oncomplete	null	String	*a((bac3 to execute when a/a0 reCue t is completed).
onsuccess	null	String	*a((bac3 to execute when a/a0 reCue t succeeds).
onerror	null	String	*a((bac3 to execute when a/a0 reCue t fails).
,(obtainable)	true	boolean	Globally a/a0 reCue t are obtained by a/a0Status. If " &onent is set, ,,(obtainable) is false and the error a/a0Status.
delay	null	String	If (e than delay) is (the second argument is between 0 and request) only the most recent one is sent and all other reCue t are discarded. If this option is not specified or if the value of delay is the iterator, none without the quote is no delay is used.
&artia(Sub) it	false	boolean	Enables aria(=ation of value be(on, in, to the &artia((y &roce ed co " &onent on(y).
disabled	false	boolean	Disables a/a0 behavior.
event	null	String	*(identifies event to trigger a/a0 reCue t).
repeatCount	false	boolean	If true, (value of in&ut co " &onent to be updated) within the a/a0 reCue t is used as repeat.

Name	Default	Type	Description
i, noAutoUpdate	+a(e	1oo(ean	I+ trueE co " &onent ' hich autoU&datePItrueI ' i((not be u&dated +or thi reCue t. I+ not &eci+iiedE or the va(ue i +a(eE no uch indication i " ade.

Getting Started with AjaxBehavior

A/a01ehavior i attached to the co " &onent to a/a0i+y.

```
<h:inputText value="#{bean.text}">
    <p:ajax update="out" />
</h:inputText>
<h:outputText id="out" value="#{bean.text}" />
```

In the e0a " &(e aboveE each ti " e the in&ut chan,e E an a/a0 reCue t i ent to the erver. When the re &on e i received out&ut teot ' ith id IoutI i u&dated ' ith va(ue o+ the in&ut.

Listener

In ca e you need to e0ecute a " ethod on a bac3in , beanE de+ine a (i tenerK

```
<h:inputText id="counter">
    <p:ajax update="out" listener="#{counterBean.increment}" />
</h:inputText>
<h:outputText id="out" value="#{counterBean.count}" />
```

```
public class CounterBean {
    private int count;
    //getter setter
    public void increment() {
        count++;
    }
}
```

Events

De+au(t c(ient ide event are de+ined by co " &onent that u&&ort c(ient behavior E +or in&ut co " &onent it i *onchange* and +or co " " and co " &onent it i *onclick*. In order to override the do " event to tri,,er the a/a0 reCue t u e *event* o&tion. In +o((o ' in, e0a " &(eE a/a0 reCue t i tri,,ered ' hen 3ey i u& on in&ut +ie(d.

```
<h:inputText id="firstname" value="#{bean.text}">
    <p:ajax update="out" event="keyup"/>
</h:inputText>
<h:outputText id="out" value="#{bean.text}" />
```

3.3 AjaxExceptionHandler

Ajax exception . and(er i a utility component for the build exception . and(er.

Info

9a,	ajaxExceptionHandler
*o " &onent *(a	org.primefaces.component.ajaxexceptionhandler.AjaxExceptionHandler
*o " &onent 9y&e	org.primefaces.component.AjaxExceptionHandler
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	nu((Object	An expression that refers to a server-side UI component instance in a backing bean.
oneException	nu((String,	Identified code to execute after an exception occurs.
update	nu((String,	Component to update after an exception occurs.
type	nu((String,	Exception type to handle.

Getting Started with AjaxExceptionHandler

!See [find the detailed information about this component at Exception . and\(er section at Utilities chapter.](#)

3.4 AjaxStatus

AjaxStatus is a component or a renderer that .



Info

9a,	ajaxStatus
*o " &onent *(a	org.primefaces.component.ajaxstatus.AjaxStatus
*o " &onent 9y&e	org.primefaces.component.AjaxStatus
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.AjaxStatusRenderer
Renderer *(a	org.primefaces.component.ajaxstatus.AjaxStatusRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify whether the component is rendered.
binding	nu((Object	An expression that binds to a server-side UI component in a bean.
onstart	nu((String,	Script to execute after a component is rendered.
oncomplete	nu((String,	Script to execute after a component is rendered.
onsuccess	nu((String,	Script to execute after a component is rendered.
onerror	nu((String,	Script to execute when an error occurs.
type	nu((String,	Inline type for the component.
type*(a	nu((String,	Style class for the component.
' id, et>ar	nu((String,	<a href="#" for the client-side ' id, et.

Getting Started with AjaxStatus

AjaxStatus uses facets to represent the different states of an ajax request. The facets are *start* and *complete*. Start facet will be visible once an ajax request begins and stay visible until it's completed. Once the ajax response is received and updated, it becomes hidden and continues to accept other requests.

```
<p:ajaxStatus>
    <f:facet name="start">
        <p:graphicImage value="ajaxloading.gif" />
    </f:facet>
    <f:facet name="complete">
        <h:outputText value="Done!" />
    </f:facet>
</p:ajaxStatus>
```

Events

There are five available events:

default Initial visibility when loaded.

start Before an ajax request begins.

success When an ajax response is received without error.

error When an ajax response is received with an HTTP error.

complete When everything is finished.

```
<p:ajaxStatus>
    <f:facet name="error">
        <h:outputText value="Error" />
    </f:facet>

    <f:facet name="success">
        <h:outputText value="Success" />
    </f:facet>

    <f:facet name="default">
        <h:outputText value="Idle" />
    </f:facet>

    <f:facet name="start">
        <h:outputText value="Sending" />
    </f:facet>

    <f:facet name="complete">
        <h:outputText value="Done" />
    </f:facet>
</p:ajaxStatus>
```

Custom Events

Facets are the declarative way to use them if you'd like to extend advanced cases with custom logic, you can take advantage of onEvent which are the event handlers attached to the facet.

```
<p:ajaxStatus onstart="alert('Start')" oncomplete="alert('End')"/>
```

A co " " an u a, e o+ &ro, ra " " atic a&&roach i to i " &(e " ent a cu to " tatu dia(o, k

```
<p:ajaxStatus onstart="PF('status').show()" oncomplete="PF('status').hide()"/>

<p:dialog widgetVar="status" modal="true" closable="false">
    Please Wait
</p:dialog>
```

Client Side API

Wid, et PrimeFaces.widget.AjaxStatus

Method	Params	Return Type	Description
tri, ,er?event@	event@	void	9ri, ,er ,iven event.

Skinning

A/a0Statu i eCui&&ed ' ith style and styleClass. Sty(in, direct(y a&&ie to a container e(e " ent ' hich contain the +acet .

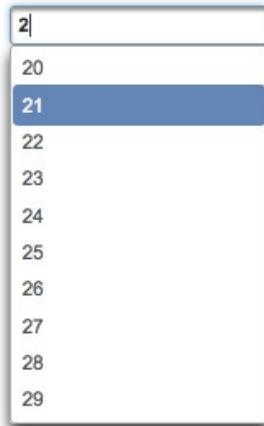
```
<p:ajaxStatus style="width:32px;height:32px" ... />
```

Tips

- Avoid u&datin, a/a0Statu it e(+ to &revent du&(icate +acet)ca((bac3 bindin, .
- !rovide a +iOed ' idth hei ht to an in(ine a/a0Statu to &revent &a, e (ayout +ro " chan,in,.
- *o " &onent (i3e co " " and1utton ha an attribute ?global) to contro(tri, ,erin, o+ A/a0Statu .
- A/a0Statu a(o u&&ort core AS# a/a0 reCue t o+ +Na/a0 a ' e().

3.4 AutoComplete

AutoComplete provides a live suggestion feature for input fields.



Info

9a,	autoComplete
*o " &onent *(a	org.primefaces.component.autocomplete.AutoComplete
*o " &onent 9y&e	org.primefaces.component.AutoComplete
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.AutoCompleteRenderer
Renderer *(a	org.primefaces.component.autocomplete.AutoCompleteRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
value	null	Object	The value of the component than can be either an Expression or a converter.
converter	null	Object	An expression or a converter that defines a converter for the component. When it's an Expression, it's resolved to a converter instance. In case it's a static text, it's used refer to a converter id.

Name	Default	Type	Description
i " " ediate	+a(e	1oo(ean	When et trueE &roce va(idation (o, ic i e0ecuted at a&&(y reCue t va(ue &ha e +or thi co " &onent.
reCured	+a(e	1oo(ean	8 ar3 co " &onent a reCured.
va(idator	nu((8 ethodE0&r	A " ethod e0&re ion that re+er to a " ethod va(idation , the in&ut.
va(ue*han,e6i tener	nu((8 ethodE0&r	A " ethod e0&re ion that re+er to a " ethod +or hand(in, a va(uchan,eevent.
reCured 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen reCured +ie(d va(idation +ai(.
converter 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen va(idation +ai(.
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
co " &(ete 8 ethod	nu((8 ethodE0&r	8 ethod &rovidin, u, ,e tion .
var	nu((Strin,	<a " e o+ the iterator u ed in &o/o ba ed u, ,e tion.
ite " 6abe(nu((Strin,	6abe(o+ the ite " .
ite " >a(ue	nu((Strin,	>a(ue o+ the ite " .
" a0Re u(t	un(i " ited	Inte,er	8 a0i " u " nu " ber o+ re u(t to be di &(ayed.
" inDuery6en,th	true	Inte,er	<u " ber o+ character to be ty&ed be+ore tartin, to Cuery.
CueryDe(ay)00	Inte,er	De(ay to ' ait in " i((i econd be+ore endin, each Cuery to the erver.
+orceSe(ection	+a(e	1oo(ean	When enab(ed auto*o " &(ete on(y acce&t in&ut +ro " the e(ection (i t.
on tart	nu((Strin,	*(ient ide ca((bac3 to e0ecute be+ore a/a0 reCue t to (oad u, ,e tion be,in .
onco " &(ete	nu((Strin,	*(ient ide ca((bac3 to e0ecute a+ter a/a0 reCue t to (oad u, ,e tion co " &(ete .
,(oba(true	1oo(ean	De+ine ' hether to tri,,er a/a0Statu or not.
cro(. ei ,ht	nu((Inte,er	De+ine the hei ,ht o+ the ite " vie ' &ort.
e+ect	nu((Strin,	E+ect to u e ' hen ho ' in,(hidin, u, ,e tion .
e+ectDuration	-00	Inte,er	Duration o+ e+ect in " i((i econd .
dro&do ' n	+a(e	1oo(ean	Enab(e dro&do ' n " ode ' hen et true.
&ane(Style	nu((Strin,	In(ine ty(e o+ the ite " container e(e " ent.

Name	Default	Type	Description
&ane(Sty(e*(a	nu((Strin,	Sty(e c(a o+ the ite " container e(e " ent.
" u(ti&(e	nu((1 oo(ean	When trueE enab(e " u(ti&(e e(ection.
&roce	nu((Strin,	*o " &onent? @ to &roce on Cuery reCue t.
acce 3ey	nu((Strin,	Acce 3ey that ' hen &re ed tran +er +ocu to the in&ut e(e " ent.
a(t	nu((Strin,	A(ternate te0tua(de cri&tion o+ the in&ut +ie(d.
autoco " &(ete	nu((Strin,	*ontro(bro ' er autoco " &(ete behavior.
dir	nu((Strin,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
di ab(ed	+a(e	1 oo(ean	Di ab(e in&ut +ie(d
(abe(nu((Strin,	A(oca(i=ed u er &re entab(e na " e.
(an,	nu((Strin,	*ode de cribin, the (an,ua,e u ed in the ,enerated " ar3u& +or thi co " &onent.
" a0(en,th	nu((Inte,er	8 a0i " u " nu " ber o+ character that " ay be entered in thi +ie(d.
onb(ur	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu .
onchan,e	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu and it va(ue ha been " odi+ied ince ,ainin, +ocu .
onc(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i c(ic3ed.
ondb(c(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i doub(e c(ic3ed.
on+ocu	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent receive +ocu .
on3eydo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over in&ut e(e " ent.
on3ey&re	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over in&ut e(e " ent.
on3eyu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over in&ut e(e " ent.
on " ou edo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over in&ut e(e " ent
on " ou e " ove	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin in&ut e(e " ent.
on " ou eout	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer

Name	Default	Type	Description
			button i " oved a ' ay +ro " in&ut e(e " ent.
on " ou eover	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on " ou eu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over in&ut e(e " ent.
on e(ect	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
&(aceho(der	nu((Strin ,	S&eci+ie a hort hint.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
i=e	nu((Inte ,er	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
ty(e	nu((Strin ,	In(ine ty(e o+ the in&ut e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the in&ut e(e " ent.
tabinde0	nu((Inte ,er	!o ition o+ the in&ut e(e " ent in the tabbin, order.
tit(e	nu((Strin ,	Advi ory too(ti& in+or " aton.
ite " 9i& 8 y!o ition	(e+t to&	Strin ,	!o ition o+ ite " ti& corner re(ative to ite " .
ite " 9i&At!o ition	ri ,ht botto "	Strin ,	!o ition o+ ite " corner re(ative to ite " ti&.
cache	+a(e	1oo(ean	When enab(ed autoco " &(ete cache the earched re u(t (i t.
cache9i " eout)00000	Inte ,er	9i " eout va(ue +or cached re u(t .
e " &ty 8 e a,e	nu((Strin ,	9e0t to di &(ay ' hen there i no data to di &(ay.
a&&end9o	nu((Strin ,	A&&end the over(ay to the e(e " ent de+ined by earch e0&re ion. De+au(t to docu " ent body.

Getting Started with AutoComplete

Auto*o " &(ete i an in&ut co " &onent o it reCuire a va(ue a u ua(. Su,,e tion are (oaded by ca((in, a erver ide co " &(ete 8 ethod that ta3e a in,(e trin , &ara " eter ' hich i the te0t entered.

```
<p:autoComplete value="#{bean.text}" completeMethod="#{bean.complete}" />
```

```

public class Bean {
    private String text;
    public List<String> complete(String query) {
        List<String> results = new ArrayList<String>();
        for (int i = 0; i < 10; i++)
            results.add(query + i);

        return results;
    }
    //getter setter
}

```

Pojo Support

So to use the `autoComplete` instead of `inputText`, you would need to do "a bit more" than just your object. You would need to implement the `Converter` interface. This is because PrimeFaces needs to convert the user input into the object type. If you don't implement the converter, it will use the default converter which converts the user input into a `String`. To avoid this, you can use the `converter` attribute of the `autoComplete` component.

```

<p:autoComplete value="#{playerBean.selectedPlayer}"
    completeMethod="#{playerBean.completePlayer}"
    var="player"
    itemLabel="#{player.name}"
    itemValue="#{player}"
    converter="playerConverter"/>

```

```

public class PlayerBean {

    private Player selectedPlayer;

    public Player getSelectedPlayer() {
        return selectedPlayer;
    }

    public void setSelectedPlayer(Player selectedPlayer) {
        this.selectedPlayer = selectedPlayer;
    }

    public List<Player> complete(String query) {
        List<Player> players = readPlayersFromDatasource(query);

        return players;
    }
}

```

```

public class Player {
    private String name;
    //getter setter
}

```

Limiting the Results

When the number of results can be limited by default there is no limit.

```
<p:autoComplete value="#{bean.text}"
    completeMethod="#{bean.complete}"
    maxResults="5" />
```

Minimum Query Length

If default queries are sent to the server and consider the method instead of a user start typing, at the initial point. This behavior is tuned using the `minQueryLength` attribute.

```
<p:autoComplete value="#{bean.text}" completeMethod="#{bean.complete}"
    minQueryLength="3" />
```

With this setting, the user will only trigger the query when the third character at the initial typed.

Query Delay

Automatic queries are triggered when the `queryDelay` option is set to a value greater than 0. If you want to tune the delay between consecutive queries, it is possible to do so after the first one. #0 in, auto*o" &ete 'ait +or 1 second after user type an input.

```
<p:autoComplete value="#{bean.text}" completeMethod="#{bean.complete}"
    queryDelay="1000" />
```

Custom Content

Automatic queries can display custom content by defining a template.

```
<p:autoComplete value="#{autoCompleteBean.selectedPlayer}"
    completeMethod="#{autoCompleteBean.completePlayer}"
    var="p" itemValue="#{p}" converter="player">

    <p:column>
        <p:graphicImage value="/images/barca/#{p.photo}" width="40" height="50"/>
    </p:column>

    <p:column>
        #{p.name} - #{p.number}
    </p:column>
</p:autoComplete>
```

Dropdown Mode

When dropdown mode is enabled a dropdown button is displayed next to the input field. This button triggers a search with the current query and calls the `complete` method.

```
<p:autoComplete value="#{bean.text}" completeMethod="#{bean.complete}"
    dropdown="true" />
```



Multiple Selection

Auto-complete supports multiple selection by setting the `multiple="true"` attribute. This allows users to select multiple items from the dropdown.

```
<p:autoComplete id="advanced" value="#{autoCompleteBean.selectedPlayers}"
    completeMethod="#{autoCompleteBean.completePlayer}"
    var="p" itemLabel="#{p.name}" itemValue="#{p}" converter="player"
    multiple="true">

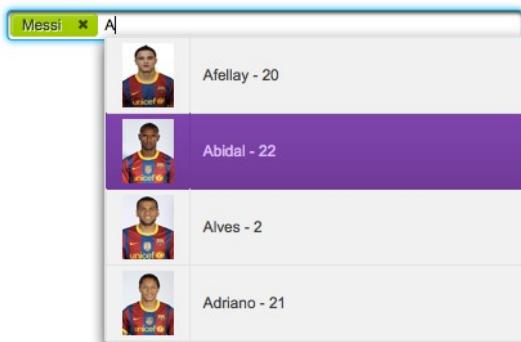
    <p:column style="width:20%;text-align:center">
        <p:graphicImage value="/images/barca/#{p.photo}"/>
    </p:column>

    <p:column style="width:80%">
        #{p.name} - #{p.number}
    </p:column>
</p:autoComplete>
```

```
public class AutoCompleteBean {

    private List<Player> selectedPlayers;

    //...
}
```



Caching

Selection can be cached on client side so that the query does not do a recursive hit which is useful to return the same selection again. To enable this set the `cache` option to true. Here is an example:

```
<p:autoComplete value="#{bean.text}" completeMethod="#{bean.complete}"
    cache="true"/>
```

Ajax Behavior Events

In stead o+ 'aitin, +or u er to ub " it the +or " "anua((y to &roce the e(eected ite " E you can enab(e in tant a/a0 e(ection by u in, the *itemSelect* a/a0 behavior. E0a " &(e be(o ' de " on trate ho ' to di &(ay a " e a,e about the e(eected ite " in tant(y.

```
<p:autoComplete value="#{bean.text}" completeMethod="#{bean.complete}">
    <p:ajax event="itemSelect" listener="bean.handleSelect" update="msg" />
</p:autoComplete>

<p:messages id="msg"/>
```

```
public class Bean {

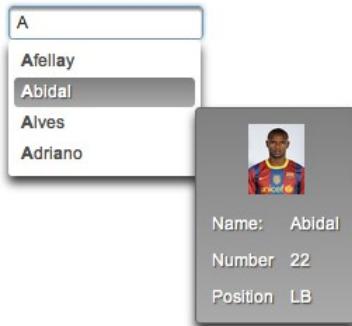
    public void handleSelect(SelectEvent event) {
        Object item = event.getObject();
        FacesMessage msg = new FacesMessage("Selected", "Item:" + item);
    }
    //...
}
```

Mour (i tener?i+ de+ined@ ' i((be invo3ed ' ith an *org.primefaces.event.Select* in tance that contain a re+erence to the e(eected ite ". <ote that auto*o " &(ete a(o u&&ort event inherited +ro " re,u(ar in&ut teOt uch a b(urE+ocu E " ou eover in addition to *itemSelect*. Si " i(ar(yE *itemUnselect* event i &rovided +or " u(ti&(e autoco " &(ete ' hen an ite " i re" oved by c(ic3in, the re " ove icon. In thi ca e *org.primefaces.event.Unselect* in tance i &a ed to a (i tener i+ de+ined.

Event	Listener Parameter	Fired
ite " Se(ect	or ,.&ri " e+ace .event.Se(ectEvent	7n ite " e(ection.
ite " Un e(ect	or ,.&ri " e+ace .event.Un e(ectEvent	7n ite " un e(ection.
Cuery	B	7n Cuery.

ItemTip

Ite " ti& i an advanced bui(tBin too(ti& ' hen " ou e i over on u,,e ted ite " . *ontent o+ the too(ti& i de+ined via the *itemtip* +acet.



```

<p:autoComplete value="#{autoCompleteBean.selectedPlayer1}" id="basicPojo"
    completeMethod="#{autoCompleteBean.completePlayer}"
    var="p" itemLabel="#{p.name}" itemValue="#{p}" converter="player">
    <f:facet name="itemtip">
        <h:panelGrid columns="2" cellpadding="5">
            <f:facet name="header">
                <p:graphicImage value="/images/barca/#{p.photo}" />
            </f:facet>

                <h:outputText value="Name: " />
                <h:outputText id="modelNo" value="#{p.name}" />

                <h:outputText value="Number " />
                <h:outputText id="year" value="#{p.number}" />

                <h:outputText value="Position " />
                <h:outputText value="#{p.position}" />
        </h:panelGrid>
    </f:facet>
</p:autoComplete>

```

Client Side API

Wid, etn *PrimeFaces.widget.AutoComplete*

Method	Params	Return Type	Description
search?va(ue@	va(ue@ 3ey ' ord +or search	void	Initiate a search ' ith ,iven va(ue
c(o e?@	B	void	. ide u,,e ted ite " " enu
di ab(e?@	B	void	Di ab(e the in&ut +ie(d
enab(e?@	B	void	Enab(e the in&ut +ie(d
deactivate?@	B	void	Deactivate search behavior
activate?@	B	void	Activate search behavior

Skinning

#o((o ' in , i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.uiBautoco " &(ete	*ontainer e(e " ent.
.uiBautoco " &(eteBin&ut	In&ut +ie(d.
.uiBautoco " &(eteB&ane(*ontainer o+ u,,e tion (i t.
.uiBautoco " &(eteBite "	6i t o+ ite "
.uiBautoco " &(eteBite "	Each ite " in the (i t.
.uiBautoco " &(eteBCuery	. i,h(i,hted &art in u,,e tion .

A 3innin , ty(e c(a e are ,(oba(E ee the " ain the " in , ection +or " ore in+or " ation.

Tips

- Do not forget to use a converter when using converters.
- Enable forceSelection if you'd like to accept values only from the dropdown list.
- Increase query delay to avoid unnecessary load to the server when using dropdowns.
- Use the <f:selectItem> option to provide feedback to the user that there are no options.
- Enable caching, to avoid duplicate queries.

3.5 BlockUI

BlockUI is used to add interactivity to components by adding a modal dialog.

Ajax Pagination			
Model	Year	Manufacturer	Color
9816c1c9	2001	Opel	Yellow
43fb87ae	1993	Renault	White
e2cb6c1a	1998	Mazda	White
aac257b5	1984		Green
79a3295d	1990		White
65d3de65	1980		Silver
61752724	2009	Opel	Red
6620632	1993	Volkswagen	White
3066aee3	1998	Audi	Black
3f309492	1991	Renault	Black

Info

9a,	blockUI
*o " &onent *(a	org.primefaces.component.blockui.BlockUI
*o " &onent 9y&e	org.primefaces.component.BlockUI
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.BlockUIRenderer
Renderer *(a	org.primefaces.component.blockui.BlockUIRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	null	Object	An expression that refers to a server-side UI component in a backing bean.
target	null	String	<a href="#" of the client id target.
target	null	String	Identifier of the component to bind.



Client Side API

Wid, et\ PrimeFaces.widget.BlockUI

Method	Params	Return Type	Description
show()	B	void	Blocks the UI.
hide()	B	void	Unblocks the UI

Skinning

#o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.ui-blockui	*ontainer e(e " ent.
.ui-blockui-content	*ontainer +or cu to " content.

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

Tips

G 1(oc3UI doe not u&&ort ab o(ute or +i0ed &o itioned co " &onent . e., dia(o,.

3.6 BreadCrumb

A breadcrumb navigation component that provides context about the hierarchy in the application.

The screenshot shows a breadcrumb navigation bar with the following items: Home > Sports > Football > Countries > Spain > F.C. Barcelona > Squad > Lionel Messi.

Info

9a,	breadCrumb
*o " &onent *(a	org.primefaces.component.breadcrumb.BreadCrumb
*o " &onent 9y&e	org.primefaces.component.BreadCrumb
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.BreadCrumbRenderer
Renderer *(a	org.primefaces.component.breadcrumb.BreadCrumbRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
clientId	null	String	The client-side ID of the component.
name	null	String	The name of the component.
style	null	String	Style of the container element.
styleClass	null	String	Style class of the container element.
icon	None	String	Defines the icon of the root (initial) value where icons default to the first item.

Getting Started with BreadCrumb

Steps are defined as children "enuite" components in breadcru "b.

```
<p:breadcrumb>
    <p:menuitem label="Categories" url="#" />
    <p:menuitem label="Sports" url="#" />
    //more menuitems
</p:breadcrumb>
```

Dynamic Menus

8 enu can be created &ro, ra " " atica((y a ' e((E ee the dyna" ic " enu &art in " enu co" &onent ection +or " ore in+or " ation and an e0a " &(e.

Skinning

1 readcru " b re ide in a container e(e " ent that *style* and *styleClass* o&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiBbreadcru " b	8 ain breadcru " b container e(e " ent.
.uiBbreadcru " b .uiB " enuBite " B(in3	Each " enuite " .
.uiBbreadcru " b .uiB " enuBite " Bte0t	Each " enuite " (abe.
.uiBbreadcru " bBchevron	Se&erator o+ " enuite " .

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

Tips

G If there is a dyna" ic +(o 'E u e " ode(o&tion in tead o+ creatin, dec(arative &N " enuite " co" &onent and bind your 8 enu 8 ode(re&re entin, the tate o+ the +(o '.

G 1 readcru " b can do a/a0QnonBa/a0 action reCue t a ' e((ince &N " enuite " ha thi o&tion. In thi ca eB breadcru " b " u t be ne ted in a +or " .

- ur(o&tion i the 3ey +or a " enuite " E i+ it i de+ined it ' i((' or3 a a i" &(e (in3. If you'd like to u e " enuite " to e0ecute co " " and ' ith or ' ithout a/a0E do not de+ine the ur(o&tion.

3.7 Button

Button is an extension to the standard `h:button` component with 3 variants, each with its own icon.



Info

9a,	button
*o " &onent *(a	org.primefaces.component.button.Button
*o " &onent 9y&e	org.primefaces.component.Button
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ButtonRenderer
Renderer *(a	org.primefaces.component.button.ButtonRenderer

Attributes

Name	Default	Type	Description
id	nu((String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	nu((Object	An expression that refers to a server-side UI component instance in a backing bean.
' id, et>ar	nu((String	<a href= the client-side ' id, et.
value	nu((Object	>a href= the component than can be either an Expression or a (iteration).
outcome	nu((String	Used to resolve a navigation case.
includeUrl	+a(e	Boolean	Whether to include a URL in the target URI.
target	nu((String	Identifier of the target and the one which should be triggered to.
disabled	+a(e	Boolean	Disabled button.
accessKey	nu((String	Access key that is rendered as a accelerator key.
alt	nu((String	Alternate text description.
dir	nu((String	Direction indication for text that does not inherit directionality. >align value are LTR and RTL.
i18n	nu((String	Style class for the button icon. Recommended value icon1
label	nu((String	Label of the button, the label used in the generated HTML &

Name	Default	Type	Description
			+or thi co "&onent.
onb(ur	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen button (o e +ocu .
onchan,e	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen button (o e +ocu and it va(ue ha been " odi+ed ince ,ainin, +ocu .
onc(ic3	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen button i c(ic3ed.
ondb(c(ic3	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen button i doub(e c(ic3ed.
on+ocu	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen button receive +ocu .
on3eydo ' n	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over button.
on3ey&re	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over button.
on3eyu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over button.
on " ou edo ' n	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over button.
on " ou e " ove	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin button
on " ou eout	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " button.
on " ou eover	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto button.
on " ou eu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over button.
ty(e	nu((Strin ,	In(ine ty(e o+ the button.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the button.
read7n(y	+a(e	1oo(ean	8 a3e button read on(y.
tabinde0	nu((Inte,er	!o ition in the tabbin, order.
tit(e	nu((Strin ,	Advi ory too(ti& in+or " aton.
hre+	nu((Strin ,	Re ource to (in3 direct(y to i " &(e " ent anchor behavior.
icon	nu((Strin ,	Icon o+ the button.
icon !o	(e+t	Strin ,	!o ition o+ the button icon.
tar,et	L e(+	Strin ,	9he ' indo ' tar,et.
e ca&e	true	1oo(ean	De+ine ' hether (abe(' ou(d be e ca&ed or not.
in(ine	+a(e	Strin ,	Di &(ay button a in(ine in tead o+ 100R ' idthE " obi(y

Name	Default	Type	Description
			on(y.
di ab(e*(ientWindo '	+a(e	1oo(ean	Di ab(e a&&endin , the *(ientWindo ' on the renderin , o+ thi e(e " ent.

Getting Started with Button

&button u a,e i a"e a standard hbutton& an outcome i nece ary to navi,ate u in, GE9 reCue t . A u" e you are at source.0ht " (and need to navi,ate tar,et.0ht " (.

```
<p:button outcome="target" value="Navigate"/>
```

Parameters

!ara " eter in URI are de+ined ' ith ne ted S+&ara " QT ta, .

```
<p:button outcome="target" value="Navigate">
    <f:param name="id" value="10" />
</p:button>
```

Icons

Icon +or button are de+ined via c and icon attribute& i+ you u e tit(e in tead o+ va(uel on(y icon ' i((be di &(ayed and tit(e te0t ' i((be di &(ayed a too(ti& on " ou eover. Mou can a(o u e icon +ro " !ri " e#ace the " e uch uiBiconBchec3.

```
<p:button outcome="target" icon="star" value="With Icon"/>
<p:button outcome="target" icon="star" title="With Icon"/>
```

```
.star {
    background-image: url("images/star.png");
}
```

Skinning

1utton render a button ta, ' hich style and styleClass a&&ie . A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiBbutton	1utton e(e " ent
.uiBbuttonBte0tBon(y	1utton e(e " ent ' hen icon i not u ed
.uiBbuttonBte0t	6abe(o+ button

3.8 Cache

*ache co " &onent i u ed to reduce &a,e (oad ti " e by cachin , the content a+ter initia(renderin ,.

Info

9a,	cache
*o " &onent *(a	org.primefaces.component.cache.UICache
*o " &onent 9y&e	org.primefaces.component.Cache
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.UICacheRenderer
Renderer *(a	org.primefaces.component.cache.UICacheRenderer

Attributes

Name	Default	Type	Description
id	null	String	UniCue identi+ier o+ the co " &onent.
rendered	true	Boolean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onent ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	null	7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean.
di ab(ed	+a(e	1oo(ean	Di ab(e cachin ,.
re ,ion	>ie ' Id	String	UniCue id o+ the cache re ,ionE de+au(t to vie ' id.
3ey	null	String	UniCue id o+ the cache entry in re ,ionE de+au(t to c(ient id o+ co " &onent.

Getting Started with Cache

A cache tore i reCquired to u e the cache co " &onentE t ' o di+erent &rovider are u&&orted a cache i " &(e " entationE . *ache and . a=e(ca t . !rovider i con+i,ured via a conte0t&ara " .

```
<context-param>
    <param-name>primefaces.CACHE_PROVIDER</param-name>
    <param-value>org.primefaces.cache.EHCacheProvider</param-value>
</context-param>
```

. ere i a a " &(e ehcache.O "(to con+i,ure cache re ,ion E there are t ' o re ,ion in thi con+i,uration.

```

<?xml version="1.0" encoding="UTF-8"?>
<ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ehcache.xsd"
  updateCheck="true" monitoring="autodetect"
  dynamicConfig="true">
  <diskStore path="java.io.tmpdir"/>
  <defaultCache
    maxEntriesLocalHeap="10000"
    eternal="false"
    timeToIdleSeconds="120"
    timeToLiveSeconds="120"
    diskSpoolBufferSizeMB="30"
    maxEntriesLocalDisk="10000000"
    diskExpiryThreadIntervalSeconds="120"
    memoryStoreEvictionPolicy="LRU">
    <persistence strategy="localTempSwap"/>
  </defaultCache>
  <cache name="testcache"
    maxEntriesLocalHeap="10000"
    eternal="false"
    timeToIdleSeconds="120"
    timeToLiveSeconds="120"
    diskSpoolBufferSizeMB="30"
    maxEntriesLocalDisk="10000000"
    diskExpiryThreadIntervalSeconds="120"
    memoryStoreEvictionPolicy="LRU">
    <persistence strategy="localTempSwap"/>
  </cache>

```

After the configuration at UI side the cached part need to be registered in side the <cache co " &onent.

```

<p:cache>
  //content to cache
</p:cache>

```

Once the &a, e is loaded initially content in side <cache co " &onent is cached in side the cache re, ion o+ the cache provider. !o tba3 on the a " e &a, e or re&enin, the &a, e retrieve the out&ut +ro " cache in stead o+ renderin, the content re, u(ar(y).

Cache Provider API

*ache !rovider can be acce ed via

RequestContext.getCurrentInstance().getApplicationContext().getCacheProvider()

#or else in , thi A !I a((cache re, ion can be c(eaned u in, *clear()* " ethod. Re+er to /avadoc o+ *ache !rovider +or the +u(((i t o+ avai(ab(e " ethod .

3.9 Calendar

The `Calendar` component is an input component used to select a date feature, displaying a date picker for selection. It includes a date selection and a calendar.



Info

9a,	calendar
*o " &onent *(a	org.primefaces.component.calendar.Calendar
*o " &onent 9y&e	org.primefaces.component.Calendar
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.CalendarRenderer
Renderer *(a	org.primefaces.component.calendar.CalendarRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
value	null	/java.util.Date	Value of the component.
converter	null	Converter	An expression or an iterator that defines a converter for the component. When it's an iterator, it's resolved to a converter instance. In case it's a static iterator, it's used to refer to a converter id.
immediate	false	Boolean	When set to true, validation (optional) is executed at the same time as the value change and the component.
required	false	Boolean	Specifies if the component is required.

Name	Default	Type	Description
validator	null	Method	A " method expression that refers to a " method validation, the input
valueChangeListener	null	Method	A " method expression that refers to a " method for hand(in, a valuechange event)
required	null	String	Specifies a message to be displayed if the required validation failed.
converter	null	String	Specifies a message to be displayed if the converter failed.
validatorMessage	null	String	Specifies a message to be displayed if the validation failed.
id, et>ar	null	String	 id, et>
" indate	null	Date or String	Set calendar initial date
" a0date	null	Date or String	Set calendar initial date
&a, e	1	Integer	Enables " update rendering.
disabled	false	Boolean	Disables the calendar when set to true.
" ope	&oult;	String	Defines how the calendar will be displayed.
&attern	8 8 dd/yyyy	String	Date#or format pattern (calendarization)
(ocafe)	null	Object	Specifies a value to be updated or (abandon) and conversion.
&o&u&Icon	null	String	Icon on the &oult;button
&o&u&Icon7n(y)	false	Boolean	When enabled icon is rendered without the button.
navi, ator	false	Boolean	Enablesontheyear navigator
ti " eUone	null	String, or a /ava.util.Ri " eUone instance to convert the ti " e=one updated or date conversion depending on the ti " eUone., etDe+au(t?)	String, or a /ava.util.Ri " eUone instance to convert the ti " e=one updated or date conversion depending on the ti " eUone., etDe+au(t?)
readonlyIn&ut	false	Boolean	Readonly input field on a &oult;caendar readonly.
ho ' 1utton!ane(false	Boolean	Today button contains today and done button.
effect	null	String	Effect to use when displaying, and hide in, the &oult;caendar.
effectDuration	normal	String	Duration of the effect.
ho ' 7n	both	String	*rient ide event that displays the &oult;caendar.
ho ' Wee3	false	Boolean	Displays the ' ee3 number ber next to each ' ee3.
disabledWee3end	false	Boolean	Disabled ' ee3end condition .

Name	Default	Type	Description
ho'7ther8onth	+a(e	1oo(ean	Di &(ay day be(on,in, to other "onth .
e(ect7ther8onth	+a(e	1oo(ean	Enab(e e(ection o+ day be(on,in, to other "onth .
yearRan,e	nu((Strin,	Mear ran,e +or the navi, atorE de+au(t IcB10Nch10I
ti"e7n(y	+a(e	1oo(ean	Sho ' on(y ti"e&ic3er' ithout date.
te&.our	1	Inte,er	.our te& .
te&8inute	1	Inte,er	8inute te& .
te&Second	1	Inte,er	Second te& .
"in.our	0	Inte,er	8ini"u" boundary +or hour e(ection.
"a0.our	%)	Inte,er	8a0i"u" boundary +or hour e(ection.
"in8inute	0	Inte,er	8ini"u" boundary +or "inute e(ection.
"a08inute	59	Inte,er	8a0i"u" boundary +or hour e(ection.
"inSecond	0	Inte,er	8ini"u" boundary +or econd e(ection.
"a0Second	59	Inte,er	8a0i"u" boundary +or econd e(ection.
&a,edate	nu((7b/ect	Initia(date to di &(ay i+ va(ue i nu((.
acce3ey	nu((Strin,	Acce 3ey that ' hen &re ed tran +er +ocu to the in&ut e(e ent.
a(t	nu((Strin,	A(ternate te0tua(de cri&tion o+ the in&ut +ie(d.
autoco"&(ete	nu((Strin,	*ontro(bro' er autoco" &(ete behavior.
dir	nu((Strin,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
(abe(nu((Strin,	A(oca(i=ed u er &re entab(e na" e.
(an,	nu((Strin,	*ode de cribin, the (an,ua,e u ed in the ,enerated "ar3u& +or thi co" &onent.
"a0(en,th	nu((Inte,er	8a0i"u" nu" ber o+ character that "ay be entered in thi +ie(d.
onb(ur	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e" ent (o e +ocu .
onchan,e	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e" ent (o e +ocu and it va(ue ha been "odified ince ,ainin, +ocu .
onc(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute onc(ic3 event.
ondb(c(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e" ent

Name	Default	Type	Description
			i doub(e c(ic3ed.
on+ocu	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent receive +ocu .
on3eydo ' n	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over in&ut e(e " ent.
on3ey&re	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over in&ut e(e " ent.
on3eyu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over in&ut e(e " ent.
on " ou edo ' n	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over in&ut e(e " ent
on " ou e " ove	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin in&ut e(e " ent.
on " ou eout	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " in&ut e(e " ent.
on " ou eover	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on " ou eu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over in&ut e(e " ent.
on e(ect	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
&(aceho(der	nu((Strin ,	S&eci+ie a hort hint.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&event chan,e by the u er.
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.
i=e	nu((Inte ,er	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
tabinde0	nu((Inte ,er	!o ition o+ the in&ut e(e " ent in the tabbin , order.
tit(e	nu((Strin ,	Advi ory too(ti& in+or " aton.
be+oreSho ' Day	nu((Strin ,	*(ient ide ca((bac3 to e0ecute be+ore di &(ayin , a dateE u ed to cu to " i=e date di &(ay.
" a 3	nu((Strin ,	A&&(ie a " a 3 u in , the &attern.

Getting Started with Calendar

> a(ue o+ the ca(endar hou(d be a /ava.uti(.Date.

```
<p:calendar value="#{dateBean.date}" />
```

```
public class DateBean {  
    private Date date;  
    //Getter and Setter  
}
```

Display Modes

* a(endar ha t 'o "ain di &(ay "ode E *popup* ?de+au(t@ and *inline*.

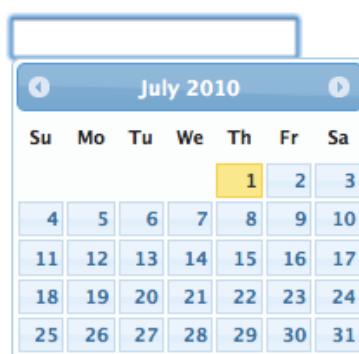
Inline

```
<p:calendar value="#{dateBean.date}" mode="inline" />
```



!o&u&

```
<p:calendar value="#{dateBean.date}" mode="popup" />
```



showOn o&tion de+ine the c(ient ide event to di &(ay the ca(endar. >a(id va(ue are

- focus When in&ut +ie(d receive +ocu
- button When &o&u& button i c(ic3ed
- both 1oth *focus* and *button* ca e

!o&u& 1utton

```
<p:calendar value="#{dateBean.date}" mode="popup" showOn="button" />
```



!o&u& Icon 7n(y

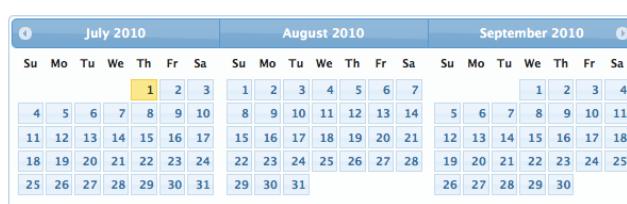
```
<p:calendar value="#{dateBean.date}" mode="popup"
            showOn="button" popupIconOnly="true" />
```



Paging

*a(en)dar can a(o be rendered in " u(ti&e &a,e ' here each &a,e corre &ond to one " onth. 9hi +eature i tuned ' ith the *pages* attribute.

```
<p:calendar value="#{dateController.date}" pages="3"/>
```



Localization

1y de+au(t (oca(e in+or " ation i retrieved +ro " the vie ' ' (oca(e and can be overridden by the (oca(e attribute. 6oca(e attribute can ta3e a (oca(e 3ey a a Strin, or a /ava.uti(.6oca(e in tance. De+au(t (an,ua,e o+ (abe(are En,(i h and you need to add the nece ary tran (ation to your &a,e " anua((y a !ri " e#ace doe not inc(ude (an,ua,e tran (ation . !ri " e#ace Wi3i !a,e +or

!ri " e#ace 6oca(e i a co " "unity driven &a,e ' here you " ay +ind the tran (ation you need.
!(ea e contribute to thi ' i3i ' ith your o ' n tran (ation .

<http://wiki.primefaces.org/display/Components/PrimeFaces+Locales>

9ran (ation i a i" &(e /ava cri&t ob/ectE ' e u , ,e t addin , the code to a /ava cri&t +i(e and inc(ude in your a&&ication. #o((o ' in , i a 9ur3i h ca(endar.

```
<h:outputScript name="path_to_your_translations.js"/>

<p:calendar value="#{dateController.date}" locale="tr" navigator="true"
             showButtonPanel="true"/>
```



9o override ca(cu(ated &attern +ro " (oca(eE u e the &attern o&tion

```
<p:calendar value="#{dateController.date1}" pattern="dd.MM.yyyy"/>
<p:calendar value="#{dateController.date2}" pattern="yy, M, d"/>
<p:calendar value="#{dateController.date3}" pattern="EEE, dd MMM, yyyy"/>
```

dd.MM.yyyy <input type="text" value="06.07.2010"/>	yy, M, d <input type="text" value="10, 7, 13"/>	EEE, dd MMM, yyyy <input type="text" value="Fri, 23 Jul, 2010"/>
--	---	--

Effects

gariou e++ect can be u ed ' hen ho ' in , and hidin , the &o&u& ca(endarE o&tion areK ho ' E
(ideDo ' nE +adelnE b(indE bounceE c(i&E dro&E +o(d and (ide.

Ajax Behavior Events

*a(endar &rovide a *dateSelect* a/a0 behavior event to e0ecute an in tant a/a0 b S P b Xetia0d ' U evioru a

```
<p:calendar value="#{calendarBean.date}">
    <p:ajax event="dateSelect" listener="#{bean.handleDateSelect}" update="msg" />
</p:calendar>

<p:messages id="msg" />
```

```
public void handleDateSelect(SelectEvent event) {
    Date date = (Date) event.getObject();
    //Add facesmessage
}
```

In order to define a date range, you can use the `mindate` and `maxdate` attributes.

Date Ranges

Using `mindate` and `maxdate` attributes, the date can be restricted. The `mode` attribute can either be a `button`, or a `/java.util.Date`.

```
<p:calendar value="#{dateBean.date}" mode="inline"
    mindate="07/10/2010" maxdate="07/15/2010"/>
```



Navigator

The navigator is an easy way to navigate through the year quickly.

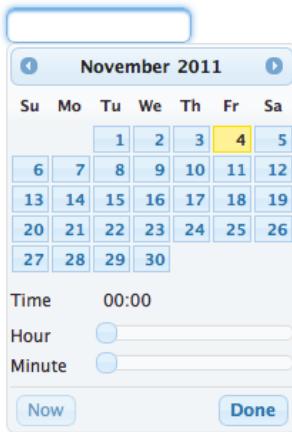
```
<p:calendar value="#{dateBean.date}" mode="inline" navigator="true" />
```



TimePicker

The TimePicker functionality is enabled by adding the `timePattern` attribute to your calendar.

```
<p:calendar value="#{dateBean.date}" pattern="MM/dd/yyyy HH:mm" />
```



Advanced Customization

Use `beforeShowDay` callback to control which days are enabled. The function returns an array with three values: one for each day of the week, to indicate if the date should be disabled or enabled and secondarily whether it is the optional type (e.g. to add to date `ceilingDay`, `endDay` and `riday`).

```
<p:calendar value="#{dateBean.date}" beforeShowDay="tuesdaysAndFridaysOnly" />
```

```
Function tuesdaysAndFridaysDisabled(date) {
    var day = date.getDay();

    return [(day != 2 && day != 5), '']
}
```

Mask

endar has a built-in feature in its internal calendar component. Set `mask` option to true to enable a mask.

Client Side API

Widet `PrimeFaces.widget.Calendar`

Method	Params	Return Type	Description
<code>getSelectedDate()</code>		Date	Return selected date
<code>setSelectedDate(Date)</code>	Date	void	Set selected date
<code>enable()</code>		void	Enable calendar
<code>disable()</code>		void	Disable calendar

Skinning

endar resides in a container element which `style` and `styleClass` option apply.

#o((o ' in, in the (i to+ tructural type class e K

Style Class	Applies
.ui-date&ic3er	8 ain container
.ui-date&ic3erBheader	. eader container
.ui-date&ic3erB&rev	!reviou " onth navi ,ator
.ui-date&ic3erBne0t	<e0t " onth navi ,ator
.ui-date&ic3erBtit(e	9it(e
.ui-date&ic3erB " onth	8 onth di &(ay
.ui-date&ic3erBtab(e	Date tab(e
.ui-date&ic3erB ' ee3end	6abe(o+ ' ee3end
.ui-date&ic3erBotherB " onth	Date be(on,in, to other " onth
.ui-date&ic3er td	Each ce((date
.ui-date&ic3erBbutton&ane	1utton &ane(
.ui-date&ic3erBcurrent	9oday button
.ui-date&ic3erBc(o e	*(o e button

A 3innin , ty(e c(a e are ,(oba(E ee the " ain the " in , ection +or " ore in+or " ation.

3.10 Captcha

*a&tcha i a +or " va(idation co " &onent ba ed on Reca&tcha A !!.



Info

9a,	captcha
*o " &onent *(a	org.primefaces.component.captcha.Captcha
*o " &onent 9y&e	org.primefaces.component.Captcha
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.CaptchaRenderer
Renderer *(a	org.primefaces.component.captcha.CaptchaRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression that adds to a server-side UI component instance in a backing bean.
value	null	Object	>a value of the component than can be either an Expression or a Iterator.
converter	null	Converter	An expression or a iterator that defines a converter for the component. When it's an Expression, it's resolved to a converter instance. In case it's a static iterator, it's used to refer to a converter id.
immediate	false	Boolean	When set true, validation (optional) is executed at all levels before the component.
required	false	Boolean	Specifies a component as required.

Name	Default	Type	Description
validator	null	Method	A " method binding, equivalent to a " method reference to a " method validation, the input.
valueChangeListener	null	Method	A " method binding, equivalent to a " method for handling a valuechange event.
required	null	String	Attribute to be displayed when required is true.
converter	null	String	Attribute to be displayed when conversion fails.
validatorMessage	null	String	Attribute to be displayed when validation fails.
label	red	String	Label for the captcha.
(an,ua,e)	en	String	Label for the user-entered (name,username).
tabIndex	null	Integer	TabIndex of the input element in the tabbing order.
(ab)	null	String	User &re enterable label.
secure	false	Boolean	Enable https support.

Getting Started with Captcha

To start using a captcha component, you need to add a private key and a public key to your web.xml file. Once you have the keys, you can add the "captcha" attribute to your input element.

```
<context-param>
    <param-name>primefaces.PRIVATE_CAPTCHA_KEY</param-name>
    <param-value>YOUR_PRIVATE_KEY</param-value>
</context-param>

<context-param>
    <param-name>primefaces.PUBLIC_CAPTCHA_KEY</param-name>
    <param-value>YOUR_PUBLIC_KEY</param-value>
</context-param>
```

Now that it's no longer necessary to use a captcha, you can use it in your application.

```
<p:captcha />
```

Themes

The theme feature is available in PrimeFaces 3.0 and newer versions. It allows you to change the look and feel of the component by specifying a theme name. The themes available are:

- Red
- White
- Black
- Green
- Blue

The themes are applied via the "theme" attribute.

```
<p:captcha theme="white"/>
```



Languages

The output of the CAPTCHA component can be customized by setting the `language` attribute. For example, if you set `language="tr"`, the generated text will be in Turkish.

```
<p:captcha language="tr"/>
```

Overriding Validation Messages

You can override validation messages for the CAPTCHA component by using the `message` and `messagelabel` attributes. These attributes can be used to define specific validation messages for different validation types.

Su " " ary	&ri " e+ace .ca&tcha.l<>A6ID
Detai(&ri " e+ace .ca&tcha.l<>A6IDLdetai(

Tips

- Use the `message` option to provide readable error messages in case of validation failure.
- Enable the `secure` option to protect against other attacks like cross-site scripting and injection.
- See [here](#) for more information about how to use CAPTCHA correctly.

3.11 Carousel

*arou e(i a " u(ti &ur&o e co " &onent to di &(ay a et o+ data or ,enera(content ' ith (ide effect .



Info

9a,	carousel
*o " &onent *(a	org.primefaces.component.carousel.Carousel
*o " &onent 9y&e	org.primefaces.component.Carousel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.CarouselRenderer
Renderer *(a	org.primefaces.component.carousel.CarouselRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1 oo(ean	1 oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in , bean
va(ue	nu((7b/ect	A va(ue e0&re ion that re+er to a co((ection
var	nu((Strin ,	<a " e o+ the reCue t co&ed iterator
nu " >i ib(e)	Inte ,er	<u " ber o+ vi ib(e ite " &er &a ,e
+ir t>i ib(e	0	Inte ,er	Indeo o+ the +ir t e(e " ent to be di &(ayed
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et.
circu(ar	+a(e	1 oo(ean	Set continuou cro((in ,
vertica(+a(e	1 oo(ean	Set vertica(cro((in ,
auto ! (ayInterva(0	Inte ,er	Set the ti " e in " i((i econd to have *arou e(tart

Name	Default	Type	Description
			cro((in, auto " atica((y a+ter bein, initia(i=ed
&a,e6in3)	Inte,er	De+ine the nu " ber o+ &a, e (in3 o+ &a, inator.
e+ect	(ide	Strin,	<a " e o+ the ani " ationE cou(d be “+ade” or “ (ide”.
ea in,	ea eln7ut*irc	Strin,	<a " e o+ the ea in, ani " ation.
e+ectDuration	500	Inte,er	Duration o+ the ani " ation in " i((i econd .
dro&do ' n9e " &(ate.	V&a,eW	Strin,	9e " &(ate trin, +or dro&do ' n o+ &a, inator.
ty(e	nu((Strin,	In(ine ty(e o+ the co " &onent..
ty(e*(a	nu((Strin,	Sty(e c(a o+ the co " &onent..
ite " Sty(e	nu((Strin,	In(ine ty(e o+ each ite " .
ite " Sty(e*(a	nu((Strin,	Sty(e c(a o+ each ite " .
header9e0t	nu((Strin,	6abe(+or header.
+ooter9e0t	nu((Strin,	6abe(+or +ooter.

Getting Started with Carousel

*arou e(ha t' o " ain u eBca e K data and ,enera(content di &(ay. 9o be,in ' ith data iteration (et' u e a(i t o+ car to di &(ay ' ith carou e(.

```
public class Car {  
  
    private String model;  
    private int year;  
    private String manufacturer;  
    private String color;  
    ...  
}
```

```
public class CarBean {  
  
    private List<Car> cars;  
  
    public CarListController() {  
        cars = new ArrayList<Car>();  
        cars.add(new Car("myModel", 2005, "ManufacturerX", "blue"));  
        //add more cars  
    }  
  
    //getter setter  
}
```

```
<p:carousel value="#{carBean.cars}" var="car" itemStyle="width:200px">
    <p:graphicImage value="/images/cars/#{car.manufacturer}.jpg"/>
    <h:outputText value="Model: #{car.model}" />
    <h:outputText value="Year: #{car.year}" />
    <h:outputText value="Color: #{car.color}" />
</p:carousel>
```

*arou e(iterate throu,h the car co((ection and render it' chi(dren +or each car). note that you a(o need to de+ine a ' idth +or each ite " .

Limiting Visible Items

1 u de+au(t carou e((i t it ite " in &a,e ' ith i=e). 9hi i cu to " i=ab(e ' ith the ro ' attribute.

```
<p:carousel value="#{carBean.cars}" var="car" numVisible="1"
    itemStyle="width:200px" >
    ...
</p:carousel>
```



Effects

!a,in, ha&&en ' ith a (ider e+ect by de+au(t and +o((o ' in, ea in, o&tion are u&&orted.

- ↳ bac31oth
- ↳ bac3In
- ↳ bac37ut
- ↳ bounce1oth
- ↳ bounceln
- ↳ bounce7ut
- ↳ ea e1oth
- ↳ ea e1othStron,
- ↳ ea eln
- ↳ ea elnStron,
- ↳ ea e<one
- ↳ ea e7ut
- ↳ ea eln7ut*irc
- ↳ ea e7utStron,
- ↳ e(a tic1oth
- ↳ e(a ticln
- ↳ e(a tic7ut

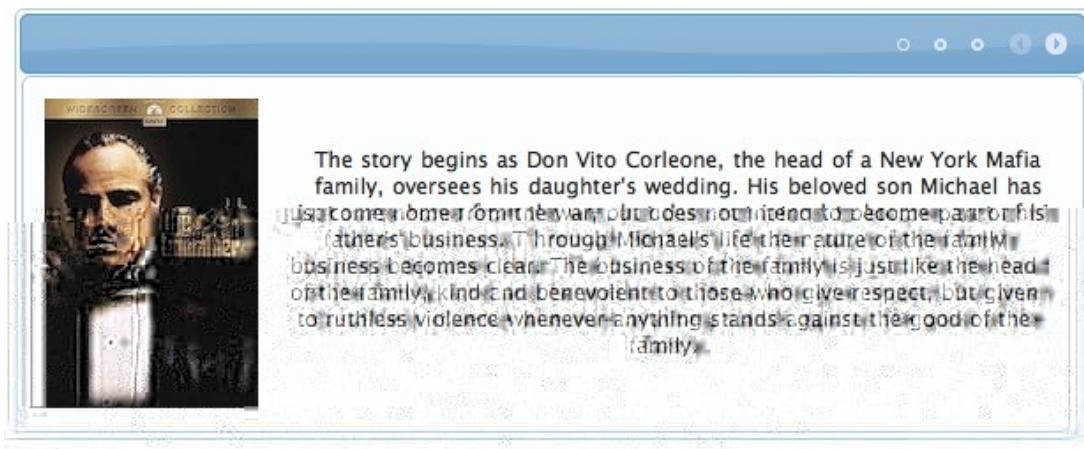
SlideShow

*arou e(can di &(ay the content in a (ide ho 'E +or thi &ur&o e *autoPlayInterval* and *circular* attribute are u ed. #o(o ' in, carou e(di &(ay a co((ection o+ i " a,e a a (ide ho '.

```
<p:carousel autoPlayInterval="2000" rows="1" effect="easeInStrong" circular="true"
    itemStyle="width:200px">
    <p:graphicImage value="/images/nature1.jpg"/>
    <p:graphicImage value="/images/nature2.jpg"/>
    <p:graphicImage value="/images/nature3.jpg"/>
    <p:graphicImage value="/images/nature4.jpg"/>
</p:carousel>
```

Content Display

Another u e ca e o+ carou e(i tab ba ed content di &(ay.



```
<p:carousel rows="1" itemStyle="height:200px;width:600px;">
    <p:tab title="Godfather Part I">
        <h:panelGrid columns="2" cellpadding="10">
            <p:graphicImage value="/images/godfather/godfather1.jpg" />
            <h:outputText value="The story begins as Don Vito ..." />
        </h:panelGrid>
    </p:tab>
    <p:tab title="Godfather Part II">
        <h:panelGrid columns="2" cellpadding="10">
            <p:graphicImage value="/images/godfather/godfather2.jpg" />
            <h:outputText value="Francis Ford Coppola's ..." />
        </h:panelGrid>
    </p:tab>
    <p:tab title="Godfather Part III">
        <h:panelGrid columns="2" cellpadding="10">
            <p:graphicImage value="/images/godfather/godfather3.jpg" />
            <h:outputText value="After a break of ..." />
        </h:panelGrid>
    </p:tab>
</p:carousel>
```

Item Selection

Show how to select an item from a carousel and display its details within a dialog.

```
<h:form id="form">
    <p:carousel value="#{carBean.cars}" var="car" itemStyle="width:200px">
        <p:graphicImage value="/images/cars/#{car.manufacturer}.jpg"/>
        <p:commandLink update=":form:detail" oncomplete="PF('dlg').show()">
            <h:outputText value="Model: #{car.model}" />
            <f:setPropertyActionListener value="#{car}" target="#{carBean.selected}" />
        </p:commandLink>
    </p:carousel>

    <p:dialog widgetVar="dlg">
        <h:outputText id="detail" value="#{carBean.selected}" />
    </p:dialog>
</h:form>
```

```
public class CarBean {
    private List<Car> cars;
    private Car selected;
    //getters and setters
}
```

Header and Footer

The header and footer of a carousel can be defined in two ways either by using the `headerText` and `footerText` options or by defining them in the `header` and `footer` facets that can take any content.

Client Side API

Widet `PrimeFaces.widget.Carousel`

Method	Params	Return Type	Description
<code>next()</code>	B	void	Displays the next item.
<code>prev()</code>	B	void	Displays the previous item.
<code>setIndex(int index)</code>	index	void	Displays the item at the specified index.
<code>startAutoPlay()</code>	B	void	Starts automatic item navigation.
<code>stopAutoPlay()</code>	B	void	Stops automatic item navigation.

Skinning

*arou e(re ide in a container e(e " ent ' hich *style* and *styleClass* o&&(y. *itemStyle* and *itemStyleClass* attribute a&&(y to each ite " di &(ayed by carou e(. #o((o ' in, i the (i t o+ structura(ty(e c(a e K

Style Class	Applies
.uiBcarou e(8 ain container
.uiBcarou e(Bheader	. eader container
.uiBcarou e(BheaderBtit(e	. eader content
.uiBcarou e(Bvie ' &ort	*ontent container
.uiBcarou e(Bbutton	<avi ,ation button
.uiBcarou e(Bne0tBbutton	<e0t navi ,ation button o+ &a ,inator
.uiBcarou e(B&revBbutton	!rev navi ,ation button o+ &a ,inator
.uiBcarou e(B&a,eB(in3	!a,e (in3 o+ &a ,inator.
.uiBcarou e(B&a,eB(in3	Each &a,e (in3 o+ &a ,inator.
.uiBcarou e(Bite "	Each ite " .

A 3innin , ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.12 CellEditor

*e((Editor i a he(&er co " &onent o+ databab(e u ed +or ince((editin,..

Info

9a,	cellEditor
*o " &onent *(a	org.primefaces.component.celleditor.CellEditor
*o " &onent 9y&e	org.primefaces.component.CellEditor
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.CellEditorRenderer
Renderer *(a	org.primefaces.component.celleditor.CellEditorRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	The value to identify the rendering, or the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.

Getting Started with CellEditor

See introduction in the documentation or more information about usage.

3.13 Chart

*chart co " &onent i a ,eneric ,ra&h co " &onent to create variou ty&e o+ chart u in, /C&(ot library. Each chart ty&e ha it o 'n ub ection ' ith code e0a " &(e and ection).1%.10 docu " ent the +u((chartin, A !).

Info

9a,	chart
*o " &onent *(a	org.primefaces.component.chart.Chart
*o " &onent 9y&e	org.primefaces.component.Chart
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ChartRenderer
Renderer *(a	org.primefaces.component.chart.ChartRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier o+ the co " &onent
rendered	true	Boolean	Boolean value to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	null	Object	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
ty&e	null	String,	9y&e o+ the chart.
" ode(null	*chart 8 ode(8 ode(ob/ect o+ data and ettin, .
ty(e	null	String,	In(ine ty(e o+ the co " &onent.
ty(e*(a	null	String,	Sty(e c(a o+ the co " &onent.
' id,et>ar	null	String,	<a " e o+ the c(ient ide ' id,et.

3.13.1 PieChart

A pie chart is created with the following code.

Basic

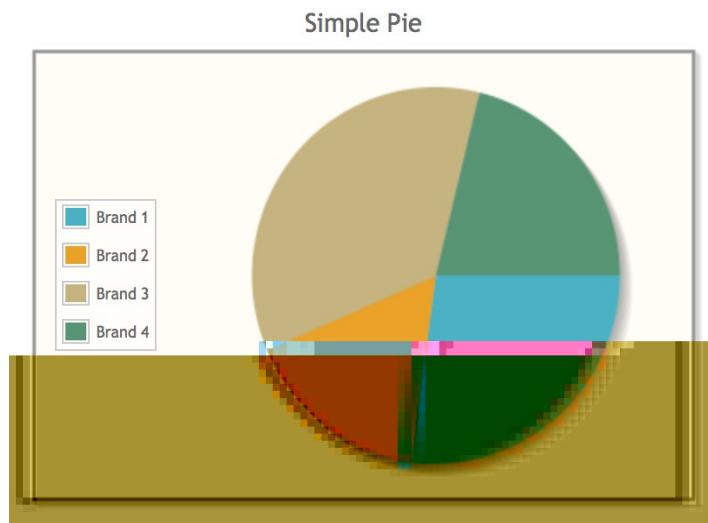
```
<p:chart type="pie" value="#{bean.model}" />
```

```
public class Bean {

    private PieChartModel model;

    public Bean() {
        model = new PieChartModel();
        model.set("Brand 1", 540);
        model.set("Brand 2", 325);
        model.set("Brand 3", 702);
        model.set("Brand 4", 421);
        model.setTitle("Simple Pie");
        model.setLegendPosition("w");
    }

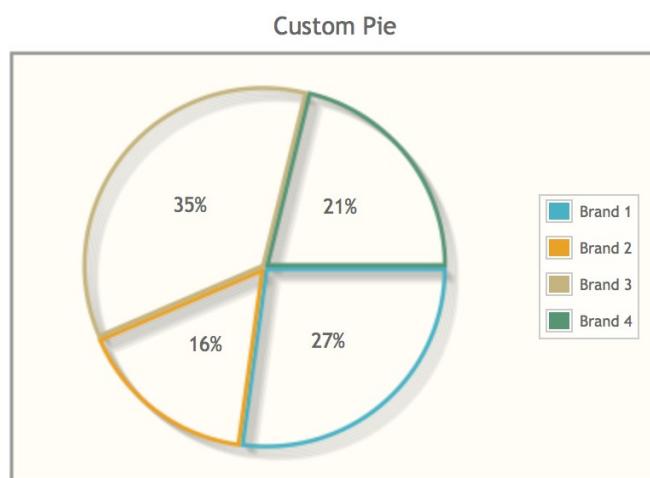
    public PieChartModel getModel() {
        return model;
    }
}
```



Customized

```
<p:chart type="pie" value="#{bean.model}" />
```

```
public class Bean {  
  
    private PieChartModel model;  
  
    public Bean() {  
        model = new PieChartModel();  
        model.set("Brand 1", 540);  
        model.set("Brand 2", 325);  
        model.set("Brand 3", 702);  
        model.set("Brand 4", 421);  
  
        model.setTitle("Custom Pie");  
        model.setLegendPosition("e");  
        model.setFill(false);  
        model.setShowDataLabels(true);  
        model.setDiameter(150);  
    }  
  
    public PieChartModel getModel() {  
        return model;  
    }  
}
```



3.13.2 LineChart

Line chart is used to create a line chart.

Basic

```
<p:chart type="line" value="#{bean.model}" />
```

```
public class Bean {

    private LineChartModel model;

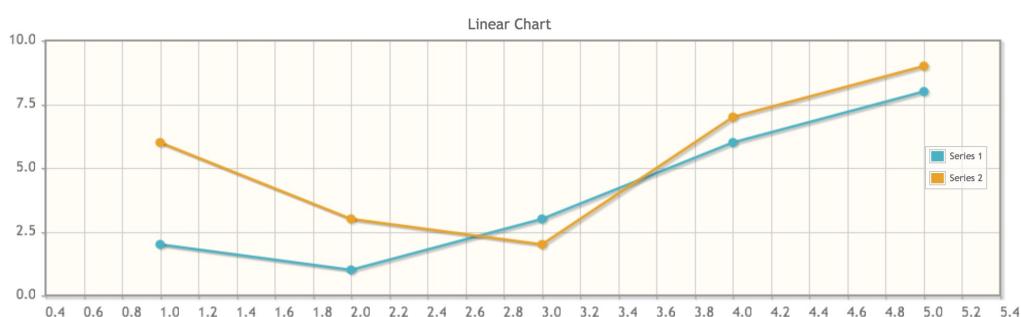
    public Bean() {
        model = new LineChartModel();

        LineChartSeries series1 = new LineChartSeries();
        series1.setLabel("Series 1");
        series1.set(1, 2);
        series1.set(2, 1);
        series1.set(3, 3);
        series1.set(4, 6);
        series1.set(5, 8);

        LineChartSeries series2 = new LineChartSeries();
        series2.setLabel("Series 2");
        series2.set(1, 6);
        series2.set(2, 3);
        series2.set(3, 2);
        series2.set(4, 7);
        series2.set(5, 9);

        model.addSeries(series1);
        model.addSeries(series2);
        model.setTitle("Linear Chart");
        model.setLegendPosition("e");
        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setMin(0);
        yAxis.setMax(10);
    }

    public LineChartModel getModel() {
        return model;
    }
}
```



Customized

```
<p:chart type="line" value="#{bean.model}" />
```

```
public class Bean {

    private LineChartModel model;

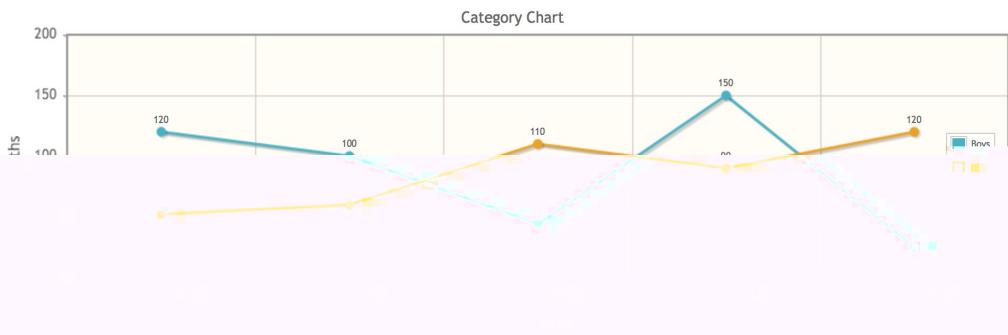
    public Bean() {
        model = new LineChartModel();

        ChartSeries boys = new ChartSeries();
        boys.setLabel("Boys");
        boys.set("2004", 120);
        boys.set("2005", 100);
        boys.set("2006", 44);
        boys.set("2007", 150);
        boys.set("2008", 25);

        ChartSeries girls = new ChartSeries();
        girls.setLabel("Girls");
        girls.set("2004", 52);
        girls.set("2005", 60);
        girls.set("2006", 110);
        girls.set("2007", 90);
        girls.set("2008", 120);

        model.addSeries(boys);
        model.addSeries(girls);
        model.setTitle("Category Chart");
        model.setLegendPosition("e");
        model.setShowPointLabels(true);
        model.getAxes().put(AxisType.X, new CategoryAxis("Years"));
        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setLabel("Births");
        yAxis.setMin(0);
        yAxis.setMax(200);
    }

    public LineChartModel getModel() {
        return model;
    }
}
```



Area

```
<p:chart type="line" value="#{bean.model}" />
```

```
public class Bean {

    private LineChartModel model;

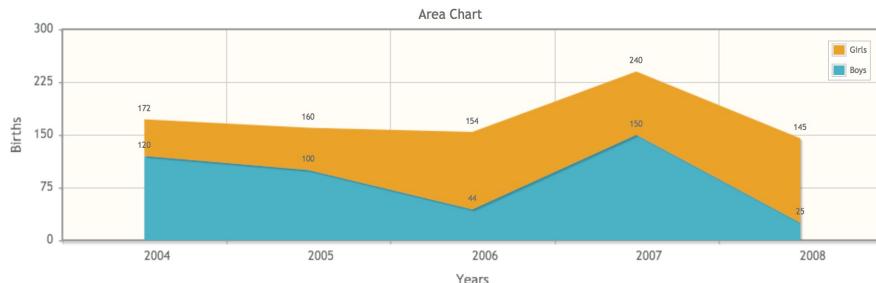
    public Bean() {
        model = new LineChartModel();

        LineChartSeries boys = new LineChartSeries();
        boys.setFill(true);
        boys.setLabel("Boys");
        boys.set("2004", 120);
        boys.set("2005", 100);
        boys.set("2006", 44);
        boys.set("2007", 150);
        boys.set("2008", 25);

        LineChartSeries girls = new LineChartSeries();
        girls.setFill(true);
        girls.setLabel("Girls");
        girls.set("2004", 52);
        girls.set("2005", 60);
        girls.set("2006", 110);
        girls.set("2007", 90);
        girls.set("2008", 120);

        model.addSeries(boys);
        model.addSeries(girls);
        model.setTitle("Area Chart");
        model.setLegendPosition("ne");
        model.setStacked(true);
        model.setShowPointLabels(true);
        model.getAxis(AxisType.X).setLabel("Years");
        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setLabel("Births");
        yAxis.setMin(0);
        yAxis.setMax(300);
    }

    public CartesianChartModel getModel() {
        return model;
    }
}
```



3.13.3 BarChart

1 ar*hart 8 ode(i u ed to created a 1 ar*hart.

Basic

```
<p:chart type="bar" value="#{bean.model}" />
```

```
public class Bean {
    private BarChartModel model;

    public ChartBean() {
        model = new BarChartModel();

        ChartSeries boys = new ChartSeries();
        boys.setLabel("Boys");
        boys.set("2004", 120);
        boys.set("2005", 100);
        boys.set("2006", 44);
        boys.set("2007", 150);
        boys.set("2008", 25);

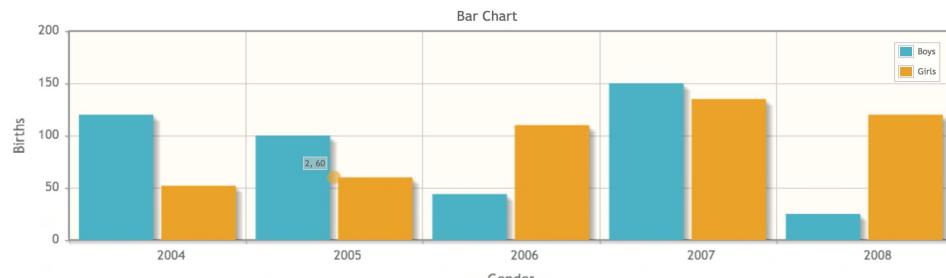
        ChartSeries girls = new ChartSeries();
        girls.setLabel("Girls");
        girls.set("2004", 52);
        girls.set("2005", 60);
        girls.set("2006", 110);
        girls.set("2007", 135);
        girls.set("2008", 120);

        model.addSeries(boys);
        model.addSeries(girls);
        model.setTitle("Bar Chart");
        model.setLegendPosition("ne");

        Axis xAxis = model.getAxis(AxisType.X);
        xAxis.setLabel("Gender");

        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setLabel("Births");
        yAxis.setMin(0);
        yAxis.setMax(200);
    }

    public BarChartModel getModel() { return model; }
}
```



Horizontal and Stacked

```
<p:chart type="bar" value="#{bean.model}" />
```

```
public class Bean {
    private HorizontalBarChartModel model;

    public ChartBean() {
        model = new HorizontalBarChartModel();

        ChartSeries boys = new ChartSeries();
        boys.setLabel("Boys");
        boys.set("2004", 50);
        boys.set("2005", 96);
        boys.set("2006", 44);
        boys.set("2007", 55);
        boys.set("2008", 25);

        ChartSeries girls = new ChartSeries();
        girls.setLabel("Girls");
        girls.set("2004", 52);
        girls.set("2005", 60);
        girls.set("2006", 82);
        girls.set("2007", 35);
        girls.set("2008", 120);

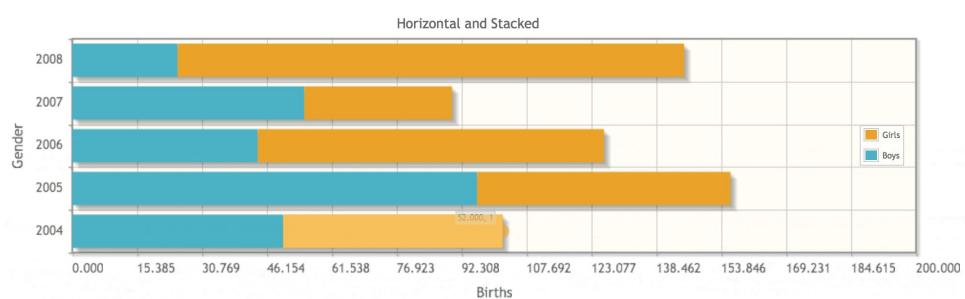
        model.addSeries(boys);
        model.addSeries(girls);

        model.setTitle("Horizontal and Stacked");
        model.setLegendPosition("e");
        model.setStacked(true);

        Axis xAxis = model.getAxis(AxisType.X);
        xAxis.setLabel("Births");
        xAxis.setMin(0);
        xAxis.setMax(200);

        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setLabel("Gender");
    }

    public HorizontalBarChartModel getModel() { return model; }
}
```



3.13.4 DonutChart

Donut*hart i ,enerated u in, Donut*hart 8 ode(.

Basic

```
<p:chart type="donut" value="#{bean.model}" />
```

```
public class Bean {
    private DonutChartModel model;

    public ChartBean() {
        model = new DonutChartModel();

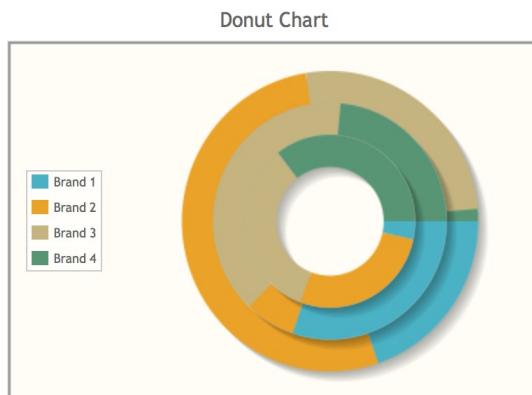
        Map<String, Number> circle1 = new LinkedHashMap<String, Number>();
        circle1.put("Brand 1", 150);
        circle1.put("Brand 2", 400);
        circle1.put("Brand 3", 200);
        circle1.put("Brand 4", 10);
        model.addCircle(circle1);

        Map<String, Number> circle2 = new LinkedHashMap<String, Number>();
        circle2.put("Brand 1", 540);
        circle2.put("Brand 2", 125);
        circle2.put("Brand 3", 702);
        circle2.put("Brand 4", 421);
        model.addCircle(circle2);

        Map<String, Number> circle3 = new LinkedHashMap<String, Number>();
        circle3.put("Brand 1", 40);
        circle3.put("Brand 2", 325);
        circle3.put("Brand 3", 402);
        circle3.put("Brand 4", 421);
        model.addCircle(circle3);

        model.setTitle("Donut Chart");
        model.setLegendPosition("w");
    }

    public DonutChartModel getModel() { return model; }
}
```



Customized

```
<p:chart type="donut" value="#{bean.model}" />
```

```
public class Bean {
    private DonutChartModel model;

    public ChartBean() {
        model = new DonutChartModel();

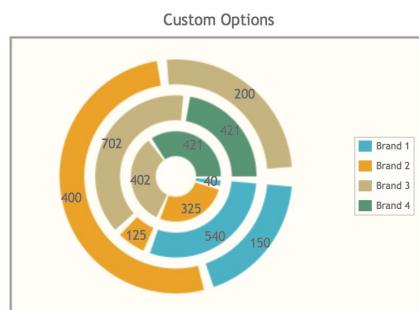
        Map<String, Number> circle1 = new LinkedHashMap<String, Number>();
        circle1.put("Brand 1", 150);
        circle1.put("Brand 2", 400);
        circle1.put("Brand 3", 200);
        circle1.put("Brand 4", 10);
        model.addCircle(circle1);

        Map<String, Number> circle2 = new LinkedHashMap<String, Number>();
        circle2.put("Brand 1", 540);
        circle2.put("Brand 2", 125);
        circle2.put("Brand 3", 702);
        circle2.put("Brand 4", 421);
        model.addCircle(circle2);

        Map<String, Number> circle3 = new LinkedHashMap<String, Number>();
        circle3.put("Brand 1", 40);
        circle3.put("Brand 2", 325);
        circle3.put("Brand 3", 402);
        circle3.put("Brand 4", 421);
        model.addCircle(circle3);

        model.setTitle("Donut Chart");
        model.setLegendPosition("w");
        model.setTitle("Custom Options");
        model.setLegendPosition("e");
        model.setSliceMargin(5);
        model.setShowDataLabels(true);
        model.setDataFormat("value");
        model.setShadow(false);
    }

    public DonutChartModel getModel() { return model; }
}
```



3.13.5 BubbleChart

A bubble chart is created with a chart model.

Basic

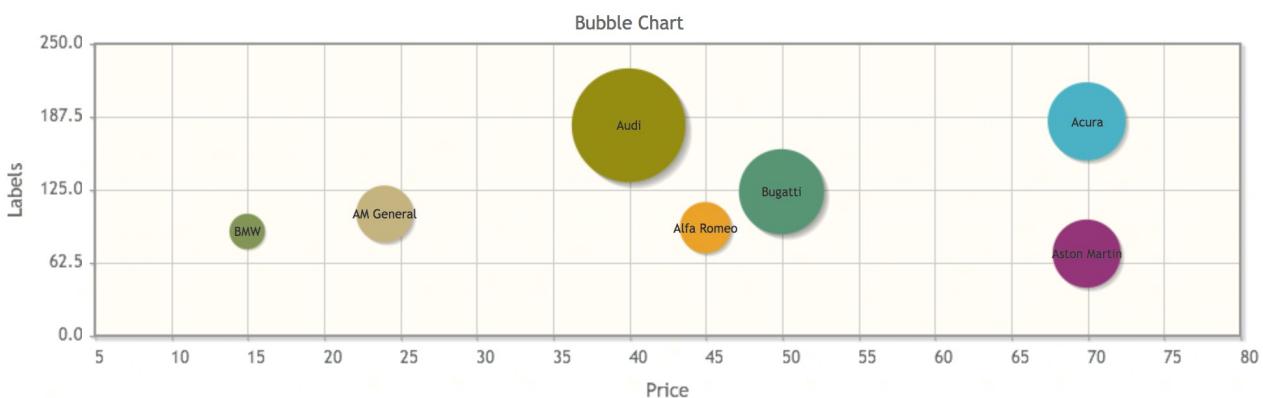
```
<p:chart type="bubble" value="#{bean.model}" />
```

```
public class Bean {
    private BubbleChartModel model;

    public ChartBean() {
        model = new BubbleChartModel();
        model.add(new BubbleChartSeries("Acura", 70, 183, 55));
        model.add(new BubbleChartSeries("Alfa Romeo", 45, 92, 36));
        model.add(new BubbleChartSeries("AM General", 24, 104, 40));
        model.add(new BubbleChartSeries("Bugatti", 50, 123, 60));
        model.add(new BubbleChartSeries("BMW", 15, 89, 25));
        model.add(new BubbleChartSeries("Audi", 40, 180, 80));
        model.add(new BubbleChartSeries("Aston Martin", 70, 70, 48));

        model.setTitle("Bubble Chart");
        model.getAxis(AxisType.X).setLabel("Price");
        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setMin(0);
        yAxis.setMax(250);
        yAxis.setLabel("Labels");
    }

    public BubbleChartModel getModel() { return model; }
}
```



Customized

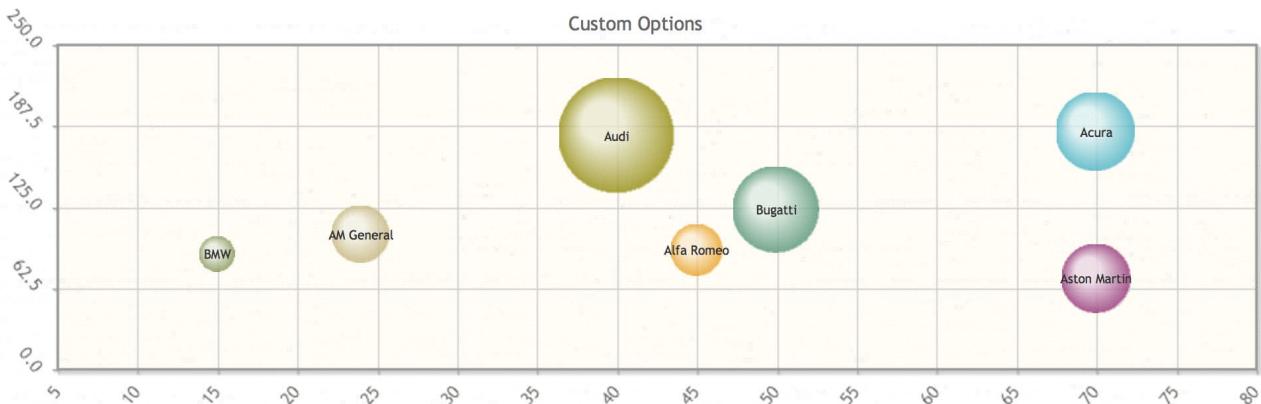
```
<p:chart type="bubble" value="#{bean.model}" />
```

```
public class Bean {
    private BubbleChartModel model;

    public ChartBean() {
        model = new BubbleChartModel();
        model.add(new BubbleChartSeries("Acura", 70, 183, 55));
        model.add(new BubbleChartSeries("Alfa Romeo", 45, 92, 36));
        model.add(new BubbleChartSeries("AM General", 24, 104, 40));
        model.add(new BubbleChartSeries("Bugatti", 50, 123, 60));
        model.add(new BubbleChartSeries("BMW", 15, 89, 25));
        model.add(new BubbleChartSeries("Audi", 40, 180, 80));
        model.add(new BubbleChartSeries("Aston Martin", 70, 70, 48));

        model = initBubbleModel();
        model.setTitle("Custom Options");
        model.setShadow(false);
        model.setBubbleGradients(true);
        model.setBubbleAlpha(0.8);
        model.getAxis(AxisType.X).setTickAngle(-50);
        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setMin(0);
        yAxis.setMax(250);
        yAxis.setTickAngle(50);
    }

    public BubbleChartModel getModel() { return model; }
}
```



3.13.6 Ohlc Chart

A chart type used to display OHLC data.

Basic

```
<p:chart type="ohlc" value="#{bean.model}" />
```

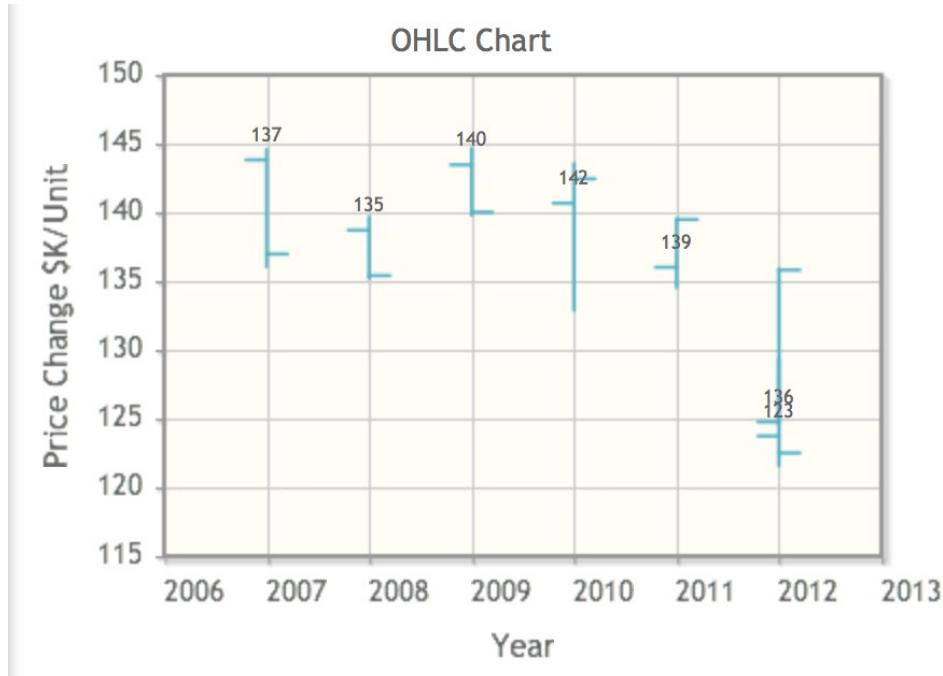
```
public class Bean {
    private OhlcChartModel model;

    public ChartBean() {
        ohlcModel = new OhlcChartModel();

        ohlcModel.add(new OhlcChartSeries(2007, 143.82, 144.56, 136.04, 136.97));
        ohlcModel.add(new OhlcChartSeries(2008, 138.7, 139.68, 135.18, 135.4));
        ohlcModel.add(new OhlcChartSeries(2009, 143.46, 144.66, 139.79, 140.02));
        ohlcModel.add(new OhlcChartSeries(2010, 140.67, 143.56, 132.88, 142.44));
        ohlcModel.add(new OhlcChartSeries(2011, 136.01, 139.5, 134.53, 139.48));
        ohlcModel.add(new OhlcChartSeries(2012, 124.76, 135.9, 124.55, 135.81));
        ohlcModel.add(new OhlcChartSeries(2012, 123.73, 129.31, 121.57, 122.5));
    }

    ohlcModel.setTitle("OHLC Chart");
    ohlcModel.getAxis(AxisType.X).setLabel("Year");
    ohlcModel.getAxis(AxisType.Y).setLabel("Price Change $K/Unit");
}

public OhlcChartModel getModel() { return model; }
}
```



Candlestick

```
<p:chart type="ohlc" value="#{bean.model}" />
```

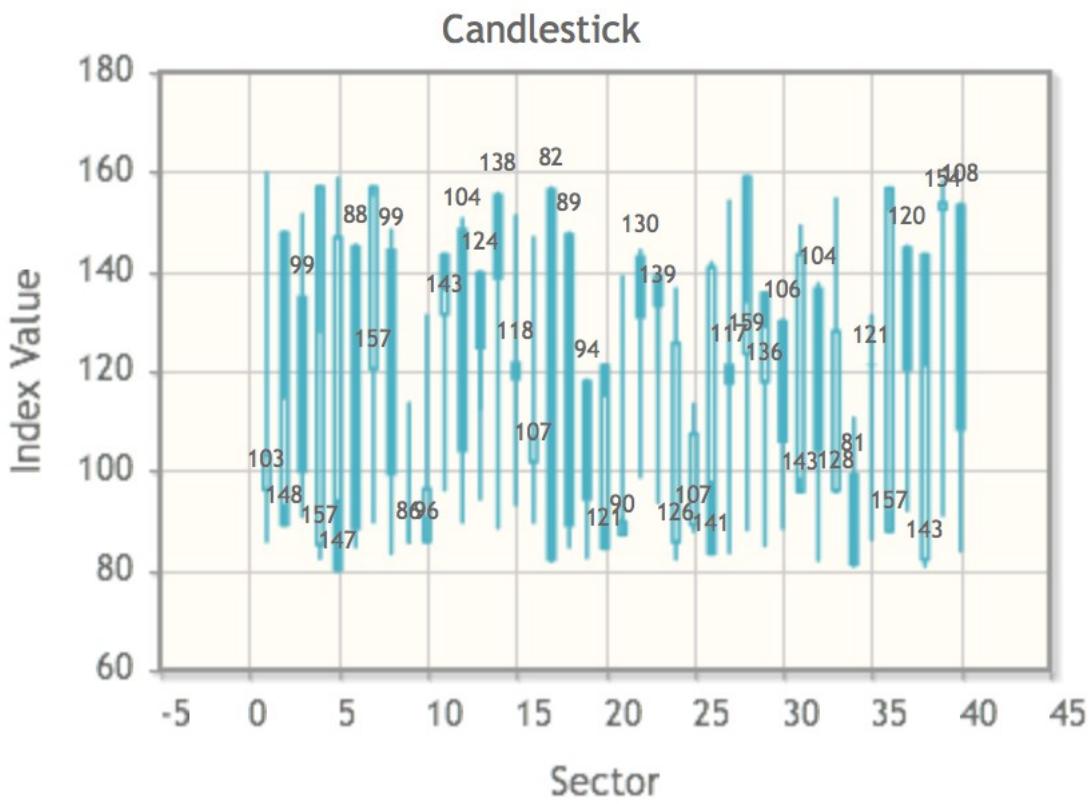
```
public class Bean {
    private OhlcChartModel model;

    public ChartBean() {
        model = new OhlcChartModel();

        for( int i=1 ; i < 41 ; i++) {
            ohlcModel2.add(new OhlcChartSeries(i, Math.random() * 80 + 80,
                Math.random() * 50 + 110, Math.random() * 20 + 80, Math.random() * 80 + 80));
        }

        model.setTitle("Candlestick");
        model.setCandleStick(true);
        model.getAxis(AxisType.X).setLabel("Sector");
        model.getAxis(AxisType.Y).setLabel("Index Value");
    }

    public OhlcChartModel getModel() { return model; }
}
```



3.13.7 MeterGauge Chart

8 eterGau,e *hart i created u in , 8 eterGau,e*hart 8 ode(.

Basic

```
<p:chart type="metergauge" value="#{bean.model}" />
```

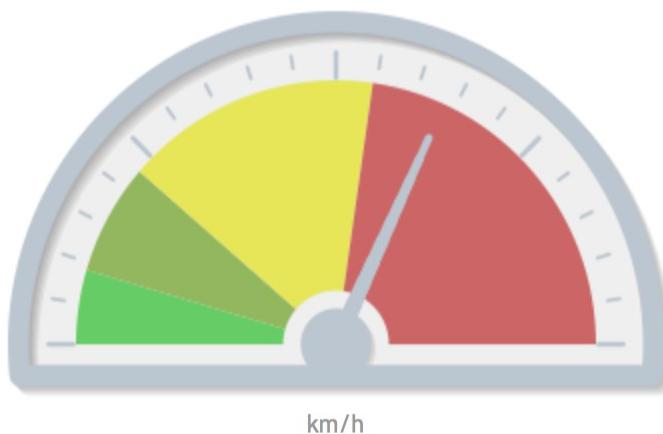
```
public class Bean {
    private MeterGaugeChartModel model;

    public ChartBean() {
        List<Number> intervals = new ArrayList<Number>(){
            add(20);
            add(50);
            add(120);
            add(220);
        };

        model = new MeterGaugeChartModel(140, intervals);
        model.setTitle("MeterGauge Chart");
        model.setGaugeLabel("km/h");
    }

    public MeterGaugeChartModel getModel() { return model; }
}
```

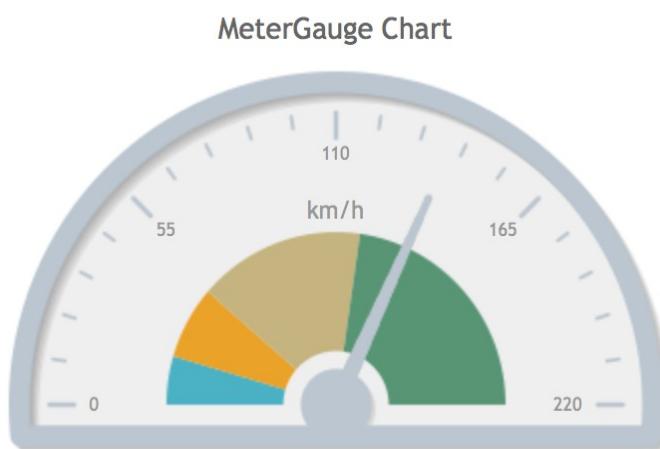
Custom Options



Customized

```
<p:chart type="metergauge" value="#{bean.model}" />
```

```
public class Bean {  
    private MeterGaugeChartModel model;  
  
    public ChartBean() {  
        List<Number> intervals = new ArrayList<Number>(){  
            add(20);  
            add(50);  
            add(120);  
            add(220);  
        };  
  
        model = new MeterGaugeChartModel(140, intervals);  
        model.setTitle("Custom Options");  
        model.setSeriesColors("66cc66,93b75f,E7E658,cc6666");  
        model.setGaugeLabel("km/h");  
        model.setGaugeLabelPosition("bottom");  
        model.setShowTickLabels(false);  
        model.setLabelHeightAdjust(110);  
        model.setIntervalOuterRadius(130);  
    }  
  
    public MeterGaugeChartModel getModel() { return model; }  
}
```



3.13.8 Combined Chart

In a chart, different series types can be displayed together.

Basic

```
<p:chart type="bar" value="#{bean.model}" />
```

```
public class Bean {
    private BarChartModel model;

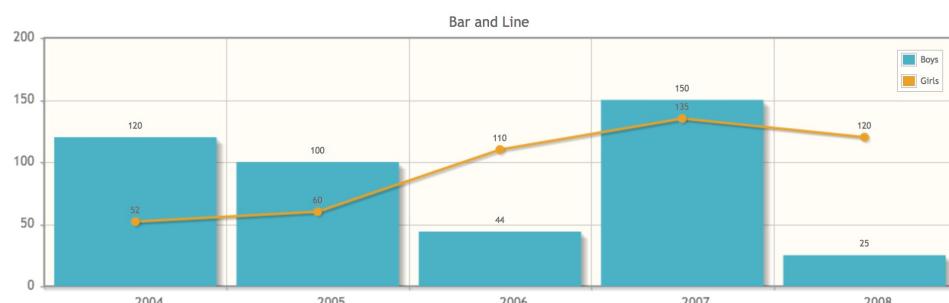
    public ChartBean() {
        combinedModel = new BarChartModel();

        BarChartSeries boys = new BarChartSeries();
        boys.setLabel("Boys");
        boys.set("2004", 120);
        boys.set("2005", 100);
        boys.set("2006", 44);
        boys.set("2007", 150);
        boys.set("2008", 25);

        LineChartSeries girls = new LineChartSeries();
        girls.setLabel("Girls");
        girls.set("2004", 52);
        girls.set("2005", 60);
        girls.set("2006", 110);
        girls.set("2007", 135);
        girls.set("2008", 120);

        model.addSeries(boys);
        model.addSeries(girls);
        model.setTitle("Bar and Line");
        model.setLegendPosition("ne");
        model.setMouseoverHighlight(false);
        model.setShowDatatip(false);
        model.setShowPointLabels(true);
        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setMin(0);
        yAxis.setMax(200);
    }

    public BarChartModel getModel() { return model; }
}
```



3.13.9 Multiple Axis

U& to 9 a0e ?0a0i B09a0i E ya0i By9a0i @ can be di &(ayed on the a " e chart.

Basic

```
<p:chart type="line" value="#{bean.model}" />
```

```
public class Bean {
    private LineChartModel model;

    public ChartBean() {
        model = new LineChartModel();

        BarChartSeries boys = new BarChartSeries();
        boys.setLabel("Boys");
        boys.set("2004", 120);
        boys.set("2005", 100);
        boys.set("2006", 44);
        boys.set("2007", 150);
        boys.set("2008", 25);

        LineChartSeries girls = new LineChartSeries();
        girls.setLabel("Girls");
        girls.setXaxis(AxisType.X2);
        girls.setYaxis(AxisType.Y2);
        girls.set("A", 52);
        girls.set("B", 60);
        girls.set("C", 110);
        girls.set("D", 135);
        girls.set("E", 120);

        model.addSeries(boys);
        model.addSeries(girls);

        model.setTitle("Multi Axis Chart");
        model.setMouseoverHighlight(false);

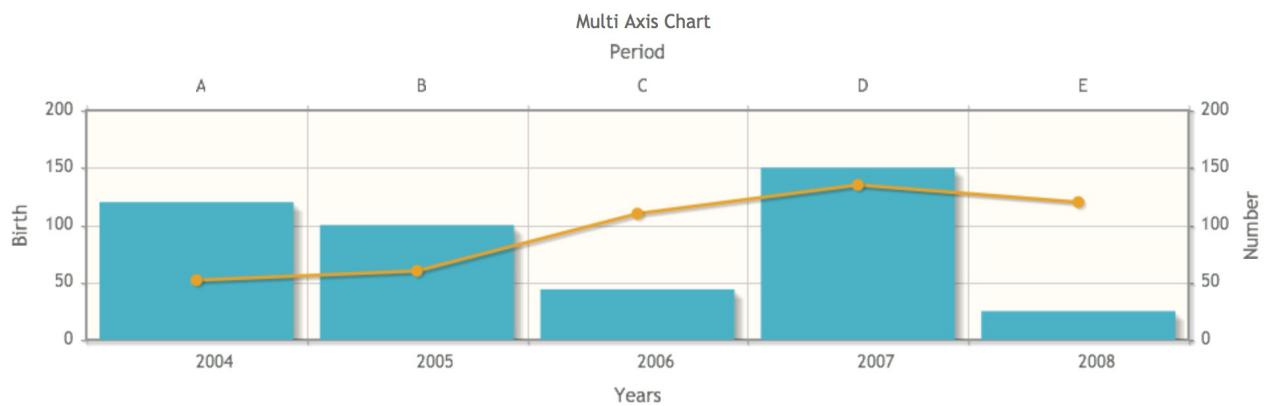
        model.getAxes().put(AxisType.X, new CategoryAxis("Years"));
        model.getAxes().put(AxisType.X2, new CategoryAxis("Period"));

        Axis yAxis = model.getAxis(AxisType.Y);
        yAxis.setLabel("Birth");
        yAxis.setMin(0);
        yAxis.setMax(200);

        Axis y2Axis = new LinearAxis("Number");
        y2Axis.setMin(0);
        y2Axis.setMax(200);

        model.getAxes().put(AxisType.Y2, y2Axis);
    }

    public LineChartModel getModel() { return model; }
}
```



3.13.10 Date Axis

Use DateAxis if you are displaying date in an axis.

Basic

```
<p:chart type="line" value="#{bean.model}" />
```

```
public class Bean {
    private LineChartModel model;

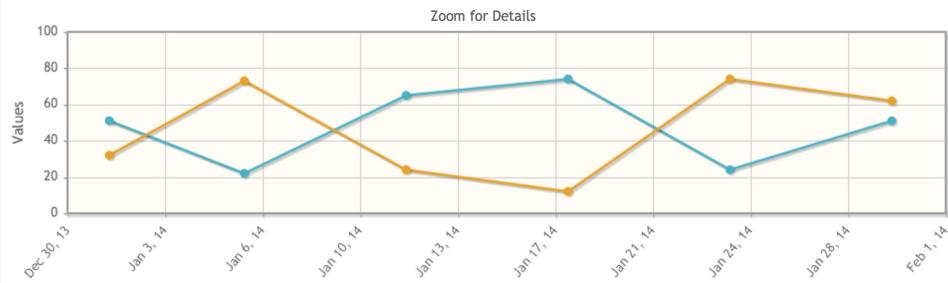
    public ChartBean() {
        dateModel = new LineChartModel();
        LineChartSeries series1 = new LineChartSeries();
        series1.setLabel("Series 1");
        series1.set("2014-01-01", 51);
        series1.set("2014-01-06", 22);
        series1.set("2014-01-12", 65);
        series1.set("2014-01-18", 74);
        series1.set("2014-01-24", 24);
        series1.set("2014-01-30", 51);

        LineChartSeries series2 = new LineChartSeries();
        series2.setLabel("Series 2");
        series2.set("2014-01-01", 32);
        series2.set("2014-01-06", 73);
        series2.set("2014-01-12", 24);
        series2.set("2014-01-18", 12);
        series2.set("2014-01-24", 74);
        series2.set("2014-01-30", 62);

        dateModel.addSeries(series1);
        dateModel.addSeries(series2);
        dateModel.setTitle("Zoom for Details");
        dateModel.setZoom(true);
        dateModel.getAxis(AxisType.Y).setLabel("Values");
        DateAxis axis = new DateAxis("Dates");
        axis.setTickAngle(-50);
        axis.setMax("2014-02-01");
        axis.setTickFormat("%b %#d, %y");

        dateModel.getAxes().put(AxisType.X, axis);
    }

    public LineChartModel getModel() { return model; }
}
```



3.13.11 Interactive Chart

*hart are interactive co " &onent E in+or " ation about e(ected erie and ite " can be &a ed via a/a0 to a AS# bac3in, bean u in, Itc " Se(ectEvent).

Basic

```
<p:chart type="pie" value="#{bean.model}">
    <p:ajax event="itemSelect" listener="#{bean.itemSelect}" />
</p:chart>
```

```
public class Bean {
    private PieChartModel model;

    public ChartBean() {
        model = new PieChartModel();

        model.set("Brand 1", 540);
        model.set("Brand 2", 325);
        model.set("Brand 3", 702);
        model.set("Brand 4", 421);

        model.setTitle("Simple Pie");
        model.setLegendPosition("w");
    }

    public PieChartModel getModel() { return model; }

    public void itemSelect(ItemSelectEvent event) {
        FacesMessage msg = new FacesMessage(FacesMessage.SEVERITY_INFO,
            "Item selected", "Item Index: " + event.getItemIndex() +
            ", Series Index:" + event.getSeriesIndex());

        FacesContext.getCurrentInstance().addMessage(null, msg);
    }
}
```

3.13.12 Export

*hart co " &onent &rovide a c(ient ide " ethod to convert the ,ra&h to an i " a,e. E0a " &(e be(o ' de " on trate ho ' to u e a button c(ic3 to e0&ort the chart a an i " a,e and di &(ay it in a dia(o, o that u er can do ' n(load it a a re,u(ar i " a,e.

```
<p:chart type="line" value="#{bean.model}" style="width:500px;height:300px"
           widgetVar="chart"/>

<p:commandButton type="button" value="Export" icon="ui-icon-extlink"
                  onclick="exportChart()"/>

<p:dialog widgetVar="dlg" showEffect="fade" modal="true" header="Chart as an Image">
    <p:outputPanel id="output" layout="block" style="width:500px;height:300px"/>
</p:dialog>

function exportChart() {
    //export image
    $('#output').empty().append(PF('chart').exportAsImage());

    //show the dialog
    PF('dlg').show();
}
```

3.13.13 Static Images

A#ree*hart ' ith Gra&hicl " a,e co " &onent i an a(ternative to the chart co " &onent.

Basic

```
<p:graphicImage value="#{bean.chart}" />
```

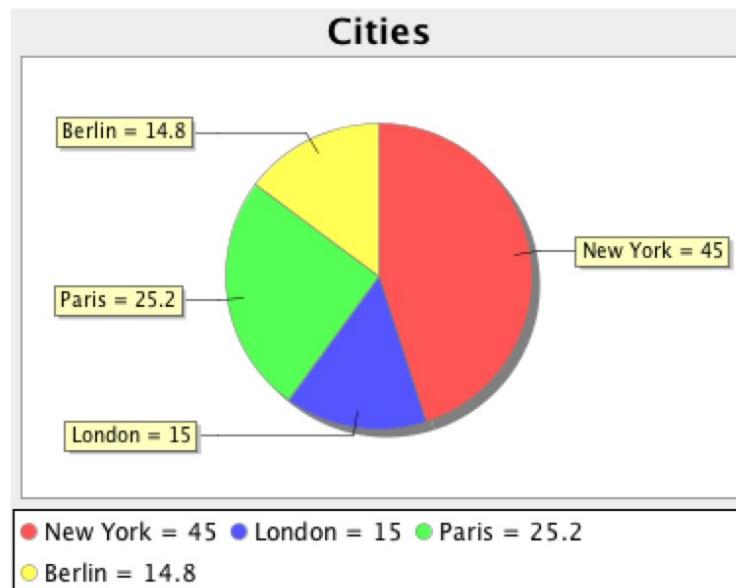
```
public class Bean {
    private StreamedContent chart;

    public Bean() {
        JFreeChart jfreechart = ChartFactory.createPieChart("Cities",
createDataset(), true, true, false);
        File chartFile = new File("dynamichart");
        ChartUtilities.saveChartAsPNG(chartFile, jfreechart, 375, 300);
        chart = new DefaultStreamedContent(new FileInputStream(chartFile),
"image/png");
    }

    public StreamedContent getChart() { return model; }

    private PieDataset createDataset() {
        DefaultPieDataset dataset = new DefaultPieDataset();
        dataset.setValue("New York", new Double(45.0));
        dataset.setValue("London", new Double(15.0));
        dataset.setValue("Paris", new Double(25.2));
        dataset.setValue("Berlin", new Double(14.8));

        return dataset;
    }
}
```



3.13.14 Skinning

*hart can be ty(ed u in, re, u(ar c . #o((o ' in, i the (i t o+ ty(e c(a e K

Style Class	Applies
./C&(otBtar, et	!(ot tar, et container.
./C&(otBa0i	A0e .
./C&(otBoa0i	!ri " ary OBa0i .
./C&(otBya0i	!ri " ary yBa0i .
./C&(otB0%a0i E ./C&(otB0)a0i ...	%ndE)rd ... OBa0i .
./C&(otBy%a0i E ./C&(otBy)a0i ...	%ndE)rd ... yBa0i .
./C&(otBa0i Btic3	A0i tic3 .
./C&(otBoa0i Btic3	!ri " ary OBa0i tic3 .
./C&(otB0%a0i Btic3	Secondary OBa0i tic3 .
./C&(otBya0i Btic3	!ri " ary yBa0i Btic3 .
./C&(otBy%a0i Btic3	Seconday yBa0i Btic3 .
tab(e./C&(otBtab(eB(e, end	6e, end tab(e.
./C&(otBtit(e	9it(e o+ the chart.
./C&(otBcur orBtoo(ti&	*ur or too(ti&.
./C&(otBhi, h(i, hterBtoo(ti&	. i, h(i, hter too(ti&.
div./C&(otBtab(eB(e, endB ' atch	*o(or ' atch o+ the (e, end.

Additiona(y style and styleClass o&tion o+ chart co " &onent a&&(y to the container e(e " ent o+ chart E u e the e attribute to &eci+y the di " en ion o+ a chart.

```
<p:pieChart value="#{bean.model}" style="width:320px;height:200px" />
```

In ca e you'd (i3e to chan,e the co(or o+ erie E u e the seriesColors o&tion in *hart 8 ode(A !I.

3.13.15 Extender

*hart A !I &rovide hi ,h (eve(acce to co " "on(y u ed /C&(ot o&tion ho ' ever there are " any " ore cu to " i=ation o&tion avai(ab(e in /C&(ot. E0tender +eature &rovide acce to (o ' (eve(a&i to do advanced cu to " i=ation by enhancin, the con+i,uration ob/ectE here i an e0a " &(e to increa e hado ' de&th o+ the (ine erie ' here " ode(0 e0tender &ro&erty i et to Ie0tI.

```
<p:chart type="line" value="#{bean.model}" />
```

```
function ext() {
    //this = chart widget instance
    //this.cfg = options
    this.cfg.seriesDefaults = {
        shadowDepth: 5
    };
}
```

Re+er to /C !(ot doc +or avai(ab(e o&tion .

3.13.16 Chart API

Axis

org.primefaces.model.chart.Axis

Property	Default	Type	Description
(abe(nu((String,	9it(e o+ the a0i .
" in	nu((7 b/ect	8 ini " u " boundary va(ue.
" a0	nu((7 b/ect	8 a0i " u " boundary va(ue.
tic3An,(e	nu((Inte ,er	An ,(e o+ teOtE " ea ured c(oc3 ' i e.
tic3#or " at	nu((String,	#or " at trin, to u e ' ith the a0i tic3 #or " after
tic3Interva(nu((String,	<u " ber o+ unit bet ' een tic3 .
tic3*ount	nu((Inte ,er	De ired nu " ber o+ tic3 .

AxisType

org.primefaces.model.chart.AxisType

A0i 9y&e i an enu " to de+ine the ty&e o+ the a0i +ro " FBM to F9BM9.

BarChartModel

org.primefaces.model.chart.BarChartModel e0tend *org.primefaces.model.chart.ChartModel*

Property	Default	Type	Description
bar !addin,	2	Inte ,er	!addin, bet ' een bar .
bar 8 ar,in	10	Inte ,er	8 ar,in bet ' een bar .
tac3ed	+a(e	1oo(ean	Di &(ay erie in tac3ed +or " at.

BarChartSeries

org.primefaces.model.chart.BarChartSeries e0tend *org.primefaces.model.chart.ChartSeries*

Property	Default	Type	Description
di ab(eStac3	+a(e	1oo(ean	When trueE erie data i not inc(uded in a tac3ed chart.

BubbleChartModel

org.primefaces.model.chart.BubbleChartModel e0tend *org.primefaces.model.chart.ChartModel*

Property	Default	Type	Description
data	nu((6i tS 1ubb(e*hartSerie T	Data a a (i t o+ 1ubb(e*hartSerie .
bubb(eGradient	+a(e	1oo(ean	Di &(ay bubb(e ' ith ,radient .
bubb(eA(&ha	1.0	Doub(e	7&acity o+ bubb(e .
ho ' 6abe(true	1oo(ean	Di &(ay (abe(o+ a erie in ide a bubb(e.

BubbleChartSeries*org.primefaces.model.chart.BubbleChartSeries**org.primefaces.model.chart.ChartSeries*

Property
x
y
radius
(label)

Property	Type	Description
x	Integer	Value of the bubble.
y	Integer	Value of the bubble.
radius	Integer	Radius of the bubble.
(label)	String	Label of the bubble.

CartesianChartModel*org.primefaces.model.chart.CartesianChartModel**org.primefaces.model.cartesianChartModel*

Property
erie
a0entG&PUS
10UWE5W-oa
ani " ate
ho ' Datati&
datati&#or " at
ho ' !oint6abe(

Property	Type	Description
erie	Boolean	Whether series.
a0entG&PUS	DayTime	Daytime chart area.
10UWE5W-oa	Boolean	Adds a feature when enabled.
ani " ate	Boolean	When enabled, areas are drawn with an effect.
ho ' Datati&	Boolean	Displays a tooltip on hover.
datati&#or " at	String	#or " at over the data tooltip.
ho ' !oint6abe(Boolean	Displays data in inline tooltip.

CategoryAxis*org.primefaces.model.chart.CategoryAxis**org.primefaces.model.categoryAxis*

Property
erie
a0entG&PUS
10UWE5W-oa
ani " ate
ho ' Datati&
datati&#or " at
ho ' !oint6abe(

Property	Type	Description
erie	Boolean	Whether series.
a0entG&PUS	DayTime	Daytime chart area.
10UWE5W-oa	Boolean	Adds a feature when enabled.
ani " ate	Boolean	When enabled, areas are drawn with an effect.
ho ' Datati&	Boolean	Displays a tooltip on hover.
datati&#or " at	String	#or " at over the data tooltip.
ho ' !oint6abe(Boolean	Displays data in inline tooltip.

Property	Default	Type	Description
data	nu((8 a&S 7 b/ectE<u " berT	Data o+ the erie a a "a&.
0a0i	nu((A0i 9y&e	FBA0i o+ the erie .
ya0i	nu((A0i 9y&e	MBA0i o+ the erie .

DateAxis

org.primefaces.model.chart.DateAxis extends *org.primefaces.model.chart.Axis*

DateA0i i u ed ' hen data on the a0i con i t o+ trin, re&re entation o+ date va(ue .

DonutChartModel

org.primefaces.model.chart.DonutChartModel extends *org.primefaces.model.chart.ChartModel*

Property	Default	Type	Description
data	nu((6i tT 8 a&SStrin,E7b/ectTT	Data a a (i t o+ "a& in tance .
(ice 8 ar,in	0	Inte,er	An,u(ar &acin, bet ' een &ie (ice in de,ree .
+i((true	1oo(ean	9rue or #a(e to +i((the (ice .
ho ' Data6abe(+a(e	1oo(ean	9rue to #a(e ho ' data (abe(on (ice .
data#or " at	&ercents	Strin,	Either Y(abe(ZE Yva(ueZE Y&ercents or an array o+ (abe(to &(ace on the &ie (ice .

HorizontalBarChartModel

org.primefaces.model.chart.HorizontalBarChartModel extends

org.primefaces.model.chart.BarChartModel

. ori=onta(1ar*hart 8 ode(i an e0ten ion to 1ar*hart 8 ode(' ith yBa0i u ed +or the cate,orie and 0Ba0i +or the data va(ue .

LineChartSeries

org.primefaces.model.chart.LineChartSeries extends *org.primefaces.model.chart.ChartSeries*

Property	Default	Type	Description
" ar3erSty(e	+i((ed*irc(e	Strin ,	Sty(e o+ the " ar3er E va(id va(ue are diamondE circleE squareE plusE dashE filledDiamondE filledCircleE filledSquare.
ho ' 6ine	true	1oo(ean	Whether to actualy dra ' the (ine or not.
ho ' 8 ar3er	true	1oo(ean	Di &(ay " ar3e at data &oint .
+i((+a(e	1oo(ean	#i((the area bet ' een (ine .
+i((A(&ha	1	Doub(e	7 &acity o+ the +i((ed area.
di ab(eStac3	+a(e	1oo(ean	When trueE erie data i not inc(uded in a tac3ed chart.

LinearAxis

org.primefaces.model.chart.LinearAxis extends *org.primefaces.model.chart.Axis*

6inearA0i i the A0i i " &(e " entation u ed to di &(ay nu " ber .

LineChartModel

org.primefaces.model.chart.LineChartModel extends

org.primefaces.model.chart.CartesianChartModel

Property	Default	Type	Description
tac3ed	nu((6i tS*hartSerie T	Di &(ay erie in tac3ed +or "at.
brea37n<u((. a h 8 a&	8 a&SA0i 9y&EA0i T	Di continue (ine &(ot +or nu((va(ue .

MeterGaugeChartModel

org.primefaces.model.chart.MeterGaugeChartModel extends *org.primefaces.model.ChartModel*

Property	Default	Type	Description
va(ue	nu((<u " ber	>a(ue o+ the ,au,e.
interva(nu((6i tS<u " berT	6i t o+ ran,e to be dra ' n around the ,au,e.
tic3	0	6i tS<u " berT	6i t o+ tic3 va(ue .
,au,e6abe(true	Strin,	6abe(te0t o+ the ,au,e.
,au,e6abe(!o ition	+a(e	Strin,	Where to &o ition the (abe(E either Yin ideZ or Ybotto " Z.
" in	nu((Doub(e	8 ini " u " va(ue on the ,au,e.
" a0	nu((Doub(e	8 ini " u " va(ue on the ,au,e.
ho ' 9ic36abel(true	1oo(ean	Di &(ay tic3 (abe(ne0t to tic3 .
interva(7uterRadius	25	Inte,er	Radius o+ the outer circ(e o+ the interva(rin,..
(abe(. ei,htAd/u t	B%5	Inte,er	<u " ber o+ !i0e(to o++ et the (abe(u& ?B@ or do ' n ?H@ +ro " it de+au(t &o ition.

OhlcChartModel

org.primefaces.model.chart.OhlcChartModel extends *org.primefaces.model.ChartModel*

Property	Default	Type	Description
data	nu((6i tS7h(c*hartSerie T	Data a a (i t o+ 7h(*hartSerie .
cand(eStic3	+a(e	1oo(ean	Di &(ay erie a cand(e tic3.

OhlcChartSeries

org.primefaces.model.chart.OhlcChartSeries extends *org.primefaces.model.ChartSeries*

Property	Default	Type	Description
va(ue	nu((6i tS7h(c*hartSerie T	Data a a (i t o+ 7h(*hartSerie .
o&en	nu((Doub(e	7&en va(ue.
hi,h	nu((Doub(e	. i,h va(ue.
(o '	nu((Doub(e	6o ' va(ue.
c(o e	nu((Doub(e	*(o e va(ue.

PieChartModel

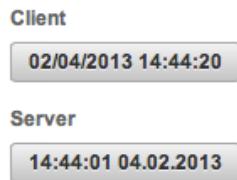
org.primefaces.model.chart.PieChartModel extends *org.primefaces.model.ChartModel*

Property	Default	Type	Description
data	nu((8 a&SStrin,B7b/ectT	Data a a 8 a& in tance.
dia " eter	nu((Inte,er	7uter dia " eter o+ the &ieE auto co " &uted by de+au(t

Property	Default	Type	Description
(ice 8 ar,in	0	Inte,er	An,u(ar &acin, bet ' een &ie (ice in de,ree .
+i((true	1oo(ean	9rue or #a(e to +i((the (ice .
ho ' Data6abe(+a(e	1oo(ean	9rue to #a(e ho ' data (abe(on (ice .
data#or " at	&ercent	Strin,	Either Y(abe(ZE Yva(ueZE Y&ercentZ or an array o+ (abe(to &(ace on the &ie (ice .

3.14 Clock

(oc3 di &(ay er or c(ient dateti " e (ive.



Info

9a,	clock
*o " &onent *(a	org.primefaces.component.clock.Clock
*o " &onent 9y&e	org.primefaces.component.Clock
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ClockRenderer
Renderer *(a	org.primefaces.component.clock.ClockRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean.
&attern	nu((Strin ,	Dateti " e +or " at.
" ode	c(ient	Strin ,	8 ode va(ueE va(id va(ue are c(ient and erver.
autoSync	+a(e	1oo(ean	Sync ti " e &eriodica((y in erver " ode.
yncInterva(40000	Inte ,er	De+ine the ync in " interva(in autoSync ettin ,.

Getting Started with Clock

(oc3 ha t ' o " ode E client ?de+au(t@ and server. In i" &(e " odeE dateti " e i di &(ayed by /u t addin , co " &onent on &a,e. 7n &a,e (oadE c(oc3 i initia(i=ed and tart runnin , ba ed on c(ient ti " e.

```
<p:clock />
```

Server Mode

In server mode, the component is initialized with the server's date/time and starts running on client side. To synchronize the client and server components you can enable the autoSync option that allows an automatic update of the server time with the client.

DateTime Format

The date/time format can be changed using the pattern attribute.

```
<p:clock pattern="HH:mm:ss dd.MM.yyyy" />
```

Skinning

*(oc3 resides in a container element which has style and styleClass options available only in the structural type of a component)

Style Class	Applies
.ui-clock	* container element.

3.15 Collector

*o((ector i a i " &(e uti(ity to " ana , e co((ection dec(arative(y).

Info

9a,	collector
Action6i tener *(a	org.primefaces.component.collector.Collector

Attributes

Name	Default	Type	Description
va(ue	nu((7b/ect	>a(ue to be u ed in co((ection o&eration
add9o	nu((/ava.utি(. *o((ection	Re+erence to the *o((ection in tance
re " ove#ro "	nu((/ava.utি(. *o((ection	Re+erence to the *o((ection in tance
uniCue	true	1oo(ean	When enab(ed) re/ect du&(icate ite " on addition.

Getting started with Collector

*o((ector reCuire a co((ection and a va(ue to 'or3 ' ith. It' i " &ortant to override eCua(and ha h*ode " ethod o+ the va(ue ob/ect to " a3e co((ector 'or3.

```
public class BookBean {
    private Book book = new Book();
    private List<Book> books;

    public CreateBookBean() {
        books = new ArrayList<Book>();
    }

    //getters and setters
}
```

```
<p:commandButton value="Add">
    <p:collector value="#{bookBean.book}" addTo="#{bookBean.books}" />
</p:commandButton>
```

```
<p:commandLink value="Remove">
    <p value="#{book}" removeFrom="#{createBookBean.books}" />
</p:commandLink>
```

3.16 Color Picker

A color picker is an input component with a color palette.



Info

9a,	colorPicker
*o " &onent *(a	org.primefaces.component.colorpicker.ColorPicker
*o " &onent 9y&e	org.primefaces.component.ColorPicker
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ColorPickerRenderer
Renderer *(a	org.primefaces.component.colorpicker.ColorPickerRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify if the component should be rendered. If false, it will not be rendered.
binding	null	Object	An expression or function that maps to a server-side UI component instance in a backing bean.
value	null	Object	The value of the component.
converter	null	Converter	An expression or function that defines a converter for the component. When it's an EL expression, it's resolved to a converter instance. In case it's a static method, it refers to a converter id.
immediate	false	Boolean	When set to true, it performs validation (of, if it's executed at any point before the value has been set to the component).
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method expression that refers to a method that performs validation on the input.

Name	Default	Type	Description
valueChangeListener	null	String	A " method binding, executed when the value changes. It receives the previous value and the new value.
requiredMessage	null	String	The message to display if the field is required and is empty.
converterMessage	null	String	The message to display if the converter fails to convert the value.
validatorMessage	null	String	The message to display if the value does not pass validation.
clientId	null	String	The client ID of the component.
mode	<>	String	Defines the mode: "dialog" or "inline".
type	null	String	The input type, such as "color".
typeOptions	null	String	Style options for the input field.

Getting started with ColorPicker

*Note! The value must be a hex string, .

```
public class Bean {

    private String color;

    public String getColor() {
        return this.color;
    }

    public void setColor(String color) {
        this.color = color;
    }
}
```

```
<p:colorPicker value="#{bean.color}" />
```

Display Mode

*Note! This has to be defined in the page and other available options are inline.

```
<p:colorPicker value="#{bean.color}" mode="inline"/>
```

Skinning

*o(or ! ic3er re ide in a container e(e " ent ' hich *style* and *styleClass* o&tion a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiBco(or&ic3er	*ontainer e(e " ent.
.uiBco(or&ic3erLco(or	1ac3, round o+, radient.
.uiBco(or&ic3erLhue	. ue e(e " ent.
.uiBco(or&ic3erLne ' Lco(or	<e ' co(or di &(ay.
.uiBco(or&ic3erLcurrentLco(or	*urrent co(or di &(ay.
.uiBco(or&ic3erBr,bBr	Red in&ut.
.uiBco(or&ic3erBr,bB,	Greed in&ut.
.uiBco(or&ic3erBr,bBb	1(ue in&ut.
.uiBco(or&ic3erBr,bBh	. ue in&ut.
.uiBco(or&ic3erBr,bB	Saturation in&ut.
.uiBco(or&ic3erBr,bBb	1ri, htne in&ut.
.uiBco(or&ic3erBr,bBhe0	. eo in&ut.

3.17 Column

*o(u " n i an e0tended ver ion o+ the standard co(u " n u ed by variou co " &onent (i3e datab(e)treetab(e and " ore.

Info

9a,	column
*o " &onent *(a	org.primefaces.component.column.Column
*o " &onent 9y&e	org.primefaces.component.Column
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onent ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in , bean
ty(e	nu((Strin ,	In(ine ty(e o+ the co(u " n.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co(u " n.
ort1y	nu((>a(ueE0&r	>a(ueE0&re ion to be u ed +or ortin ,.
ort#unction	nu((8 ethodE0&r	*u to " &(u , ,ab(e ort#unction.
+i(ter1y	nu((>a(ueE0&r	>a(ueE0&re ion to be u ed +or +i(terin ,.
+i(terSty(e	nu((Strin ,	In(ine ty(e o+ the +i(ter e(e " ent
+i(terSty(e*(a	nu((Strin ,	Sty(e c(a o+ the +i(ter e(e " ent
+i(ter 7 &tion	nu((7b/ect	A co((ection o+ e(ectite " +or +i(ter dro&do ' n.
+i(ter 8 atch 8 ode	tart With	Strin ,	8 atch " ode +or +i(terin ,.
ro ' &an	1	Inte ,er	De+ine the nu " ber o+ ro ' the co(u " n &an .
co(&an	1	Inte ,er	De+ine the nu " ber o+ co(u " n the co(u " n &an .
header9e0t	nu((Strin ,	Shortcut +or header +acet.
+ooter9e0t	nu((Strin ,	Shortcut +or +ooter +acet.
e(ection 8 ode	nu((Strin ,	Enab(e e(ection " ode.
di ab(edSe(ection	+a(e	1oo(ean	Di ab(e ro ' e(ection.

Name	Default	Type	Description
+i(ter 8 a06en,th	nu((Inte,er	8 a0i " u " nu " ber o+ character +or an in&ut +i(ter.
re i=ab(e	true	1oo(ean	S&eci+ie re i=ab(e +eature at co(u " n (eve(. Datatab(e) re i=ab(e*o(u " n " u t be enab(ed to u e thi o&tion.
' idth	nu((Strin,	Width in &i0e(or &ercenta,e.
e0&ortab(e	true	1oo(ean	De+ine i+ the co(u " n hou(d be e0&orted by datae0&orter.
+i(ter>a(ue	nu((7b/ect	>a(ue o+ the +i(ter +ie(d.
to,,(eab(e	true	1oo(ean	De+ine i+ &ane(i to,,(eab(e by co(u " n9o,,(er co " &onent. De+au(t va(ue i true and a +a(e va(ue " ar3 the co(u " n a static.
+i(ter#unction	nu((8 ethodE0&r	*u to " i " &(e " entation to +i(ter a va(ue a,ain t a con traint.
+ie(d	nu((Strin,	<a " e o+ the +ie(d to &a (a=y (oad " ethod +or +i(terin, and ortin,. I+ not &eci+iedE +i(ter1yB ort1y va(ue are u ed to identi+y the +ie(d na " e.

Getting Started with Column

A co(u " n i a reu ed co " &onentE ee docu " entation o+ co " &onent that u e a co(u " n.

Note

<ot a((attribute o+ co(u " n are i " &(e " ented by the co " &onent that uti(i=e co(u " n.

3.18 Columns

*o(u " n i u ed by databab(e to create co(u " n dyna " ica((y.

Info

9a,	columns
*o " &onent *(a	org.primefaces.component.column.Columns
*o " &onent 9y&e	org.primefaces.component.Columns
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
va(ue	nu((7b/ect	Data to re&re ent co(u " n .
var	nu((Strin ,	<a " e o+ iterator to acce a co(u " n.
ty(e	nu((Strin ,	In(ine ty(e o+ the co(u " n.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co(u " n.
ort1y	nu((>a(ueE0&r	>a(ueE0&re ion to be u ed +or ortin,..
ort#unction	nu((8 ethodE0&r	*u to " &(u , ab(e ort#unction.
+i(ter1y	nu((>a(ueE0&r	>a(ueE0&re ion to be u ed +or +i(terin,..
+i(terSty(e	nu((Strin ,	In(ine ty(e o+ the +i(ter e(e " ent
+i(terSty(e*(a	nu((Strin ,	Sty(e c(a o+ the +i(ter e(e " ent
+i(ter7 &tion	nu((7b/ect	A co((ection o+ e(ectite " +or +i(ter dro&do ' n.
+i(ter 8 atch 8 ode	tart With	Strin ,	8 atch " ode +or +i(terin,..
ro ' &an	1	Inte ,er	De+ine the nu " ber o+ ro ' the co(u " n &an .
co(&an	1	Inte ,er	De+ine the nu " ber o+ co(u " n the co(u " n &an .
header9e0t	nu((Strin ,	Shortcut +or header +acet.

Name	Default	Type	Description
+ooter9e0t	nu((Strin ,	Shortcut +or +ooter +acet.
+i(ter 8 a06en,th	nu((Inte ,er	8 a0i " u " nu " ber o+ character +or an in&ut +i(ter.
re i=ab(e	true	1oo(ean	S&eci+ie re i=ab(e +eature at co(u " n (eve(. Datatab(e) re i=ab(e*o(u " n " u t be enab(ed to u e thi o&tion.
' idth	nu((Strin ,	Width in &i0e(or &ercenta,e.
e0&ortab(e	true	1oo(ean	De+ine i+ the co(u " n hou(d be e0&orted by datae0&orter.
co(u " nInde0>ar	nu((Strin ,	<a " e o+ iterator to re+er each inde0.
+i(ter>a(ue	nu((7b/ect	>a(ue o+ the +i(ter +ie(d.
to , ,eab(e	true	1oo(ean	De+ine i+ &ane(i to , ,eab(e by co(u " n9o , ,(er co " &onent. De+au(t va(ue i true and a +a(e va(ue " ar3 the co(u " n a tatic.
+i(ter#unction	nu((8 ethodE0&r	*u to " i "&(e " entation to +i(ter a va(ue a,ain t a con traint.
+ie(d	nu((Strin ,	<a " e o+ the +ie(d to &a (a=y (oad " ethod +or +i(terin, and ortin,. I+ not &eci+iedE+i(ter1yB ort1y va(ue are u ed to identi+y the +ie(d na " e.

Getting Started with Columns

See dyna " ic co(u " n ection in datatab(e docu " entation +or detai(ed in+or " ation.

3.19 ColumnGroup

*o(u " nGrou& i u ed by databab(e +or co(u " n ,rou&in,,.

Info

9a,	columnGroup
*o " &onent *(a	org.primefaces.component.columngroup.ColumnGroup
*o " &onent 9y&e	org.primefaces.component. ColumnGroup
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
ty&e	nu((Strin ,	9y&e o+ ,rou&E va(id va(ue are “header” and “+ooter”.

Getting Started with ColumnGroup

See ,rou&in, ection in databab(e docu " entation +or detai(ed in+or " ation.

3.20 CommandButton

*o " " and 1utton i an e0tended ver ion o+ standard co " " and 1utton ' ith a/a0 and the " in, .

Ajax Submit

Non-Ajax Submit

With Icon



Disabled

Info

9a,	commandButton
*o " &onent *(a	org.primefaces.component.commandbutton.CommandButton
*o " &onent 9y&e	org.primefaces.component.CommandButton
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.CommandButtonRenderer
Renderer *(a	org.primefaces.component.commandbutton.CommandButtonRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier o+ the co " &onent
rendered	true	Boolean	True(ean va(ue to &ecify the renderin, o+ the co " &onent) ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	null	Object	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
va(ue	null	String,	6abe(+or the button
action	null	MethodExpression String,	A " ethod e0&re ion or a String, outco " e that'd be &roce ed ' hen button i c(ic3ed.
action6i tener	null	MethodExpression	An action(i tener that'd be &roce ed ' hen button i c(ic3ed.
i " " ediate	+a(e	Boolean	True(ean va(ue that deter " ine the &ha elde ' hen true action are &roce ed at a&&(yLreCue tLva(ue E ' hen +a(e at inv03eLa&&(ication &ha e.
ty&e	ub " it	String,	Set the behavior o+ the button.
a/a0	true	Boolean	S&eci+ie the ub " it " odeE ' hen et to true?de+au(tEE ub " it ' ou(d be " ade ' ith A/a0.
a ync	+a(e	Boolean	When et to trueE a/a0 reCue t are not Cueued.
&roce	null	String,	*o " &onent? @ to &roce &artia((y in tead o+ ' ho(e vie ' .
u&date	null	String,	*o " &onent? @ to be u&dated ' ith a/a0.

Name	Default	Type	Description
on:tart	null	String,	*(identideca((bac3 to execute before a/a0 reCue t i be,in).
onco:"&(ete	null	String,	*(identideca((bac3 to execute 'hen a/a0 reCue t i co" &(eted).
on:ucces	null	String,	*(identideca((bac3 to execute 'hen a/a0 reCue t ucceed).
on:error	null	String,	*(identideca((bac3 to execute 'hen a/a0 reCue t +ai().
,(oba(true	Boolean	Defines whether to tri, ,er a/a0Statu or not.
de:ay	null	String,	I+ (e than delay " i((i econd e(a& e bet ' een ca((to request() on(y the " o t recent one i ent and a((other reCue t are di cated. I+ thi option i not &eciated or i+ the va(ue o+ delay i the (itera(trin, (none) ' ithout the Cuote E no de(ay i u ed.
&artia(Sub " it	+a(e	Boolean	Enab(e eria(i=ation o+ va(ue be(on,in, to the &artia((y &roce ed co" &onent on(y).
re:et>a(ue	+a(e	Boolean	I+ trueE (oca(va(ue o+ in&ut co" &onent to be u&dated ' ithin the a/a0 reCue t ' ou(d be re et.
i,noreAutoU&date	+a(e	Boolean	I+ trueE co" &onent ' hich autoU&datePItrueI ' i((not be u&dated +or thi reCue t. I+ not &eciated or the va(ue i +a(eE no uch indication i " ade.
ty(e	null	String,	In(ine ty(e o+ the button e(e " ent.
ty(e*(a	null	String,	Sty(e*(a o+ the button e(e " ent.
on:button	null	String,	*(identideca((bac3 to execute 'hen button (o e +ocu).
on:chan,e	null	String,	*(identideca((bac3 to execute 'hen button (o e +ocu and it va(ue ha been " odi+ied ince ,ainin, +ocu).
on:click3	null	String,	*(identideca((bac3 to execute 'hen button i c(ic3ed).
on:dblclick3	null	String,	*(identideca((bac3 to execute 'hen button i doub(e c(ic3ed).
on:ocu	null	String,	*(identideca((bac3 to execute 'hen button receive +ocu).
on:3eydo'n	null	String,	*(identideca((bac3 to execute 'hen a 3ey i &re ed do'n over button.
on:3ey&re	null	String,	*(identideca((bac3 to execute 'hen a 3ey i &re ed and re(ea ed over button.
on:3eyu&	null	String,	*(identideca((bac3 to execute 'hen a 3ey i re(ea ed over button.
on:"ou:edo'n	null	String,	*(identideca((bac3 to execute 'hen a &ointer button i &re ed do'n over button.

Name	Default	Type	Description
on " ou e " ove	null	String,	*(ient ide ca((bac3 to execute ' hen a &ointer button i " oved ' ithin button.
on " ou eout	null	String,	*(ient ide ca((bac3 to execute ' hen a &ointer button i " oved a ' ay +ro " button.
on " ou eover	null	String,	*(ient ide ca((bac3 to execute ' hen a &ointer button i " oved onto button.
on " ou eu&	null	String,	*(ient ide ca((bac3 to execute ' hen a &ointer button i re(ea ed over button.
on e(ect	null	String,	*(ient ide ca((bac3 to execute ' hen te0t ' ithin button i e(ected by u er.
acce 3ey	null	String,	Acce 3ey that ' hen &re ed tran +er +ocu to the button.
a(t	null	String,	A(ternate te0tua(de cri&tion o+ the button.
dir	null	String,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
di ab(ed	+a(e	Boolean	Di ab(e the button.
i " a,e	null	String,	Sty(e c(a +or the button icon. ?de&recatedN u e icon@
(abe(null	String,	A(oca(i=ed u er &re entab(e na " e.
(an,	null	String,	*ode de cribin, the (an,ua,e u ed in the ,enerated " ar3u& +or thi co " &onent.
tabinde0	null	Integer	!o ition o+ the button e(e " ent in the tabbin, order.
tit(e	null	String,	Advi ory too(ti& in+or " ation.
readon(y	+a(e	Boolean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
icon	null	String,	Icon o+ the button a a c c(a .
icon !o	(e+e	String,	!o ition o+ the icon.
in(ine	+a(e	String,	U ed by !ri " e#ace " obi(e on(y.
e ca&e	true	Boolean	De+ine ' hether (abe(' ou(d be e ca&ed or not.
' id,et>ar	null	String,	<a " e o+ the c(ient ide ' id,et.

Getting started with CommandButton

*o " " and1utton u a,e i i " i(ar to standard co " " and1uttonE by de+au(t co " " and1utton ub " it it enc(o in, +or " ' ith a/a0.

```
<p:commandButton value="Save" actionListener="#{bookBean.saveBook}" />
```

```
public class BookBean {  
  
    public void saveBook() {  
        //Save book  
    }  
}
```

Reset Buttons

Reset button do not submit the form but instead reset the form content.

```
<p:commandButton type="reset" value="Reset" />
```

Push Buttons

Push buttons are used to execute code without causing an asynchronous request. To create a push button set type to button.

```
<p:commandButton type="button" value="Alert" onclick="alert('Prime')"/>
```

AJAX and Non-AJAX

Normal buttons have built-in functionality to submit the form and reload it by default and controlled via the ajax attribute. When ajax attribute is set to true the form is submitted with a regular POST request.

The update attribute is used to update other components after the normal response is received. Update attribute takes a component ID or the id of a separated (in another JSF component) component to be updated. It can apply any JSF component not just the one that triggered the update component. In this case, the bean's value is updated with the display output component's value.

```
<h:form>  
    <h:inputText value="#{bean.text}" />  
    <p:commandButton value="Submit" update="display"/>  
    <h:outputText value="#{bean.text}" id="display" />  
</h:form>
```

Tip: You can use the ajaxStatus component to notify users about the ajax request.

Icons

An icon on a button is provided using the `icon` option. icon!o is used to define the icon on the button which can be "(elt" or "ri,ht".

```
<p:commandButton value="With Icon" icon="disk"/>
<p:commandButton icon="disk"/>
```

.di 3 i a i" &(e c c(a ' ith a bac3, round &ro&erty

```
.disk {
    background-image: url('disk.png') !important;
}
```

You can also use the predefined icon +ro " 9he " eRo(er (i3e `ui-icon-search`.

Client Side API

Widget `PrimeFaces.widget.CommandButton`

3.21 CommandLink

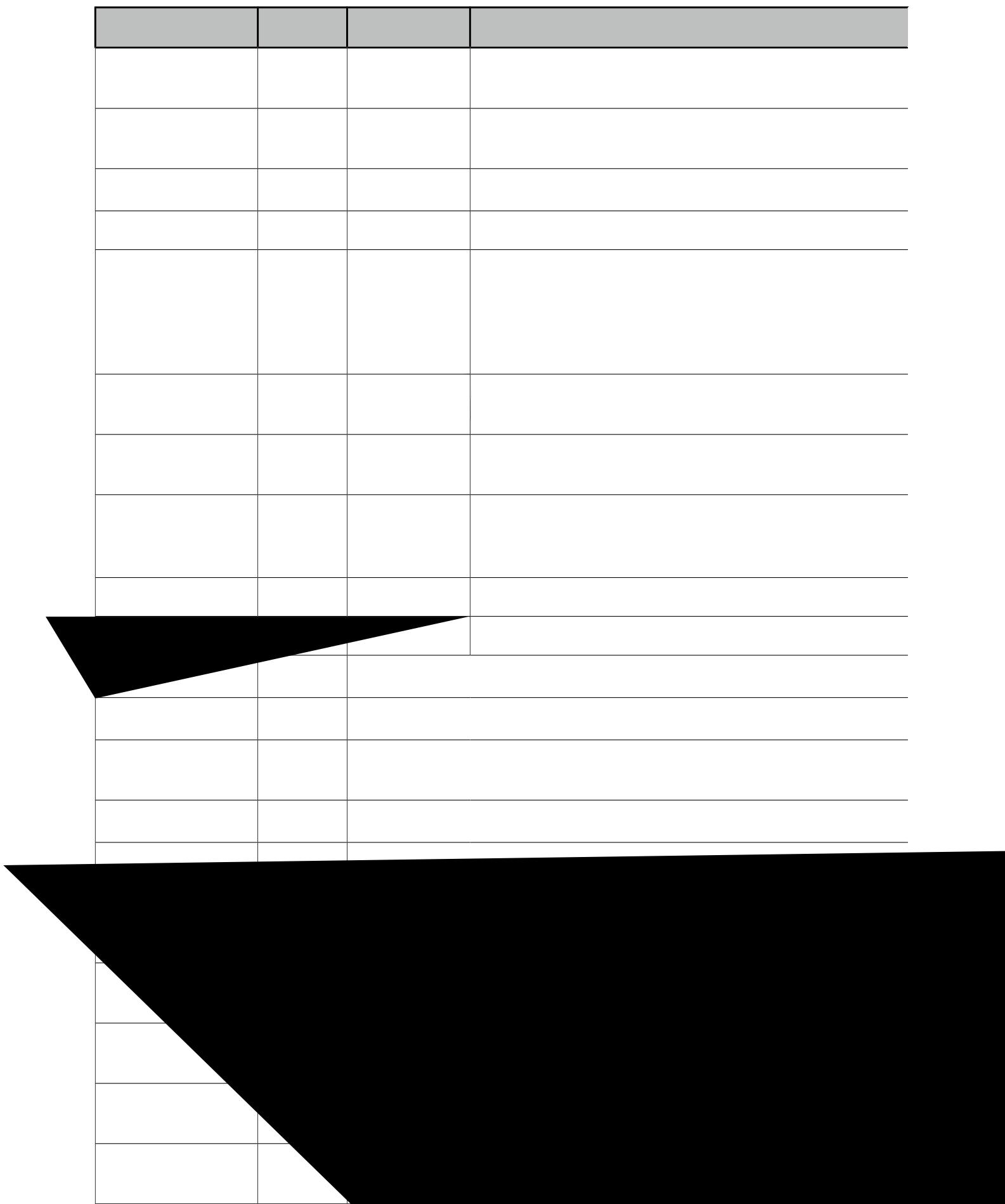
*o " " and6in3 e0tend standard AS# co " " and6in3 ' ith A/a0 ca&abi(itie .

Info

9a,	commandLink
*o " &onent *(a	org.primefaces.component.commandlink.CommandLink
*o " &onent 9y&e	org.primefaces.component.CommandLink
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.CommandLinkRenderer
Renderer *(a	org.primefaces.component. commandlink.CommandLinkRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier for the component.
rendered	true	Boolean	True (value to identify the rendering, or the component) when set to false the component will not be rendered.
binding	null	Object	An expression identifying that it is to be rendered in a server-side UI component instance in a backing bean.
value	null	String,	The value of the rendered anchor.
action	null	MethodExpression or String,	A method expression or a String, output which is to be processed when (in) invoked.
actionListener	null	MethodExpression	An action listener that's to be processed when (in) invoked.
immediate	false	Boolean	True (value that defines whether the component is to be processed at the time of the action rather than at the end of the request) if true the action is to be processed at the end of the request.
async	false	Boolean	When set to true the components are not queued.
process	null	String,	*o " &onent? @ to process &attribute (y instead of ' href').
auto	true	Boolean	Specifies the submit button to be processed when set to true?de=auto the button is to be submitted with the form.
update	null	String,	*o " &onent? @ to be updated with auto.
onstart	null	String,	(identifies the bac3 to execute before auto update is to be performed).



Name	Default	Type	Description
on " ou eover	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto (in3.
on " ou eu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over (in3.
acce 3ey	nu((Strin ,	Acce 3ey that ' hen &re ed tran +er +ocu to the (in3.
char et	nu((Strin ,	*haracter encodin , o+ the re ource de i ,nated by thi hy&er(in3.
coord	nu((Strin ,	!o ition and ha&e o+ the hot &ot on the creen +or c(ient u e in i " a,e " a& .
dir	nu((Strin ,	Direction indication +or te0t that doe not inherit directiona(it. >a(id va(ue are 69R and R96.
di ab(ed	nu((1oo(ean	Di ab(e the (in3
hre+(an ,	nu((Strin ,	6an ,uae code o+ the re ource de i ,nated by the (in3.
re(nu((Strin ,	Re(ation hi& +ro " the current docu " ent to the anchor &eci+ied by the (in3E va(ue are &rovvided by a &aceB e&arated (i t o+ (in3 ty&e .
rev	nu((Strin ,	A rever e (in3 +ro " the anchor &eci+ied by thi (in3 to the current docu " entE va(ue are &rovvided by a &aceB e&arated (i t o+ (in3 ty&e .
ha&e	nu((Strin ,	Sha&e o+ hot &ot on the creenE va(id va(ue are de+au(tE rectE circ(e and &o(y.
tabinde0	nu((Inte ,er	!o ition o+ the button e(e " ent in the tabbin , order.
tar ,et	nu((Strin ,	<a " e o+ a +ra " e ' here the re ource tar ,eted by thi (in3 ' i((be di &(ayed.
tit(e	nu((Strin ,	Advi ory too(ti& in+or " ation.
ty&e	nu((Strin ,	9y&e o+ re ource re+erenced by the (in3.

Getting Started with CommandLink

*o " " and6in3 i u ed /u t (i3e the standard h&co " " and6in3E di+ference i +or " i ub " itted ' ith a/a0 by de+au(t.

```
public class BookBean {

    public void saveBook() {
        //Save book
    }
}
```

```
<p:commandLink actionListener="#{bookBean.saveBook}">
    <h:outputText value="Save" />
</p:commandLink>
```

Skinning

o " " and render an h t " (anchor element that *style* and *styleClass* attribute a&&(y.

3.22 Confirm

*on+ir" i a behavior e(e " ent u ed to inte ,rate ' ith ,(oba(con+ir" dia(o,..

Info

9a,	confirm
Behavior Id	org.primefaces.behavior.ConfirmBehavior

Attributes

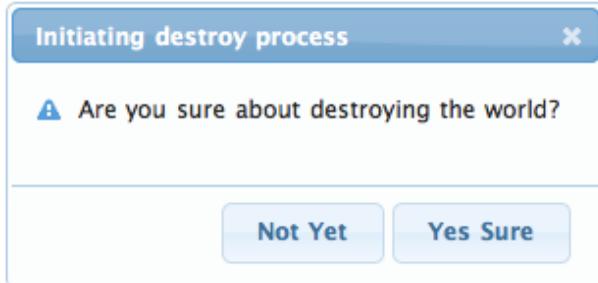
Name	Default	Type	Description
header	null	String	. header o+ con+ir" dia(o,..
"e a,e	null	String	8 e a,e to di &(ay in con+ir" dia(o,..
icon	null	String	Icon to di &(ay ne0t to "e a,e.

Getting started with Confirm

See ,(oba(con+ir" dia(o, to&ic in ne0t ection +or detail(.

3.22 ConfirmDialog

*on+ir " Dia(o, i a re&(ace " ent to the (e,acy /ava cri&t con+ir " ation bo0. S3innin, E cu to " i=ation and avoidin, &o&u& b(oc3er are notab(e advanta,e over c(a ic /ava cri&t con+ir " ation.



Info

9a,	confirmDialog
*o " &onent *(a	org.primefaces.component.confirmdialog.ConfirmDialog
*o " &onent 9y&e	org.primefaces.component.ConfirmDialog
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ConfirmDialogRenderer
Renderer *(a	org.primefaces.component.confirmdialog.ConfirmDialogRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
" e a,e	nu((Strin ,	9e0t to be di &(ayed in body.
header	nu((Strin ,	9e0t +or the header.
everity	nu((Strin ,	8 e a,e everity +or the di &(ayed icon.
' idth	auto	Inte ,er	Width o+ the dia(o , in &i0e(
hei ,ht	auto	Inte ,er	Width o+ the dia(o , in &i0e(
ty(e	nu((Strin ,	In(ine ty(e o+ the dia(o , container.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the dia(o , container

Name	Default	Type	Description
c(o ab(e	true	1oo(ean	De+ine i+ c(o e icon hou(d be di &(ayed or not
a&&end9o	+a(e	1oo(ean	A&&end the dia(o, to the e(e " ent de+ined by the ,iven earch e0&re ion.
vi ib(e	+a(e	1oo(ean	Whether to di &(ay con+ir " dia(o, on (oad.
ho ' E++ect	nu((Strin,	E++ect to u e on ho ' in, dia(o,..
hideE++ect	nu((Strin,	E++ect to u e on hidin, dia(o,..
c(o e7nE ca&e	+a(e	1oo(ean	De+ine i+ dia(o, hou(d hide on e ca&e 3ey.
dir	(tr	Strin,	De+ine te0t directionE va(id va(ue are <i>ltr</i> and <i>rtl</i> .
,(oba(+a(e	1oo(ean	When enab(ed) con+ir " Dia(o, beco " e a hared +or other co " &onent that reCuire con+ir " ation.

Getting started with ConfirmDialog

*on+ir " Dia(o, ha t' o " ode K ,(oba(and nonB,(oba(. <onBG(oba(" ode i a(" o t a" e a the dia(o, co " &onent u ed ' ith a i " &(e c(ient ide a&iE *show()* and *hide()*.

```
<h:form>
    <p:commandButton type="button" onclick="PF('cd').show()" />

    <p:confirmDialog message="Are you sure about destroying the world?"
        header="Initiating destroy process" severity="alert"
        widgetVar="cd">
        <p:commandButton value="Yes Sure" actionListener="#{buttonBean.destroyWorld}"
            update="messages" oncomplete="PF('cd').hide()"/>
        <p:commandButton value="Not Yet" onclick="PF('cd').hide();" type="button" />
    </p:confirmDialog>
</h:form>
```

Message and Severity

8 e a,e can be de+ined in t' o ' ay E either via " e a,e o&tion or " e a,a,e +acet. 8 e a,a,e +acet i u e+u(i+ you need to &(ace cu to " content in tead o+ i " &(e te0t. <ote that header can a(o be de+ined u in, the *header* attribute or the *header* +acet. Severity de+ine the icon to di &(ay ne0t to the " e a,a,eE de+au(t everity i *alert* and the other o&tion i *info*.

```
<p:confirmDialog widgetVar="cd" header="Confirm">
    <f:facet name="message">
        <h:outputText value="Are you sure?" />
    </f:facet>
    //content
</p:confirmDialog>
```

Global

*reatin, a con+ir " Dia(o, +or a &eci+ic action i a re&etitive ta 3E to o(ve thi ,,(oba(con+ir " Dia(o, ' hich i a in,(eton ha been introduced. 9ri, ,er co " &onent need to have &Ncon+ir " behavior to

Use the `oncomplete` attribute to trigger the actual code and in dialog, "if you have `ui-confirm-dialog-yes` type confirmation to cancel the code and "if you have `ui-confirm-dialog-no`. At the "oncomplete" event ×ir" is triggered by ×o" and 1utton and ×o" and 6in3.

```
<p:growl id="messages" />

<p:commandButton value="Save" actionListener="#{bean.save}" update="messages">
    <p:confirm header="Confirmation" message="Sure?" icon="ui-icon-alert"/>
</p:commandButton>

<p:confirmDialog global="true">
    <p:commandButton value="Yes" type="button" styleClass="ui-confirmdialog-yes" icon="ui-icon-check"/>
    <p:commandButton value="No" type="button" styleClass="ui-confirmdialog-no" icon="ui-icon-close"/>
</p:confirmDialog>
```

Client Side API

Widget `PrimeFaces.widget.ConfirmDialog`

Method	Params	Return Type	Description
<code>show()</code>		<code>void</code>	Show dialog.
<code>hide()</code>		<code>void</code>	Hides dialog.

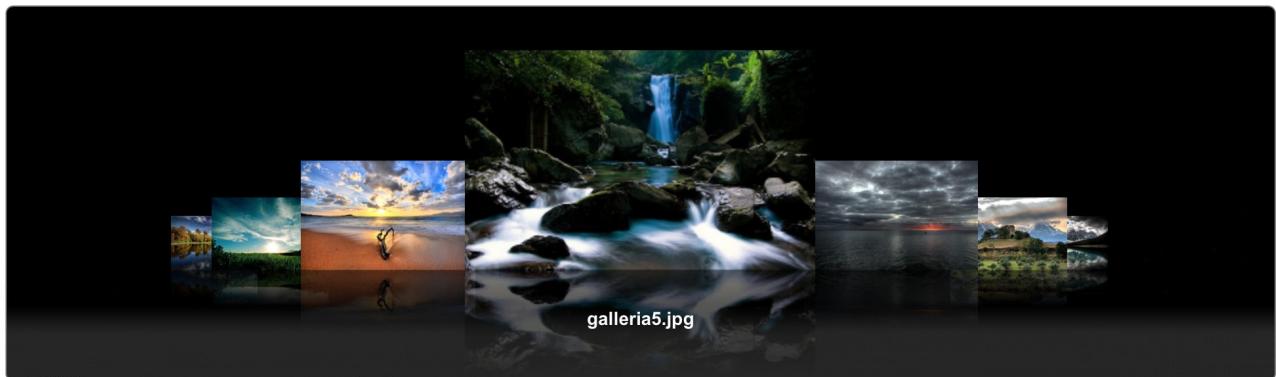
Skinning

*on+ir" Dia(o, re ide in a "ain container e(e " ent ' hich *style* and *styleClass* o&&(y. A 3innin, ty(e c(a e are ,(obaE ee the "ain the " in, ection +or "ore in+or "ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
<code>.uiBdia(o,</code>	*ontainer e(e " ent o+ dia(o,
<code>.uiBdia(o,Btit(ebar</code>	9it(e bar
<code>.uiBdia(o,Btit(e</code>	. eader te0t
<code>.uiBdia(o,Btit(ebarBc(o e</code>	*(o e icon
<code>.uiBdia(o,Bcontent</code>	Dia(o, body
<code>.uiBdia(o,Bbutton&ane</code>	#ooter button &ane(

3.23 ContentFlow

*ontent#(o' i a hori=onta(content ,a((ery co " &onent ' ith a (ide ani " ation.



Info

9a,	contentFlow
*o" &onent *(a	org.primefaces.component.contentflow.ContentFlow
*o" &onent 9y&e	org.primefaces.component.ContentFlow
*o" &onent #a" i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ContentFlowRenderer
Renderer *(a	org.primefaces.component.contentflow.ContentFlowRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o" &onent in tance in a bac3in, bean.
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
va(ue	nu((Strin ,	*o((ection o+ ite " to di &(ay.
var	nu((Strin ,	<a " e o+ the iterator to di &(ay an ite " .
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.

Getting Started with ContentFlow

*ontent#(o ' reCuire content a chi(dren that can either be de+ined dyna" ica((y u in, iteration or one by one. Each ite " " u t have the content ty(e c(a a&&(ied a ' e((.

Static | "a,e

```
<p:contentFlow>
    <p:graphicImage value="/images/photo1.jpg" styleClass="content" />
    <p:graphicImage value="/images/photo2.jpg" styleClass="content" />
    <p:graphicImage value="/images/photo2.jpg" styleClass="content" />
</p:contentFlow>
```

Dyna " ic | "a,e

```
<p:contentFlow var="image" value="#{bean.images}">
    <p:graphicImage value="#{image}" styleClass="content" />
</p:contentFlow>
```

Caption

9o &re ent a ca&tion a(on, ' ith an ite " E e " bed a div ' ith Ica&tionI ty(e c(a in ide.

```
<p:contentFlow var="image" value="#{bean.images}">
    <p:graphicImage value="#{image.path}" styleClass="content" />
    <div class="caption">#{image.title}</div>
</p:contentFlow>
```

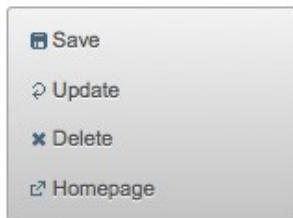
Skinning

*ontent#(o ' re ide in a container e(e " ent ' hich *style* and *styleClass* o&ption a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiContent+(o '	*ontainer e(e " ent.
.+(o '	*ontainer o+ ite " (i t
.ite "	It e " container
.ca&tion	*a&tion e(e " ent

3.24 ContextMenu

*onteot 8 enu &rovide an over(ay " enu di &(ayed on " ou e ri ,htBc(ic3 event.



Info

9a,	contextMenu
*o " &onent *(a	org.primefaces.component.contextmenu.ContextMenu
*o " &onent 9y&e	org.primefaces.component.ContextMenu
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ContextMenuRenderer
Renderer *(a	org.primefaces.component.contextmenu.ContextMenuRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onent ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in , bean
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et.
+or	nu((Strin ,	Id o+ the co " &onent to attach to
ty(e	nu((Strin ,	Sty(e o+ the " ain container e(e " ent
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the " ain container e(e " ent
" ode(nu((8 enu 8 ode(8 enu " ode(in tance to create " enu &ro ,ra " " atica((y.
node9y&e	nu((Strin ,	S&eci+ic ty&e o+ tree node to attach to.
event	nu((Strin ,	Event to bind conteot 8 enu di &(ayed de+au(t i conteot " enu a3a ri ,ht c(ic3.
be+oreSho '	nu((Strin ,	*(ient ide ca((bac3 to e0ecute be+ore ho ' in ,.

Getting started with ContextMenu

*ontenot 8 enu i created 'ith ub "enu and "enuite" . 7&tional(+or attribute define ' hich co " &onent the conteot 8 enu i attached to. When +or i not defined conteot 8 enu i attached to the &a,e "eanin,E ri,htBc(ic3 on any ' here on &a,e ' i((di &(ay the "enu.

```
<p:contextMenu>
    <p:menuItem value="Save" actionListener="#{bean.save}" update="msg"/>
    <p:menuItem value="Delete" actionListener="#{bean.delete}" ajax="false"/>
    <p:menuItem value="Go Home" url="www.primefaces.org" target="_blank"/>
</p:contextMenu>
```

*ontenot 8 enu e0a " &(e above i attached to the ' ho(e &a,e and con i t o+ three di+erent "enuite" ' ith di+erent u e ca e . #ir t "enuite" tri,,er an a/a0 actionE econd one tri,,er a nonBa/a0 action and third one i u ed +or navi,ation.

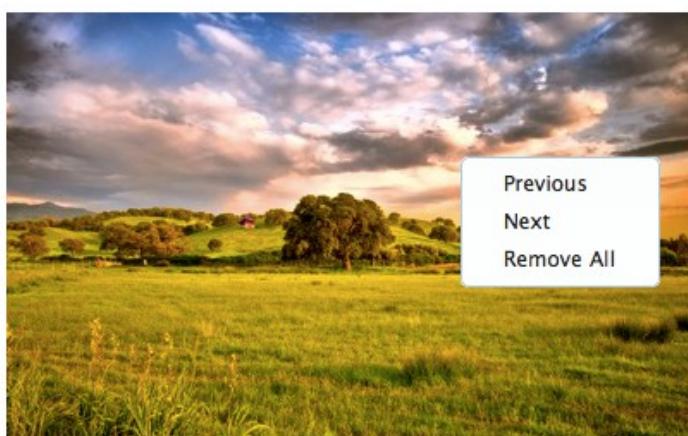
Attachment

*ontenot 8 enu can be attached to any AS# co " &onent thi "ean ri,ht c(ic3in, on the attached co " &onent ' i((di &(ay the conteot 8 enu. #o((o in, e0a " &(e de "on trate an inte,ration bet ' een conteot 8 enu and i " a,eS ' itcher conteot 8 enu here i u ed to navi,ate bet ' een i " a,e .

```
<p:imageSwitch id="images" widgetVar="gallery" slideshowAuto="false">
    <p:graphicImage value="/images/nature1.jpg" />
    <p:graphicImage value="/images/nature2.jpg" />
    <p:graphicImage value="/images/nature3.jpg" />
    <p:graphicImage value="/images/nature4.jpg" />
</p:imageSwitch>

<p:contextMenu for="images">
    <p:menuItem value="Previous" url="#" onclick="PF('gallery').previous()" />
    <p:menuItem value="Next" url="#" onclick="PF('gallery').next()" />
</p:contextMenu>
```

<o ' ri,htBc(ic3in, any ' here on an i " a,e ' i((di &(ay the conteot 8 enu (i3eK



Data Components

Data co " &onent (i3e databab(e tree and tree9ab(e ha &ecia(inte,ration ' ith conteot " enuE ee the docu " entation o+ the e co " &onent +or " ore in+or " ation.

Dynamic Menus

*onteot 8 enu can be created &ro, ra" "atica((y a 'e(E ee the dyna" ic "enu &art in "enu co " &onent ection +or " ore in+or " ation and an e0a " &(e.

Skinning

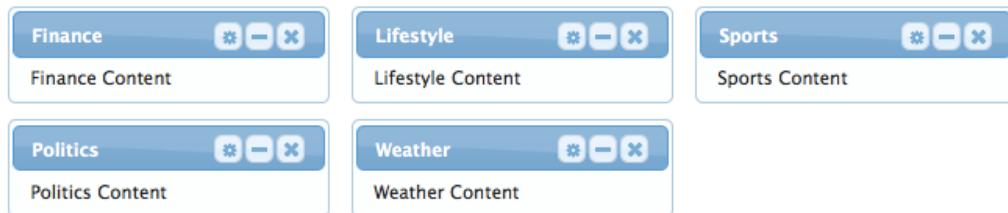
*onteot 8 enu re ide in a "ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiBconteot " enu	*ontainer e(e " ent o+ " enu
.uiB " enuB(i t	6i t container
.uiB " enuite "	Each " enu ite "
.uiB " enuite " B(in3	Anchor e(e " ent in a (in3 ite "
.uiB " enuite " Bteot	9e0t e(e " ent in an ite "

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or " ore in+or " ation.

3.25 Dashboard

Da hboard &rovide a &orta(i3e (ayout ' ith dra, : dro& ba ed reorder ca&abi(itie .



Info

9a,	dashboard
*o "&onent *(a	org.primefaces.component.dashboard.Dashboard
*o "&onent 9y&e	org.primefaces.component.Dashboard
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DashboardRenderer
Renderer *(a	org.primefaces.component.dashboard.DashboardRenderer

Attributes

Getting started with Dashboard

The dashboard is based on a `Dashboard` component and contains one or more components.

```
<p:dashboard model="#{bean.model}">
    <p:panel id="sports">
        //Sports Content
    </p:panel>
    <p:panel id="finance">
        //Finance Content
    </p:panel>

    //more panels like lifestyle, weather, politics...
</p:dashboard>
```

The dashboard "model" defines the number of columns and the height to be used in each column. See the end of this section for the detailed Dashboard API.

```
public class Bean {

    private DashboardModel model;

    public Bean() {
        model = new DefaultDashboardModel();
        DashboardColumn column1 = new DefaultDashboardColumn();
        DashboardColumn column2 = new DefaultDashboardColumn();
        DashboardColumn column3 = new DefaultDashboardColumn();

        column1.addWidget("sports");
        column1.addWidget("finance");
        column2.addWidget("lifestyle");
        column2.addWidget("weather");
        column3.addWidget("politics");

        model.addColumn(column1);
        model.addColumn(column2);
        model.addColumn(column3);
    }
}
```

State

The dashboard is a stateful component whenever a panel is reordered the dashboard "model" is updated by the user in, the user changes so you can easily create a stateful dashboard.

Ajax Behavior Events

"reorder" is the one and only available behavior event provided by the dashboard. This event is fired when the panels are reordered. A defined (it is) is invoked by and in, an `org.primefaces.event.DashboardReorderEvent` instance containing information about the reorder.

#0(0 in, the dashboard displays a message about the reorder event

```
<p:dashboard model="#{bean.model}">
    <p:ajax event="reorder" update="messages" listener="#{bean.handleReorder}" />
    //panels
</p:dashboard>

<p:growl id="messages" />
```

```
public class Bean {

    ...

    public void handleReorder(DashboardReorderEvent event) {
        String widgetId = event.getWidgetId();
        int widgetIndex = event.getItemIndex();
        int columnIndex = event.getColumnIndex();
        int senderColumnIndex = event.getSenderColumnIndex();

        //Add facesmessage
    }
}
```

If a 'id,et i reordered in the a"e co(u "nE senderColumnIndex ' i((be nu(. 9hi +ie(d i &o&u(ated on(y 'hen a 'id,et i tran +erred to a co(u "n +ro" another co(u "n. A(o 'hen the (i tener i involed da hboard ha a(ready u&dated it' "ode(.

Disabling Dashboard

If you'd like to disable reordering, feature set *disabled* option to true.

```
<p:dashboard disabled="true" ...>
    //panels
</p:dashboard>
```

Toggle, Close and Options Menu

Widgets presented in da hboard can be closed to , ,eab(e and have option "enu a 'e((E da hboard doe n't i "(&e" ent the e by it e(+ a the e +eature are a(ready &rovied by the &ane(co " &onent. See &ane(co " &onent ection +or " ore in+or " ation.

```
<p:dashboard model="#{dashboardBean.model}">
    <p:panel id="sports" closable="true" toggleable="true">
        //Sports Content
    </p:panel>
</p:dashboard>
```

New Widgets

Dra, ,ab(e co " &onent i u ed to add ne ' ' id,et to the da hboard. 9hi ' ay you can add ne ' &ane(+ro " out ide o+ the da hboard.

```
<p:dashboard model="#{dashboardBean.model}" id="board">
    //panels
</p:dashboard>
<p:panel id="newwidget" />
<p:draggable for="newwidget" helper="clone" dashboard="board" />
```

Skinning

Da hboard re ide in a container e(e " ent ' hich ty(e and ty(e*(a o&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiDashboard	*ontainer e(e " ent o+ da hboard
.uiDashboardContent	Each co(u " n in da hboard
div.uiB tateBhover	!aceho(der

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. . ere i an e0a " &(e ba ed on a di+ferent the " ek



Tips

- Provide a co(u " n ' idth u in, *ui-dashboard-column* ty(e c(a other ' i e e " &ty co(u " n " i, ht not receive ne ' ' id, et .

Dashboard Model API

org.primefaces.model.DashboardModel ?*org.primefaces.model.map.DefaultDashboardModel* i the de+au(t i " &(e " entation@

Method	Description
void add*(o(u " n?Da hboard*o(u " n co(u " n@	Add a co(u " n to the da hboard
6i tSDa hboard*o(u " nT ,et*o(u " n ?@	Return a((co(u " n in da hboard
int ,et*o(u " n*ount?@	Return the nu " ber o+ co(u " n in da hboard
Da hboard*o(u " n ,et*o(u " n?int inde0@	Return the da hboard co(u " n at ,iven inde0
void tran +erWid,et?Da hboard*o(u " n +ro " E Da hboard*o(u " n toE Strin, ' id,etIdE int inde0@	Re(ocate the ' id,et identi+ed ' ith ' id,et id to the ,iven inde0 o+ the ne ' co(u " n +ro " o(d co(u " n.

*org.primefaces.model.DashboardColumn ?org.primefaces.model.map.DefaultDashboardModel i
the de+au(t i " &(e " entation@*

Method	Description
void re " oveWid,et?String, ' id,etId@	Re " ove the ' id,et ' ith the ,iven id
6i tSStrin,T ,etWid,et ?@	Return the id o+ ' id,et in co(u " n
int ,etWid,et*ount?@	Return the count o+ ' id,et in co(u " n
Strin , ,etWid,et?int inde0@	Return the ' id,et id ' ith the ,iven inde0
void addWid,et?String, ' id,etId@	Add a ne ' ' id,et ' ith the ,iven id
void addWid,et?int inde0@ Strin, ' id,etId@	Add a ne ' ' id,et at ,iven inde0
void reorderWid,et?int inde0@ Strin, ' id,etId@	U&date the inde0 o+ ' id,et in co(u " n

3.26 DataExporter

DataExporter is handy for exporting data (inserted in a table) to various formats such as Excel and CSV and so on.

Info

9a,	dataExporter
9a, *(a	org.primefaces.component.export.DataExporterTag
Action(s) tener *(a	org.primefaces.component.export.DataExporter

Attributes

Name	Default	Type	Description
type	null	String	Export type (HTML, PDF, XLS, XML, CSV, etc.)
target	null	String	Id of the database to hold the data to export.
fileName	null	String	#(name) generated export file name.
forceDownload	false	Boolean	Force download of current page instead of holding the data.
fileType	null	String	File type or format of the exported document.
encoding	UTF-8	String	Character encoding to use.
enableScript	false	Boolean	When enabled, the script will be executed.

Getting Started with DataExporter

DataExporter is used in a UI component and contains such components as column and button or components and so on. #or #and #d+ export, **itext** and #or #O(export, **poi** library are required in the classpath. #ar, et "use it to point to a !#ri " e#ace Database. A #u" e the table to be exported is defined as follows:

```
<p: dataTable id="tableId" ...>
    //columns
</p: dataTable>
```

Excel export

```
<p: commandButton value="Export as Excel" ajax="false">
    <p: dataExporter type="xls" target="tableId" fileName="cars"/>
</p: commandButton>
```

PDF export

```
<p:commandButton value="Export as PDF" ajax="false" >
    <p:dataExporter type="pdf" target="tableId" fileName="cars"/>
</p:commandButton>
```

CSV export

```
<p:commandButton value="Export as CSV" ajax="false" >
    <p:dataExporter type="csv" target="tableId" fileName="cars"/>
</p:commandButton>
```

XML export

```
<p:commandButton value="Export as XML" ajax="false" >
    <p:dataExporter type="xml" target="tableId" fileName="cars"/>
</p:commandLink>
```

PageOnly

If you want to export only the data from the current page, set the `pageOnly` option to `true`.

```
<p:dataExporter type="pdf" target="tableId" fileName="cars" pageOnly="true"/>
```

Excluding Columns

In case you need one or more columns to be ignored, set the `exportable` option to `false`.

```
<p:column exportable="false">
    //...
</p:column>
```

Monitor Status

DataExporters in a non-functional mode or a status component cannot do this. See #include 'nloadonitor Status' section to find out how monitor export mode works. Same option applies to data export as well.

Pre and Post Processors

Processors or are handy to run before the sorted document is added to the collection. Preprocessors or are executed before the data is sorted and postprocessors or are run after data is included in the document. Processors or are in the /avaprocessor method taking the document as a parameter.

*handleHeader

#include "header" or change the background color of the sorted header.

```
<h:commandButton value="Export as XLS">
    <p:dataExporter type="xls" target="tableId" fileName="cars"
                    postProcessor="#{bean.postProcessXLS}"/>
</h:commandButton>
```

```
public void postProcessXLS(Object document) {
    HSSFWorkbook wb = (HSSFWorkbook) document;
    HSSFSheet sheet = wb.getSheetAt(0);
    HSSFRow header = sheet.getRow(0);
    HSSFCellStyle cellStyle = wb.createCellStyle();
    cellStyle.setFillForegroundColor(HSSFColor.GREEN.index);
    cellStyle.setFillPattern(HSSFCellStyle.SOLID_FOREGROUND);

    for(int i=0; i < header.getPhysicalNumberOfCells();i++) {
        header.getCell(i).setCellStyle(cellStyle);
    }
}
```

Add 60.o to !D#

9hi e0a " &(e add a (o,o to the !D# before e0&ortin, be,in .

```
<h:commandButton value="Export as PDF">
    <p:dataExporter type="pdf" target="tableId" fileName="cars"
                    preProcessor="#{bean.preProcessPDF}"/>
</h:commandButton>
```

```
public void preProcessPDF(Object document) throws IOException,
    BadElementException, DocumentException {
    Document pdf = (Document) document;
    ServletContext servletContext = (ServletContext)
FacesContext.getCurrentInstance().getExternalContext().getContext();
    String logo = servletContext.getRealPath("") + File.separator + "images" +
File.separator + "prime_logo.png";
    pdf.add(Image.getInstance(logo));
}
```

3.27 DataGrid

DataGrid di &ay a co((ection o+ data in a ,rid (ayout.

(1 of 5)			<<	<<	1	2	3	4	5	>>	>>	12	▼
f06fb44c	d56c3fa8	3e9f738a											
 1965	 1960	 2009											
78353e38	47d7d8bb	47226f48											
 1975	 2006	 1961											
c3e711f5	6132cae4	45d6db2d											
 1977	 1991	 1994											
8d324378	99a87d39	dd845f93											
 1990	 1977	 1995											
(1 of 5)			<<	<<	1	2	3	4	5	>>	>>	12	▼

Info

9a,	dataGrid
*o " &onent *(a	org.primefaces.component.datagrid.DataGrid
*o " &onent 9y&e	org.primefaces.component.DataGrid
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DataGridRenderer
Renderer *(a	org.primefaces.component.datagrid.DataGridRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	boolean	Boolean value to specify the rendering of the component. If false, the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.

Name	Default	Type	Description
value	null	Object	Data to display.
var	null	String,	 the reuse the indexed variable to refer each data.
rowIndex	null	Integer	<u href="#" to display the row index.
first	0	Integer	Index of the first row to be displayed.
id,etc>ar	null	String,	<a href="#" to the client identifier.
co(u " n)	Integer	<u href="#" to co(u " n in ,rid.
&animation	+active	boolean	Enable animation.
&animation9e "&(ate	null	String,	9e "&(ate of the &animation.
rowIndex!er!a,e9e "&(ate	null	String,	9e "&(ate of the rowIndex!a,e dro&do ' n.
currentRow, eReport9e "&(ate	null	String,	9e "&(ate of the currentRow, eReport UI.
&e6in3	10	Integer	8��i " u " nu " ber of &e (in3 to display.
&animation!option	both	String,	!option of the &animation.
&animationA('ay >i ib(e	true	Boolean	Define if &animation should be hidden if total data count is less than nu " ber of rows &er &e.
type	null	String,	Defines type of the data, rid.
type*(a	null	String,	Style class of the data, rid.
rowIndexIterator	null	String,	<a href="#" to the iterator to refer each rowIndex iterator.
(a=y	+active	Boolean	Defines if (a=y (oadin, i enable(ed for the data column).
e" &type8e a,e	<no record found.	String,	9e0t to display when there is no data to display.

Getting started with the DataGrid

A (i t o+ car ' i((be used through the data, rid) data(i t and datab(e e0a " &(e .

```
public class Car {

    private String model;
    private int year;
    private String manufacturer;
    private String color;
    ...

}
```

The code for *ar1ean that ' ou(d be used to bind the data, rid to the car (i t.

```
public class CarBean {

    private List<Car> cars;

    public CarBean() {
        cars = new ArrayList<Car>();
        cars.add(new Car("myModel", 2005, "ManufacturerX", "blue"));
        //add more cars
    }

    public List<Car> getCars() {
        return cars;
    }
}
```

```
<p:dataGrid var="car" value="#{carBean.cars}" columns="3" rows="12">

    <p:column>
        <p:panel header="#{car.model}">
            <h:panelGrid columns="1">
                <p:graphicImage value="/images/cars/#{car.manufacturer}.jpg"/>

                <h:outputText value="#{car.year}" />
            </h:panelGrid>
        </p:panel>
    </p:column>

</p:dataGrid>
```

9hi data,rid ha) co(u "n and 1% ro' . A data,rid e0tend +ro" standard UIDataE ro' corre &ond to the nu "ber o+ data to di &(ay not the nu "ber o+ ro' to render o the actua(nu "ber o+ ro' to render i ro' lco(u "n P -. A a re u(t data,rid i di &(ayed a K

 5a0e3ce6 1978	 c0a66869 1991	 cd25ac27 1991
 68d039c4 1992	 0c2874f1 1992	 0a32e04e 2002
 518a6446 2009	 be52e4d7 1969	 6192c9e2 1987
 c2e29105 1992	 957c4405 2008	 b3b3cbe8 1983

Ajax Pagination

DataGrid has a built-in paginator that is enabled by setting the `paginator` attribute to true.

```
<p:dataGrid var="car" value="#{carBean.cars}" columns="3" rows="12"
    paginator="true">
    ...
</p:dataGrid>
```

Paginator Template

The paginator is composed of several components that accept various UI controls. Note that this section applies to dataGrid and dataTable.

#ir t!a,e6in3
 6a t!a,e6in3
 !reviou !a,e6in3
 <e0t!a,e6in3
 !a,e6in3
 *urrent!a,eRe&ort
 Ro' !er!a,eDro&do'n

Note that Ro' !er!a,eDro&do'n has it' o'n te" &(at) o&tion to di &(ay i &rovided via ro' !er!a,e9e" &(at) attribute ?e.,.. ro' !er!a,e9e" &(at)PI9E1%E15I@.

Current!a,eRe&ortW has it' o'n te" &(at) defined 'ith current!a,eRe&ort9e" &(at) o&tion. You can use current!a,eWEVtota(!a,e WEVtota(Record WEV tartRecordWEVendRecordW 3ey ' ord ' ithin current!a,eRe&ort9e" &(at). Default i Vcurrent!a,eW o+Vtota(!a,e W. Default UI i K

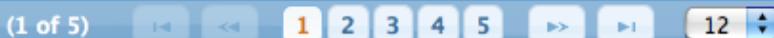


which corresponds to the following template:

IV#ir t!a,e6in3W V!reviou !a,e6in3W V!a,e6in3 W V<e0t!a,e6in3W V6a t!a,e6in3WI

. here are " ore e0a" &(e ba ed on di+erent te" &(at) K

V*urrent!a,eRe&ortW V#ir t!a,e6in3W V!reviou !a,e6in3W V!a,e6in3 W V<e0t!a,e6in3W V6a t!a,e6in3W VRo' !er!a,eDro&do'nWI



V!reviou !a,e6in3W V*urrent!a,eRe&ortW V<e0t!a,e6in3WI



Paginator Position

The paginator can be positioned using the `paginatorPosition` attribute in three different locations: Top, Bottom, or Both.

Selecting Data

Selection of data displayed in dataGrid is very similar to selection in database. You can access the current data using the var reference. There is an event listener on the row to detect data changes. The row data and its index within a dialog are available.

```
<h:form id="carForm">

    <p:dataGrid var="car" value="#{carBean.cars}" columns="3" rows="12">
        <p:panel header="#{car.model}">
            <p:commandLink update=":carForm:display" oncomplete="PF('dlg').show()">
                <f:setPropertyActionListener value="#{car}" target="#{carBean.selectedCar}" />
                <h:outputText value="#{car.model}" />
            </p:commandLink>
        </p:panel>
    </p:dataGrid>

    <p:dialog modal="true" widgetVar="dlg">
        <h:panelGrid id="display" columns="2">
            <f:facet name="header">
                <p:graphicImage value="/images/cars/#{car.manufacturer}.jpg"/>
            </f:facet>
            <h:outputText value="Model:</h:outputText>
            <h:outputText value="#{carBean.selectedCar.year}" />
            //more selectedCar properties
        </h:panelGrid>
    </p:dialog>
</h:form>
```

```
public class CarBean {
    private List<Car> cars;
    private Car selectedCar;
    //getters and setters
}
```

Ajax Behavior Events

Event	Listener Parameter	Fired
onRowSelect	or, onRowSelect="e+ace .event.data. !a, eEvent"	When a row is selected.

```
<p:dataGrid var="car" value="#{carBean.model}">
    <p:ajax event="page" update="anothercomponent" />
    //content
</p:dataGrid>
```

Client Side API

Wid, etⁿ PrimeFaces.widget.DataGrid

Method	Params	Return Type	Description
, et !a, inator?@	B	!a, inator	Return the &a, inator ' id, et.

Skinning

DataGrid re ide in a "ain div container ' hich ty(e and ty(e*(a attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.uiBdata, rid	8 ain container e(e " ent
.uiBdata, ridBcontent	*ontent container.
.uiBdata, ridBdata	9ab(e e(e " ent containin , data
.uiBdata, ridBro '	A ro ' in ,rid
.uiBdata, ridBco(u " n	A co(u " n in ,rid

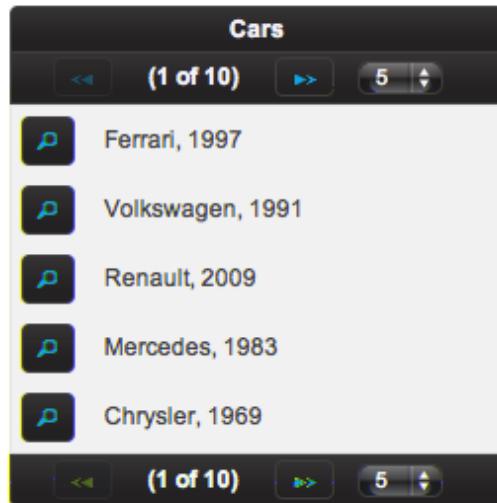
A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or " ore in+or" ation.

Tips

- DataGrid u&&ort (a=y (oadin, data via 6a=yData 8 ode(E ee Data9ab(e (a=y (oadin, ection.
- DataGrid &rovide t ' o +acet na " ed header and footer that you can u e to &rovide cu to " content at the e (ocation .

3.28 DataList

DataList is a component that displays data in a layout with several different types.



Info

9a,	dataList
*o " &onent *(a	org.primefaces.component.datalist.DataList
*o " &onent 9y&e	org.primefaces.component.DataList.DataListTag
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DataListRenderer
Renderer *(a	org.primefaces.component.datalist.DataListRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression that binds to a server-side UI component in a backing bean.
value	null	Object	Data to display.
var	null	String	A variable that receives the received value for each data item.
row	null	Integer	The row index to display.
size	0	Integer	The size of the list to be displayed.

Name	Default	Type	Description
type	unordered	String,	Specifies whether the items in the list are unordered or ordered. If set to 'none', the list will not be rendered.
ite"9y&e	nu((String,	Specifies the iteration expression for the list item.
'id,et>ar	nu((String,	<a " e o+ the c(ient ide ' id,et.
&a,inator	+a(e	boolean	Enables &amination.
&a,inator9e "&(ate	nu((String,	9e " &(ate o+ the &a,inator.
ro'!er!a,e9e" &(ate	nu((String,	9e " &(ate o+ the ro'!er!a,e dro&do ' n.
current!a,eRe&ort9e" &(ate	nu((String,	9e " &(ate o+ the current!a,eRe&ort UI.
&a,e6in3	10	Integer	8 a0i " u " nu " ber o+ &a,e (in3 to di &(ay.
&a,inator!o ition	both	String,	!o ition o+ the &a,inator.
&a,inatorA('ay>i ib(e	true	boolean	Defines if the &a,inator should be hidden if total data count is less than number of rows per &a,e.
ty(e	nu((String,	In-line ty(e o+ the "ain container.
ty(e*(a	<u((String,	Sty(e c(a o+ the "ain container.
ro'Inde0>ar	nu((String,	<a " e o+ the iterator to render each ro' inde0.
varStatu	nu((String,	<a " e o+ the sorted reCue t co&red variab(e to re&re ent state o+ the iteration a"ea in uiNre&eat varStatu .
(a=y	+a(e	boolean	Defines if (a=y (oadin, i enab(ed +or the data co "onent.
e" &ty 8 e a,e	<o record+ound.	String,	9e0t to di &(ay ' hen there is no data to di &(ay.
ite" Sty(e*(a	nu((String,	Sty(e c(a o+ an ite" in (i t.

Getting started with the DataList

Since DataList is a data iteration component, it renders its children for each data item rendered with its var option. See ite"9y&e section for more information about the &o ib(e value.

```
<p: dataList value="#{carBean.cars}" var="car" itemType="disc">
    #{car.manufacturer}, #{car.year}
</p: dataList>
```

Ordered Lists

DataList displays the data in an ordered or flat way depending on the type option. Set type to 'ordered'.

```
<p:dataList value="#{carBean.cars}" var="car" type="ordered">
    #{car.manufacturer}, #{car.year}
</p:dataList>
```

Item Type

itemType defines the bucket type of each item. For ordered items in addition to column ordering decimal types such as "integer" are available.

B A

B a

B i

And for unordered items available values are:

B d c

B circ(e

B Cuare

Definition Lists

Third type of dataList items definition (items that display inline descriptions for each item) is to use definition items (set type option to "definition"). Detailed content is provided with the facet called "description".

```
<p:dataList value="#{carBean.cars}" var="car" type="definition">
    Model: #{car.model}, Year: #{car.year}
    <f:facet name="description">
        <p:graphicImage value="/images/cars/#{car.manufacturer}.jpg"/>
    </f:facet>
</p:dataList>
```



Ajax Pagination

DataList has a built-in paginator that is enabled by setting the paginator option to true.

```
<p:dataList value="#{carBean.cars}" var="car" paginator="true" rows="10">
    #{car.manufacturer}, #{car.year}
</p:dataList>
```

Ajax pagination configuration and using a "dataProvider" configuration section in dataGrid documentation or "more information" and see also "here".

Selecting Data

Data selection can be initiated at any point in dataGrid by selecting, data selection in dataGrid document or more in or action and an event "on".

Client Side API

Wid, et PrimeFaces.widget.DataList

Method	Params	Return Type	Description
, et!a, inator?@	B	!a, inator	Return the &a, inator ' id, et.

Skinning

Data items are rendered in a main div container which has type and type*(attribute) attribute added(y. #o((o ' in, in the item structure(type class e K

Class	Applies
.ui-data-item	Each item in the main container element
.ui-data-item-content	* content container
.ui-data-item-data	Data container
.ui-data-item-item	Each item in (item)

A skinning, type class are , (obliged to be the main item, selection or more in or action.

Tips

- Data items are rendered via `data-item` and `data-item-content` elements. You can override these via `header` and `footer` attributes.
- If you need control over item types "arbitrary" and type to "none". With this setting, the `data-item` does not render item type, (i.e. it and behave like ui-repeater.
- Data items provide the option to add `header` and `footer` that you can use to provide custom content at the location .

3.29 DataScroller

DataScroller di & (ay a co (ection o+ data ' ith on de " and (oadin, u in, cro (in, .

The screenshot shows a vertical scrollable list of car details. At the top, there is a button labeled "Scroll Down to Load More Cars". Below it, four car entries are listed, each with a small manufacturer logo and three data fields: Model, Year, and Color.

- Ford**: Model: f9f60f05, Year: 2003, Color: Maroon
- Honda**: Model: 5c5fd121, Year: 1997, Color: Yellow
- Mercedes-Benz**: Model: ace80ce4, Year: 2004, Color: Maroon
- Jaguar**: Model: 73e6a125, Year: 1999, Color: White

A green vertical scroll bar is visible on the right side of the list, indicating that more content can be viewed by scrolling down.

Info

9a,	dataScroller
*o " &onent *(a	org.primefaces.component.datascroller.DataScroller
*o " &onent 9y&e	org.primefaces.component.DataScroller
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DataScrollerRenderer
Renderer *(a	org.primefaces.component.datascroller.DataScrollerRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	True (value to identify the rendering, or the component is hidden and set to false the component will not be rendered).
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean

Name	Default	Type	Description
value	null	Object	Data to display.
var	null	String	 more content
rowIndex	0	Integer	Index of the iterator to refer each row index.
clientId>ar	null	String	 more content
tye	null	String	Inline type of the main container.
tye*(a	<u href="#{car.manufacturer}" value="#{car.manufacturer}"/> more content	String	Style of the main container.
chunkSize	0	int	 more content
rowIndexVar	null	String	 more content
"ode	document	String	Define the target to (i.e. for example) events where value are Idocsument and inline.
cro(. ei , ht	null	String	Defines & IOException height of the view port in inline mode.
(a=y	+able	boolean	Defines if (a) valid, i.e. enabled or the data contains an error.
buffer	10	Integer	Percentage of height of the buffer between the bottom " of the page and the scroll position to initiate the load of the next chunk. Default is 10% of the page height after 90% of the view port is scrolled down.

Getting started with the DataScroller

DataScroller requires a collection of data to display when the user interacts with the scroll bar and the scroll event triggers the data retrieval from the server. It does this by retrieving the next chunk of data and appending it at the bottom of the page.

```
<p:datascroller value="#{carBean.cars}" var="car" chunkSize="10">
    #{car.manufacturer}
    //more content
</p:datascroller>
```

Scroll Mode

Default scroll mode is 'horizontal' which monitors the scroll bar and triggers the scroll event. Other modes include vertical and mousewheel. The scroll mode option is used to control how the scroll event is triggered. The scroll mode and scroll event attributes are [document] and [inline]. Document mode refers to default and inline mode refers to inline events.

```
<p:dataScroller value="#{carBean.cars}" var="car" mode="inline" chunkSize="10">
    #{car.manufacturer}
    //more content
</p:dataList>
```

Loader

In case of creating a UI element such a button can be defined as the header so that new data is loaded when the header button is clicked. Loader component is defined using `Iheader` facet.

```
<p:dataScroller value="#{carBean.cars}" var="car" mode="inline" chunkSize="10">
    #{car.manufacturer}
    //more content
    <f:facet name="loader">
        <p:commandButton type="button" value="View More" />
    </f:facet>
</p:dataList>
```

Lazy Loading

Lazy loading is enabled by enabling the `lazy` attribute and providing a `LazyDataModel` instance as the value. Refer to `LazyModel` or `DataModel` for more about lazy loading.

```
<p:dataScroller value="#{carBean.lazyModel}" var="car" lazy="true">
    #{car.manufacturer}
    //more content
</p:dataList>
```

Header

Header or the component is defined using `header` facet.

```
<p:dataScroller value="#{carBean.lazyModel}" var="car">
    <f:facet name="header">Cars</f:facet>
    #{car.manufacturer}
    //more content
</p:dataList>
```

Client Side API

Widget `PrimeFaces.widget.DataScroller`

Method	Params	Return Type	Description
<code>load()</code>		<code>void</code>	Load the next chunk.

Skinning

DataScro((er re ide in a "ain div container 'hich ty(e and ty(e*(a attribute a&&(y. #o((o ' in, i the(i t o+ tructura(ty(e c(a e K

Class	Applies
.uiBdata cro((er	8 ain container e(e " ent
.uiBdata cro((erBin(ine	8 ain container e(e " ent in in(ine " ode
.uiBdata(i tBheader	. eader e(e " ent
.uiBdata(i tBcontent	*ontent e(e " ent
.uiBdata cro((erB(i t	6i t e(e " ent container
.uiBdata cro((erBite "	*ontainer o+ each ite " in the (i t
.uiBdata cro((erB(oader	*ontainer o+ cu to " (oader e(e " ent.
.uiBdata cro((erB(oadin,	1 ui(tBn (oad tatu indicator

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or "ore in+or "ation.

3.30 DataTable

DataTable di &ay data in tabular or at.

List of Cars			
Model	Year	Manufacturer	Color
fe1c52bb	1968	Volkswagen	Orange
4f95b4f8	1993	BMW	Red
cb9d1d28	2005	Opel	Green
e297b6eb	1980	Volvo	Silver
c67450c1	1971	Volkswagen	Brown
669699ae	2000	Jaguar	Blue
d632787b	1965	Ford	Silver
5d8b24ce	1988	Jaguar	Maroon
814ef317	1970	BMW	Red
d1b57e10	1970	Ford	Red

Info

9a,	dataTable
*o " &onent *(a	org.primefaces.component.datatable.DataTable
*o " &onent 9y&e	org.primefaces.component.DataTable
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DataTableRenderer
Renderer *(a	org.primefaces.component.datatable.DataTableRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	boolean	Boolean value to specify the rendering of the component when set to false the component will not be rendered.
binding	null	Object	An expression that adds to a server-side UIComponent instance in a backing bean.
value	null	Object	Data to display.
var	null	String	 the reuse tag coded variable used to refer each data.
row	null	Integer	<u href="#" to display the row.
first	0	Integer	Index of the first row to be displayed.

Name	Default	Type	Description
' id, et>ar	null	String,	<a " e o+ the c(ient ide ' id,et.
&a,inator	+a(e	Boolean	Enab(e &a,inator.
&a,inator9e "&(ate	null	String,	9e "&(ate o+ the &a,inator.
ro' !er!a,e9e "&(ate	null	String,	9e "&(ate o+ the ro' !er!a,e dro&do ' n.
ro' !er!a,e6abe(null	String,	6abe(+or the ro' !er!a,e dro&do ' n.
current!a,eRe&ort9e "&(ate	null	String,	9e "&(ate o+ the current!a,eRe&ort UI.
&a,e6in3	10	Integer	8 a0i " u " nu " ber o+ &a,e (in3 to di &(ay.
&a,inator!o ition	both	String,	!o ition o+ the &a,inator.
&a,inatorA(' ay >i ib(e	true	Boolean	De+ine i+ &a,inator hou(d be hidden i+ tota(data count i (e than nu " ber o+ ro' &er &a,e.
cro((ab(e	+a(e	Boolean	8 a3e data cro((ab(e ' ith +i0ed header.
cro(. ei,ht	null	Integer	Scro((vie ' &ort hei,ht.
cro((Width	null	Integer	Scro((vie ' &ort ' idth.
e(ection 8 ode	null	String,	Enab(e ro' e(ectionE va(id va(ue are " in,(e" and " " u(ti&(e".
e(ection	null	Object	Re+erence to the e(ection data.
ro' Inde0>ar	null	String,	<a " e o+ iterator to re+er each ro' inde0.
e" &ty 8 e a,e	<o record +ound.	String,	9e0t to di &(ay ' hen there i no data to di &(ay. A(ternative i e" &ty 8 e a,e +acet.
ty(e	null	String,	In(ine ty(e o+ the co " &onent.
ty(e*(a	null	String,	Sty(e c(a o+ the co " &onent.
db(*ic3Se(ect	+a(e	Boolean	Enab(e ro' e(ection on doub(e c(ic3.
(iveScro((+a(e	Boolean	Enab(e (ive cro((in,..
ro' Sty(e*(a	null	String,	Sty(e c(a +or each ro' .
onE0&andStart	null	String,	*(ient ide ca((bac3 to e0ecute be+ore e0&an ion.
re i=ab(e*o(u" n	+a(e	Boolean	Enab(e co(u" n re i=in,..
ort1y	null	Object	!ro&erty to be u ed +or de+au(t ortin,..
ort7rder	a cendin,	String,	"a cendin," or "de cendin,".
cro((Ro'	0	Integer	<u " ber o+ ro' to (oad on (ive cro((.
ro' ;ey	null	String,	UniCue identi+ier o+ a ro' .
tab(eSty(e	null	String,	In(ine ty(e o+ the tab(e e(e" ent.

Name	Default	Type	Description
tab(eStyle*(a	null	String,	Style to apply to the tab(e element.
+interEvent	MouseEvent	String,	Event to invoke when an interaction occurs.
+interDelay	000	Integer	Delay in milliseconds before sending an enable query.
drag, ,ab(e*o(u " n	+able	Boolean	*o(u " n can be reordered with drag, drop and enable.
editable	+able	Boolean	*ontrue(once(editin,,.
(a=y	+able	Boolean	*ontrue((a=y (oadin,,.
+intered>a(ue	null	String	Text to see when interred data.
rowMode	single	String,	Define row mode, "mode values are single and multiple.
editMode	row	String,	Define edit mode values are row and cell.
editInRow	+able	Boolean	Define if cell editor mode is enabled or not.
separator	null	String,	Separator text to use in output mode or edit mode.
u " " ary	null	String,	Space " " attribute or Whitespace.
+rowEvenRow	null	Boolean	*option to define a paired row.
dir	(ltr	String,	Define text direction values are ltr and rtl.
(iveRe i=e	+able	Boolean	*o(u " n are rendered (ive in this mode without using a real header.
stickyHeader	+able	Boolean	Sticky header stay in 'indoor' view & mode during scroll.
evenOddRow	+able	Boolean	Define if rows are rendered even by default.
disabledSelection	+able	Boolean	Disable selection when true.
rowSelectionMode	none	String,	Define row selection mode "mode or "use(t) for selection.
rowEvenOddSelectionMode	none	String,	Define row even odd selection mode "mode values are I in,(eI and I " use(t) for selection.
dataLocality	null	Boolean	Locality to be used in feature such as interactivity, and row mode.
nativeElement	+able	Boolean	Use native radio checkbox element or row selection.
+rowEven*o(u " n	null	Integer	<u " ber o+ co(u " n to +ree=enable" start index 0.

Name	Default	Type	Description
dra, ,ab(eRo '	+a(e	1oo(ean	When enab(ed) ro ' can be reordered u in, dra,dro&.
ca eSen itiveSort	+a(e	1oo(ean	*a e en itivity +or ortin, E in en itive by de+au(t).
3i&*hi(dren	+a(e	1oo(ean	I, nore &roce in, o+ chi(dren durin, (i+ecyc(eE i " &rove &er+or " ance i+ tab(e on(y ha out&ut co " &onent .

Getting started with the DataTable

We ' i((be u in, the a " e *ar and *ar1ean c(a e de cribed in DataGrid ection.

```
<p:DataTable var="car" value="#{carBean.cars}">
    <p:column>
        <f:facet name="header">
            <h:outputText value="Model" />
        </f:facet>
        <h:outputText value="#{car.model}" />
    </p:column>
    //more columns
</p:DataTable>
```

Header and Footer

1oth datatab(e it e(+ and co(u " n can have cu to " content in their header and footer u in, header and footer +acet re &ective(y. A(ternative(y +or co(u " n there are header9e0t and footer9e0t hortcut to di &(ay i " &(e te0t .

List of Cars			
Model	Manufacturer	Color	Year
16c9b7c6	Mercedes	Maroon	1979
de0e4475	Volkswagen	Maroon	1994
d17a0cac	Ford	Black	1998
0db0095d	Ford	Red	1983
c09b2d08	Renault	Red	1962
a5e3c203	Volkswagen	Green	2007
196bd9e9	Ford	White	1994
111db4d2	Ford	Silver	1994
73b17bd0	Volvo	Blue	1973
8 digit code			1960–2010
In total there are 9 cars.			

```

<p:DataTable var="car" value="#{carBean.cars}">
    <f:facet name="header">
        List of Cars
    </f:facet>
    <p:column>
        <f:facet name="header">
            Model
        </f:facet>
        #{car.model}
    <f:facet name="footer">
        8 digit code
    </f:facet>
    </p:column>
    <p:column headerText="Year" footerText="1960-2010">
        #{car.year}
    </p:column>
    //more columns
    <f:facet name="footer">
        In total there are #{fn:length(carBean.cars)} cars.
    </f:facet>
</p:DataTable>

```

Pagination

DataTable has a built-in paginator that is enabled by setting, & a, inator o&tion to true. See &a, ination section in dataGrid documentation for more information about configuration options.

```

<p:DataTable var="car" value="#{carBean.cars}" paginator="true" rows="10">
    //columns
</p:DataTable>

```

Sorting

Defining, sortBy attribute enables automatic sorting, on that particular column.

```

<p:DataTable var="car" value="#{carBean.cars}">
    <p:column sortBy="#{car.model}" headerText="Model">
        <h:outputText value="#{car.model}" />
    </p:column>
    ...more columns
</p:DataTable>

```

In stead of using the default sorting, a comparator which uses a comparator you can define your own sort method as follows.

```

public int sortByModel(Object car1, Object car2) {
    //return -1, 0, 1 if car1 is less than, equal to or greater than car2
}

```

```
<p:DataTable var="car" value="#{carBean.cars}">
    <p:column sortBy="#{car.model}" sortFunction="#{carBean.sortByModel}"
        headerText="Model">
        <h:outputText value="#{car.model}" />
    </p:column>

    ...
</p:DataTable>
```

8 u(ti&e ortin, i enab(ed by ettin, *sortMode* to *multiple*. In thi " odeE c(ic3in, a ort co(u " n ' hi(e " eta3ey i on add ort co(u " n to the order ,rou&.

```
<p:DataTable var="car" value="#{carBean.cars}" sortMode="multiple">
    ...
</p:DataTable>
```

Data9ab(e can di &(ay data orted by de+au(t) to i " &(e " ent thi u e the *sortBy* o&tion o+ datatab(e and the o&tiona(*sortOrder*. 9ab(e be(o ' ' ou(d be initia((y di &(ayed a orted by " ode(.

```
<p:DataTable var="car" value="#{carBean.cars}" sortBy="#{car.model}">

    <p:column sortBy="#{car.model}" headerText="Model">
        <h:outputText value="#{car.model}" />
    </p:column>

    <p:column sortBy="#{car.year}" headerText="Year">
        <h:outputText value="#{car.year}" />
    </p:column>

    ...
</p:DataTable>
```

Filtering

A/a0 ba ed +i(terin, i enab(ed by ettin, *filterBy* at co(u " n (eve(and &rovidin, a (i t to 3ee& the +i(tered ub(i t. It i u ,e ted to u e a co&e (on,er than reCue t (i3e vie ' co&e to 3ee& the +i(tered>a(ue o that +i(tered (i t i ti((acce ib(e a+ter +i(terin,,.

```
<p:DataTable var="car" value="#{carBean.cars}"
    filteredValue="#{carBean.filteredCars}">
    <p:column filterBy="#{car.model}" headerText="Model">
        <h:outputText value="#{car.model}" />
    </p:column>
    ...
</p:DataTable>
```

#i(terin, i tri,,ered ' ith 3eyu& event and +i(ter in&ut can be ty(ed u in, *filterStyle*&
filterStyleClass attribute . I+ you'd (i3e to u e a dro&do ' n in tead o+ an in&ut +ie(d to on(y a((o '&redefined +i(ter va(ue u e *filterOptions* attribute and a co((ection&array o+ e(ectite " a va(ue. In additionE *filterMatchMode* de+ine the bui(tbin " atcher ' hich i *startsWith* by de+au(t.

#o((o ' in, i a ba ic +i(terin, datatab(e ' ith the e o&tion de " on trated

```

<p:dataTable var="car" value="#{carBean.cars}"
    filteredValue="#{carBean.filteredCars}" widgetVar="carsTable">

    <f:facet name="header">
        <p:outputPanel>
            <h:outputText value="Search all fields:" />
            <h:inputText id="globalFilter" onkeyup="PF('carsTable').filter()" />
        </p:outputPanel>
    </f:facet>

    <p:column filterBy="#{car.model}" headerText="Model" filterMatchMode="contains">
        <h:outputText value="#{car.model}" />
    </p:column>

    <p:column filterBy="#{car.year}" headerText="Year" footerText="startsWith">
        <h:outputText value="#{car.year}" />
    </p:column>

    <p:column filterBy="#{car.manufacturer}" headerText="Manufacturer"
        filterOptions="#{carBean.manufacturerOptions}" filterMatchMode="exact">
        <h:outputText value="#{car.manufacturer}" />
    </p:column>

    <p:column filterBy="#{car.color}" headerText="Color" filterMatchMode="endsWith">
        <h:outputText value="#{car.color}" />
    </p:column>

    <p:column filterBy="#{car.price}" headerText="Price" filterMatchMode="exact">
        <h:outputText value="#{car.price}" />
    </p:column>

</p:dataTable>

```

#i(ter (ocated at header i a , (oba(one a&&(yin, on a((+ie(d E thi i i " &(e " ented by ca((in, c(ient ide A ! I " ethod ca((ed *filter()*, i " &ortant &art i to &eci+y the id o+ the in&ut te0t a *globalFilter* ' hich i a re erved identi+ier +or databab(e).

In addition to de+au(t +i(terin, ' ith ,enerated e(e " ent E cu to " e(e " ent can a(o be u ed a a +i(ter +acet. E0a " &(e be(o ' u e cu to " +i(ter co " &onent in co " bination ' ith ,enerated e(e " ent . When a cu to " co " &onent i u ed a a +i(ter +acetE +i(terin, need to be ca((ed " anua(y +ro " a &re+erred event uch a onchan,ePI!#?!!car 9ab(e@.+i(ter?@I. A(o de+inin, a converter " i,ht be nece ary i+ the va(ue o+ the +i(ter +acet i not de+ined.

```

<p:dataTable id="dataTable" var="car" value="#{tableBean.carsSmall}"
    widgetVar="carsTable" filteredValue="#{tableBean.filteredCars}">

    <p:column id="modelColumn" filterBy="#{car.model}" headerText="Model"
        footerText="contains" filterMatchMode="contains">
        <h:outputText value="#{car.model}" />
    </p:column>

    <p:column id="yearColumn" filterBy="#{car.year}" headerText="Year"
        footerText="lte" filterMatchMode="lte">
        <f:facet name="filter">
            <p:spinner onchange="PF('carsTable').filter()" min="1960" max="2010">
                <f:converter converterId="javax.faces.Integer" />
            </p:spinner>
        </f:facet>
        <h:outputText value="#{car.year}" />
    </p:column>

    <p:column id="manufacturerColumn" filterBy="#{car.manufacturer}"
        headerText="Manufacturer" footerText="exact" filterMatchMode="exact">
        <f:facet name="filter">
            <p:selectOneMenu onchange="PF('carsTable').filter()" >
                <f:selectItems value="#{tableBean.manufacturerOptions}" />
            </p:selectOneMenu>
        </f:facet>
        <h:outputText value="#{car.manufacturer}" />
    </p:column>

    <p:column id="colorColumn" filterBy="#{car.color}" headerText="Color"
        footerText="in" filterMatchMode="in">
        <f:facet name="filter">
            <p:selectCheckboxMenu label="Colors"
onchange="PF('carsTable').filter()">
                <f:selectItems value="#{tableBean.colors}" />
            </p:selectCheckboxMenu>
        </f:facet>
        <h:outputText value="#{car.color}" />
    </p:column>

    <p:column id="soldColumn" filterBy="#{car.sold}" headerText="Status"
        footerText="equals" filterMatchMode="equals">
        <f:facet name="filter">
            <p:selectOneButton onchange="PF('carsTable').filter()">
                <f:converter converterId="javax.faces.Boolean" />
                <f:selectItem itemLabel="All" itemValue="" />
                <f:selectItem itemLabel="Sold" itemValue="true" />
                <f:selectItem itemLabel="Sale" itemValue="false" />
            </p:selectOneButton>
        </f:facet>
        <h:outputText value="#{car.sold ? 'Sold' : 'Sale'}" />
    </p:column>
</p:dataTable>

```

+i(ter 8 atch 8 ode de+ine ' hich bui(tBin +i(terin, a(,orith " ' ou(d be u ed &er co(u " nE va(id va(ue +or thi attribute areK

- B **startsWith***hec3 i+ co(u " n va(ue tart ' ith the +i(ter va(ue.
- B **endsWith***hec3 i+ co(u " n va(ue end ' ith the +i(ter va(ue.
- B **contains***hec3 i+ co(u " n va(ue contain the +i(ter va(ue).
- B **exact***hec3 i+ trin, re&re entation o+ co(u " n va(ue and +i(ter va(ue are a " e.
- B **lt***hec3 i+ co(u " n va(ue i (e than the +i(ter va(ue.
- B **lte***hec3 i+ co(u " n va(ue i (e than or eCua(the +i(ter va(ue.
- B **gt***hec3 i+ co(u " n va(ue i ,reater than the +i(ter va(ue.
- B **gte***hec3 i+ co(u " n va(ue i ,reater than or eCua(the +i(ter va(ue.
- B **equals***hec3 i+ co(u " n va(ue eCua(the +i(ter va(ue.
- B **in***hec3 i+ co(u " n va(ue i in the co((ection o+ the +i(ter va(ue.

In ca e the bui(tBin " ethod do not u+iceE cu to " +i(terin, can be i" &(e " ented u in, +i(ter#unction a&&roach.

```
<p:column filterBy="#{car.price}" filterFunction="#{tableBean.filterByPrice}">
    <h:outputText value="#{car.price}">
        <f:convertNumber currencySymbol="$" type="currency"/>
    </h:outputText>
</p:column>
```

filterFunction hou(d be a " ethod ' ith three &ara " eter K co(u " n va(ueE +i(ter va(ue and (oca(e. Return va(ue i a boo(eanE true acce&t the va(ue and +a(e re/ect it.

```
public boolean filterByPrice(Object value, Object filter, Locale locale) {
    //return true or false
}
```

6oca(e i &rovided a o&tiona(in ca e you need to u e a (oca(e a ' are " ethod (i3e *toLowerCase(Locale locale)*). <ote that Strin, ba ed +i(ter (i3e tart WithE end With u e to6o ' er*a e a(ready and *dataLocale* attribute i u ed to &rovide the (oca(e to u e ' hen +i(terin,..

Row Selection

9here are evera(' ay to e(ect ro ' ? @+ro " datatab(e. 6et' be,in by addin, a *ar re+erence +or in,(e e(ection and a *ar array +or " u(ti&(e e(ection to the *ar1ean to ho(d the e(ectioned data.

```
public class CarBean {
    private List<Car> cars;
    private Car selectedCar;
    private Car[] selectedCars;
    public CarBean() {
        cars = new ArrayList<Car>();
        //populate cars
    }
    //getters and setters
}
```

Sin,(e Se(ection ' ith a *o " "and *o "&onent

9hi " ethod i i" &(e " ented ' ith a co " "and co " &onent uch a co " "and6in3 or co " "and1utton. Se(ectioned ro ' can be et to a erver ide in tance by &a in, a a &ara " eter i+ you are u in, E6 %.% or u in, +N et!ro&ertyAction6i tener.

```
<p:DataTable var="car" value="#{carBean.cars}">

    <p:column>
        <p:commandButton value="Select">
            <f:setPropertyActionListener value="#{car}" target="#{carBean.selectedCar}" />
        </p:commandButton>
    </p:column>

    ...columns

</p:DataTable>
```

Single Selection with Radio Buttons

Reviewing the code above, you'll notice that the button inside the `commandButton` tag has a `target` attribute set to `#{carBean.selectedCar}`. This is to enable selection wherever the row is selected. If you want to enable selection on the entire row, you can use the `selectionMode` option.

```
<p:DataTable var="car" value="#{carBean.cars}" selectionMode="single"
    selection="#{carBean.selectedCar}" rowKey="#{car.id}">
    ...columns
</p:DataTable>
```

Multiple Selection with Radio Buttons

If you want to enable selection on each row individually, you can do so by setting the `selectionMode` attribute to `"single"`. In order to enable this feature, define a column with `selectionMode="single"`.

```
<p:DataTable var="car" value="#{carBean.cars}" selectionMode="multiple"
    selection="#{carBean.selectedCars}" rowKey="#{car.id}">
    ...columns
</p:DataTable>
```

Single Selection with RadioButton

Selection of a row with a radio button is done on each row in a column. On each row, there is a data table header built-in. If you want to enable selection on the entire row, you can do so by setting the `selectionMode` attribute to `"single"`. In order to enable this feature, define a column with `selectionMode="single"`.

```
<p:DataTable var="car" value="#{carBean.cars}" selection="#{carBean.selectedCar}"
    rowKey="#{car.id}">
    <p:column selectionMode="single"/>
    ...columns
</p:DataTable>
```

Multiple Selection with Checkboxes

If you want to enable selection on each row individually, you can do so by setting the `selectionMode` attribute to `"multiple"`. In order to enable this feature, define a column with `selectionMode="multiple"`.

```
<p:DataTable var="car" value="#{carBean.cars}" selection="#{carBean.selectedCars}"
    rowKey="#{car.id}">
    <p:column selectionMode="multiple"/>
    ...columns
</p:DataTable>
```

J U e **rowSelectMode** o&tion to cu to " i=e the de+au(t behavior on ro ' c(ic3 o+ a " u(ti&(e e(ection enab(ed datatab(e. De+au(t va(ue i Ine ' I that c(ear &reviou e(ection and IaddI " ode 3ee& &reviou e(ection a" e a e(ectin, a ro ' ' ith " ou e c(ic3 ' hen " eta3ey i on.

RowKey

Ro ' ; ey hou(d a uniCue identi+ier +ro" your data "ode(and u ed by datatab(e to +ind the e(ection ro ' . Mou can either de+ine thi 3ey by u in , the ro ' ; ey attribute or by bindin , a data "ode(' hich i " &(e " ent *org.primefaces.model.SelectableDataModel*.

Dynamic Columns

Dyna" ic co(u "n i handy in ca e you can't 3no ' ho ' "any co(u "n to render. *o(u "n co " &onent i u ed to de+ine the co(u "n &ro,ra " " atica((y. It reCuire a co((ection a the va(ueE t ' o iterator variab(e ca((ed *var* and *columnIndexVar*.

```
<p:DataTable var="cars" value="#{tableBean.cars}">
    <p:columns value="#{tableBean.columns}" var="column"
        sortBy="#{column.property}" filterBy="#{column.property}">
            <f:facet name="header">
                #{column.header}
            </f:facet>
            <h:outputText value="#{cars[column.property]}" />
        </p:columns>
</p:DataTable>
```

```

public class CarBean {

    private List<ColumnModel> columns = new ArrayList<ColumnModel>();
    private List<Car> cars;

    public CarBean() {
        populateColumns();
        cars = //populate cars;
    }

    public void populateColumns() {
        String[] columnKeys = new String[]{"model", "year", "color"};

        for(String columnKey : columnKeys) {
            columns.add(new ColumnModel(columnKey.toUpperCase(), columnKey));
        }
    }

    //getters and setters

    static public class ColumnModel implements Serializable {

        private String header;
        private String property;

        public ColumnModel(String header, String property) {
            this.header = header;
            this.property = property;
        }

        public String getHeader() {
            return header;
        }

        public String getProperty() {
            return property;
        }
    }
}

```

Grouping

Grouping is defined by using "group" &onent used to define header and footer .

Manufacturer	Sales			
	Sales Count		Profit	
	Last Year	This Year	Last Year	This Year
Mercedes	90%	8%	28031\$	25102\$
BMW	14%	91%	18640\$	28023\$
Volvo	82%	24%	130\$	77724\$
Audi	7%	40%	2272\$	33672\$
Renault	10%	54%	98115\$	40664\$
Opel	63%	28%	10549\$	93746\$
Volkswagen	6.7%	3.8%	38242\$	9035\$
	40%	63%	10146\$	7697\$
	26%	70%	40384\$	62298\$
	14%	94%	96052\$	42233\$
Totals:		342561\$	430222\$	

Chrysler
Ferrari
Ford

```

<p:DataTable var="sale" value="#{carBean.sales}">
    <p:columnGroup type="header">
        <p:row>
            <p:column rowspan="3" headerText="Manufacturer" />
            <p:column colspan="4" headerText="Sales" />
        </p:row>
        <p:row>
            <p:column colspan="2" headerText="Sales Count" />
            <p:column colspan="2" headerText="Profit" />
        </p:row>
        <p:row>
            <p:column headerText="Last Year" />
            <p:column headerText="This Year" />
            <p:column headerText="Last Year" />
            <p:column headerText="This Year" />
        </p:row>
    </p:columnGroup>
    <p:column>
        #{sale.manufacturer}
    </p:column>
    <p:column>
        #{sale.lastYearProfit}%
    </p:column>
    <p:column>
        #{sale.thisYearProfit}%
    </p:column>
    <p:column>
        #{sale.lastYearSale}$
    </p:column>
    <p:column>
        #{sale.thisYearSale}$
    </p:column>
    <p:columnGroup type="footer">
        <p:row>
            <p:column colspan="3" style="text-align:right" footerText="Totals:"/>
            <p:column footerText="#{tableBean.lastYearTotal} $" />
            <p:column footerText="#{tableBean.thisYearTotal} $" />
        </p:row>
    </p:columnGroup>
</p:DataTable>

```

```

public class CarBean {

    private List<Sale> sales;

    public CarBean() {
        sales = //create a list of BrandSale objects
    }

    public List<Sale> getSales() {
        return this.sales;
    }
}

```

Scrolling

Scro(in, " a3e the headerB+oter o+ the tab(e +i0ed and the body &art cro((ab(e. scrollable attribute " u t be enab(ed and de&endin, on the cro((in, orientation scrollHeight and/or scrollWidth o&tion " u t be &eci+iied.

```
<p: dataTable var="car" value="#{bean.data}" scrollable="true" scrollHeight="150">
    <p: column />
    //columns
</p: dataTable>
```

Model	Year	Manufacturer	Color
Model	Year	Year	Year
069794d7	1991	Volvo	Silver
4aeeec6c	1993	Ford	Green
09cbc05c	1983	Chrysler	Maroon
2d374a04	1964	Ferrari	Red
9c09bc54	1987	Volkswagen	Blue
25d45a08	1993	Opel	White

Si "&(e cro((in, render a((data to c(ient ' herea (ive cro((in, i u e+u(to dea(' ith hu,e dataE in thi ca e data i +etched ' henever the cro((bar reache botto ". Set *liveScroll* to enab(e thi o&tionK

```
<p: dataTable var="car" value="#{bean.data}" scrollable="true" scrollHeight="150"
    liveScroll="true">

    <p: column />
    //columns
</p: dataTable>
```

Scro((in, ha) " ode K OE y and OBy cro((in, that are de+ined by *scrollHeight* and *scrollWidth*. 9he e t ' o cro((attribute can be de+ined u in, inte ,er va(ue indicatin , +i0ed &i0e(or &ercenta,e re(ative to the container di "en ion .

Frozen Rows

*ertain ro ' can be +i0ed in a cro((ab(e tab(e by u in, the *frozenRows* attribute that define the nu "ber o+ ro ' to +ree=e +ro " the tart.

```
<p: dataTable var="car" value="#{bean.data}" scrollable="true" scrollHeight="150"
    frozenRows="2">

    <p: column />
    //columns
</p: dataTable>
```

Frozen Columns

S&eci+iic co(u "n can be +i0ed ' hi(e the re t o+ the " re "ain a cro((ab(e. *frozenColumns* de+ine the nu "ber o+ co(u "n to +ree=e +ro " the tart.

```
<p:DataTable var="car" value="#{bean.data}" scrollable="true" scrollWidth="200"
    frozenColumns="2">

    <p:column />
    //columns
</p:DataTable>
```

Expandable Rows

RowToggler and *RowExpansion* +acet are u ed to i " &(e " ent e0&andab(e ro ' .

```
<p:DataTable var="car" value="#{carBean.cars}">

    <f:facet name="header">
        Expand rows to see detailed information
    </f:facet>

    <p:column>
        <p:rowToggler />
    </p:column>

    //columns

    <p:rowExpansion>
        //Detailed content of a car
    </p:rowExpansion>

</p:DataTable>
```

&Nro ' 9o , ,(er co " &onent &(ace an e0&andlco((a& e iconE c(ic3in, on a co((a& ed ro ' (oad e0&anded content ' ith a/a0. I+ you need to di &(ay a ro ' a e0&anded by de+au(tE u e *expandedRow* attribute ' hich i eva(uated be+ore renderin, o+ each ro ' o va(ue e0&re ion are u&&orted. Additional(y ro ' E0&and 8 ode attribute de+ine i+ " u(ti&(e ro ' can be e0&anded at the a " e ti " e or notE va(id va(ue are I in,(eI and I " u(ti&(eI ?de+au(t@.

Expand rows to see detailed information		
	Model	Year
0b8313c2		1976
2be34a8c		1995
08e342c4		2004
b5d03231		1998
<div style="border: 1px solid #ccc; padding: 5px; width: fit-content;">  Model: b5d03231 Year: 1998 Manufacturer: Mercedes Color: Red </div>		
b50b6dc		1974
db39801c		1995
f76c474f		1989
2c9b67a2		2005
94fb553f		1973

Editing

Ince((editin, &rovide an ea y ' ay to di &(ay editab(e data. *p:cellEditor* i u ed to de+ine the ce((editor o+ a &articu(ar co(u " n. 9here are t ' o ty&e o+ editin, *E row* and *cell*. Ro ' editin, i the de+au(t " ode and u ed by addin, a *p:rowEditor* co " &onent a ro ' contro(.

```
<p:DataTable var="car" value="#{carBean.cars}" editable="true">

    <f:facet name="header">
        In-Cell Editing
    </f:facet>

    <p:column headerText="Model">
        <p:cellEditor>
            <f:facet name="output">
                <h:outputText value="#{car.model}" />
            </f:facet>
            <f:facet name="input">
                <h:inputText value="#{car.model}" />
            </f:facet>
        </p:cellEditor>
    </p:column>
    //more columns with cell editors

    <p:column>
        <p:rowEditor />
    </p:column>
</p:DataTable>
```

In-Cell Editing				
Model	Year	Manufacturer	Color	Options
824641ad	1976	Volvo	Yellow	
d859a7ba	1977	Ferrari	Brown	
	9379f6f5	1961	Renault	Silver
	744a8017	1960	Chrysler	Silver
	80feefef	2000	Opel	Yellow
	9e0c7267	1982	Opel	Red
	33124250	1984	Ford	Red
	0349899f	1977	Renault	Red

When the edit icon is clicked, the row is displayed in edit mode, and the edit and cancel buttons are displayed, and the save and cancel buttons are hidden. The edit icon only has that functionality and the cancel icon reverts the changes if both options are selected with a mouse interaction.

Another option is to include editing in the row editor. In this case, it's better to edit the row when it is rendered, so in the render event, an a/a0 event to save the changes.

Lazy Loading

Lazy loading is an approach to deal with huge datasets efficiently by rendering only a portion of the data at a time. It's achieved by rendering a portion of the data into memory, loading the remaining data into memory only when needed. To do this, you need to bind a `org.primefaces.model.LazyDataModel` to the value and implement the `load` method and enable the `lazy` option. Additionally, it's required to implement `getRowData` and `getRowKey` if you have pagination enabled.

```
<p:DataTable var="car" value="#{carBean.model}" paginator="true" rows="10"
    lazy="true">
    //columns
</p:DataTable>
```

```
public class CarBean {

    private LazyDataModel model;

    public CarBean() {
        model = new LazyDataModel() {
            @Override
            public void load(int first, int pageSize, String sortField,
                SortOrder sortOrder, Map<String, Object> filters) {
                //load physical data
            }
        };

        int totalRowCount = //logical row count based on a count query
        model.setRowCount(totalRowCount);
    }

    public LazyDataModel getModel() {
        return model;
    }
}
```

Data9ab(e ca((your (oad i " &(e " entation ' henever a &a,in,E ortin, or +i(terin, occur ' ith
+o((o ' in, &ara " eter K

G +ir tN 7++ et o+ +ir t data to tart +ro "
G &a,eSi=eN <u " ber o+ data to (oad
G ort#ie(dN <a " e o+ ort +ie(d
G ort7rderN Sort7rder enu ".
G +i(terN #i(ter " a& ' ith +ie(d na " e a 3ey ?e.,. I " ode(I +or +i(ter1yPIXVcar." ode(WI@ and va(ue.

In addition to (oad " ethodE tota(Ro ' *ount need to be &rovided o that &a,inator can di &(ay it e(+ accordin , to the (o,ica(nu " ber o+ ro ' to di &(ay.

It i u,,e ted to u e *field* attribute o+ co(u " n co " &onent to de+ine the +ie(d na " e &a ed a ort#ie(d and +i(ter#ie(d E other ' i e the e +ie(d ' ou(d be tried to ,et e0tracted +ro " the va(ue e0&re ion ' hich i not &o ib(e in ca e (i3e co " &o ite co " &onent .

Sticky Header

Stic3y . eader +eature " a3e the databab(e header vi ib(e on &a,e cro((in,.

```
<p:DataTable var="car" value="#{carBean.model}" stickyHeader="true">
    //columns
</p:DataTable>
```

Model	Year	Manufacturer	Color
d975132e	2006	Volvo	White
9479fe68	2002	Opel	White
1aaad80a	2000	Opel	Orange
6082eb65	1965	Audi	Red
359eeebe	1967	Mercedes	White
40a6a578	2006	Ferrari	Red
3c96cad6	1983	Volkswagen	Blue



Model	Year	Manufacturer	Color
9479fe68	2002	Opel	White
1aaad80a	2000	Opel	Orange
6082eb65	1965	Audi	Red
359eeebe	1967	Mercedes	White
40a6a578	2006	Ferrari	Red
3c96cad6	1983	Volkswagen	Blue
2f146e89	2002	Audi	Blue
beaa8f3d	1989	BMW	Blue
90049864	1984	Audi	White
acf9632e	1992	Audi	Yellow
a45a5bfa	1992	Chrysler	White

SummaryRow

Summary rows are header components that can be placed in the body of a table to define summary information for the rows above them. This is typically done by defining the summary row in the body of the table and then defining the summary value in the footer of the table.

Model	Year	Manufacturer	Color
30d423c1	1995	Volvo	Orange
caa74a90	2005	Volvo	White
2295d17b	1996	Volvo	Blue
d9548573	1990	Volvo	Black
3f2fddb1	1979	Volvo	Blue
c9cb10af	2007	Volvo	Maroon
d69007fb	1998	Volvo	Black
Total:			40272\$
986742ea	1966	Volkswagen	Orange
f5045e9a	2006	Volkswagen	Red
3498c563	1994	Volkswagen	Red
Total:			61413\$

```

<p:dataTable var="car" value="#{tableBean.cars}">

    <p:column headerText="Model">
        #{car.model}
    </p:column>

    <p:column headerText="Year" sortBy="year">
        #{car.year}
    </p:column>

    <p:column headerText="Manufacturer" sortBy="manufacturer">
        #{car.manufacturer}
    </p:column>

    <p:column headerText="Color" sortBy="color">
        #{car.color}
    </p:column>

    <p:summaryRow>
        <p:column colspan="3" style="text-align:right">
            Total:
        </p:column>

        <p:column>
            #{tableBean.randomPrice}$
        </p:column>
    </p:summaryRow>
</p:dataTable>

```

SubTable

Sub9ab(e i a he(&er co " &onent to di &(ay ne ted co((ection . E0a " &(e be(o ' di &(ay a co((ection o+ &(ayer and a ubtab(e +or the tat co((ection o+ each &(ayer.

FCB Statistics		
Player	Stats	
	Goals	Assists
Messi		
2005-2006	4	2
2006-2007	10	7
2007-2008	16	10
2008-2009	32	15
2009-2010	51	22
2010-2011	55	30
Totals:	168	86
Xavi		
2005-2006	6	15
2006-2007	10	20
2007-2008	12	22
2008-2009	9	24
2009-2010	8	21
2010-2011	10	25
Totals:	55	127
Iniesta		
2005-2006	4	12
2006-2007	7	9
2007-2008	10	14
2008-2009	15	17
2009-2010	14	16
2010-2011	17	22
Totals:	67	90

```

<p:DataTable var="player" value="#{tableBean.players}">

    <f:facet name="header">
        FCB Statistics
    </f:facet>

    <p:columnGroup type="header">
        <p:row>
            <p:column rowspan="2" headerText="Player" />
            <p:column colspan="2" headerText="Stats" />
        </p:row>

        <p:row>
            <p:column headerText="Goals" />
            <p:column headerText="Assists" />
        </p:row>
    </p:columnGroup>

    <p:subTable var="stats" value="#{player.stats}">
        <f:facet name="header">
            #{player.name}
        </f:facet>

        <p:column>
            #{stats.season}
        </p:column>

        <p:column>
            #{stats.goals}
        </p:column>

        <p:column>
            #{stats.assists}
        </p:column>

        <p:columnGroup type="footer">
            <p:row>
                <p:column footerText="Totals: " style="text-align:right"/>
                <p:column footerText="#{player.allGoals}" />
                <p:column footerText="#{player.allAssists}" />
            </p:row>
        </p:columnGroup>
    </p:subTable>

</p:DataTable>

```

Column Toggler

In addition to a table can be to, , (ed u in, the co(u " n to, ,(er he(&er co " &onent.

```
<p:dataTable var="car" value="#{tableBean.cars}">

    <f:facet name="header">
        List of Cars

        <p:commandButton id="toggler" type="button" value="Columns"
            style="float:right" icon="ui-icon-calculator" />
            <p:columnToggler datasource="cars" trigger="toggler" />
    </f:facet>

    <p:column headerText="Model">
        #{car.model}
    </p:column>

    <p:column headerText="Year" sortBy="year">
        #{car.year}
    </p:column>

    <p:column headerText="Manufacturer" sortBy="manufacturer">
        #{car.manufacturer}
    </p:column>

    <p:column headerText="Color" sortBy="color">
        #{car.color}
    </p:column>

</p:dataTable>
```

List of Cars		
Model	Color	Year
7ec4ef00	Yellow	2007
96b9492d	Yellow	1984
ae9742a4	Orange	1972
e4d0922b	Green	2000
176a7f17	Black	1994
bef3563	Brown	1977
b9a045fd	Black	1960
82cafaf5	Silver	1995
49d09cbd	Silver	1963

The column toggler component allows you to dynamically change the columns displayed in a table. It provides a "Columns" button that opens a dropdown menu where you can select which columns to show. In the example above, the "Year" column is currently selected.

Reordering Rows

Rows in the table can be reordered by dragging and dropping. Set the `rowReorderable` attribute to true to enable this feature.

The row reordering behavior is provided by the `rowReorder` attribute. It triggers a `ReorderEvent` at the server side. If you want to handle this event, you can implement the `org.primefaces.event.ReorderEvent` interface.

```
<p:DataTable var="car" value="#{tableBean.cars}" draggableRows="true">
    <p:ajax event="rowReorder" listener="#{tableBean.onRowReorder}" />
    <p:column headerText="Model">
        #{car.model}
    </p:column>
    //columns
</p:DataTable>
```

```
public class TableBean {
    //...
    public void onRowReorder(ReorderEvent event) {
        //int from = event.getFromIndex();
        //int end = event.getEndIndex();
    }
}
```

Reordering Columns

*o(u " n o+ the tab(e can be reordered u in , dra, : dro&a ' e(. Set dra,,ab(e*o(u " n attribute to true to enab(e thi +eature. 7&tiona(colReorder a/a0 behavior i &rovided to (i ten to reorder event at erver ide.

```
<p:DataTable var="car" value="#{tableBean.cars}" draggableColumns="true">
    <p:column headerText="Model">
        #{car.model}
    </p:column>
    //columns
</p:DataTable>
```

Ajax Behavior Events

Event	Listener Parameter	Fired
&a,e	or,.&ri "e+ace .event.data.!a,eEvent	When a &a, ination.
ort	or,.&ri "e+ace .event.data.SortEvent	When a co(u " n i orted.
+i(ter	or,.&ri "e+ace .event.data.#i(terEvent	When a +i(terin,.
ro ' Se(ect	or,.&ri "e+ace .event.Se(ectEvent	When a ro ' i bein, e(ected.
ro ' Un e(ect	or,.&ri "e+ace .event.Un e(ectEvent	When a ro ' i bein, un e(ected.
ro ' Edit	or,.&ri "e+ace .event.Ro ' EditEvent	When a ro ' i edited.
ro ' EditInit	or,.&ri "e+ace .event.Ro ' EditEvent	When a ro ' itche to edit " ode
ro ' Edit*ance(or,.&ri "e+ace .event.Ro ' EditEvent	When ro ' edit i cance((ed.
co(Re i=e	or,.&ri "e+ace .event.*o(u " nRe i=eEvent	When a co(u " n i bein, e(ected.
to,,(eSe(ect	or,.&ri "e+ace .event.9o,,(eSe(ectEvent	When header chec3bo0 i to,,(ed.
co(Reorder	B	When co(u " n are reordered.
ro ' Se(ectRadio	or,.&ri "e+ace .event.Se(ectEvent	Ro ' e(ection ' ith radio.
ro ' Se(ect*hec3bo0	or,.&ri "e+ace .event.Se(ectEvent	Ro ' e(ection ' ith chec3bo0.
ro ' Un e(ect*hec3bo0	or,.&ri "e+ace .event.Un e(ectEvent	Ro ' un e(ection ' ith chec3bo0.
ro ' Db(e(ect	or,.&ri "e+ace .event.Se(ectEvent	Ro ' e(ection ' ith doub(e c(ic3.
ro ' 9o,,(e	or,.&ri "e+ace .event.9o,,(eEvent	Ro ' e0&and or co((a& e.
conte0t 8 enu	or,.&ri "e+ace .event.Se(ectEvent	*onte0t 8 enu di &(ay.
ce((Edit	or,.&ri "e+ace .event.*e((EditEvent	When a ce((i edited.
ro ' Reorder	or,.&ri "e+ace .event.ReorderEvent	When ro ' reorder.

#or e0a " &(eE datatab(e be(o ' " a3e an a/a0 reCue t ' hen a ro ' i e(ected ' ith a c(ic3 on ro ' .

```
<p:dataTable var="car" value="#{carBean.model}">
    <p:ajax event="rowSelect" update="another_component" />
    //columns
</p:dataTable>
```

Client Side API

Wid, et^N PrimeFaces.widget.DataTable

Method	Params	Return Type	Description
, et!a, inator?@	B	!a, inator	Return the &a, inator in ance.
c(ear#i(ter ?@	B	void	*(ear a((co(u " n +i(ter
, etSe(ectedRo ' *ount?@		<u " ber	Return nu " ber o+ e(ected ro ' .
e(ectRo ' ?rE i(ent@	r\ nu " ber or tr e(e " ent a /Duery ob/ect <i>silent</i> \ +(a, to +ire ro ' e(ect a/a0 behavior	void	Se(ect the ,iven ro ' .
un e(ectRo ' ?rE i(ent@	r\ nu " ber or tr e(e " ent a /Duery ob/ect <i>silent</i> \ +(a, to +ire ro ' e(ect a/a0 behavior	void	Un e(ect the ,iven ro ' .
un e(ectA((Ro ' ?@	-	void	Un e(ect a((ro ' .
to,,(e*hec3A((?@	-	void	9o,,(e header chec3bo0 tate.
+i(ter?@	-	>oid	#i(ter the data.
e(ectA((Ro ' ?@	-	void	Se(ect a((ro ' .
e(ectA((Ro ' 7n!a,e?@	-	void	Se(ect a((ro ' on current &a,e.
un e(ectA((Ro ' 7n!a,e?@	-	void	Un e(ect a((ro ' on current &a,e.

Skinning

Data9ab(e re ide in a " ain container e(e " ent ' hich *style* and *styleClass* o&tion a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.uiBdatab(e	8 ain container e(e " ent
.uiBdatab(eBdata	9ab(e body
.uiBdatab(eBe "&tyB " e a,e	E " &ty " e a,e ro '
.uiBdatab(eBheader	9ab(e header
.uiBdatab(eB+ooter	9ab(e +ooter
.uiBortab(eBco(u " n	Sortab(e co(u " n
.uiBortab(eBco(u " nBicon	Icon o+ a ortab(e icon
.uiBe0&andedBro ' Bcontent	*ontent o+ an e0&anded ro '
.uiBro ' Bto , ,(er	Ro ' to , ,(er +or ro ' e0&an ion
.uiBeditab(eBco(u " n	*o(u " n ' ith a ce((editor
.uiBce((Beditor	*ontainer o+ in&ut and out&ut contro(o+ an editab(e ce((
.uiBce((BeditorBin&ut	*ontainer o+ in&ut contro(o+ an editab(e ce((
.uiBce((BeditorBout&ut	*ontainer o+ out&ut contro(o+ an editab(e ce((
.uiBdatab(eBeven	Even nu " bered ro '
.uiBdatab(eBodd	7 dd nu " bered ro '
.uiBdatab(eB cro((ab(e	8 ain container e(e " ent o+ a cro((ab(e tab(e.
.uiBdatab(eB cro((ab(eBheader	. eader ' ra&&er o+ a cro((ab(e tab(e.
.uiBdatab(eB cro((ab(eBheaderBbo0	. eader container o+ a cro((ab(e tab(e.
.uiBdatab(eB cro((ab(eBbody	1ody container o+ a cro((ab(e tab(e.
.uiBdatab(eB cro((ab(eB+ooter	#ooter ' ra&&er o+ a cro((ab(e tab(e.
.uiBdatab(eB cro((ab(eB+ooterBbo0	#ooter container o+ a cro((ab(e tab(e.
.uiBdatab(eBre i=ab(e	8 ain container e(e " ent o+ a re i=ab(e tab(e.

3.31 DefaultCommand

Which co " " and to ub " it the +or " ' ith ' hen enter 3ey i &re ed a co " " on &rob(e" in ' eb a&& not /u t &ecific to AS#. 1ro ' er tend to behave different(y a there doe n't ee" to be a standard and even i+ a standard e0i t E &robab(y ' i((not care about it. 9here are o " e u,(y ' or3around (i3e &(acin, a hidden button and ' ritin, /ava cri&t +or every +or" in your a&&. De+au(t*o " " and o(ve thi &rob(e" by nor " a(i=in, the co " " and?e.,.. button or (in3@ to ub " it the +or" ' ith on enter 3ey &re .

Info

9a,	defaultCommand
*o "&onent *(a	org.primefaces.component.defaultcommand.DefaultCommand
*o "&onent 9y&e	org.primefaces.component.DefaultCommand
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DefaultCommandRenderer
Renderer *(a	org.primefaces.component.defaultcommand.DefaultCommandRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that "a& to a server ide UI*o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et
tar,et	nu((Strin,	Identifi+ier o+ the de+au(t co " " and co " &onent.
co&e	nu((Strin,	Identifi+ier o+ the ance tor co " &onent to enab(e " u(ti&e de+au(t co " " and in a +or".

Getting Started with the DefaultCommand

De+au(t*o " " and " u t be ne ted in ide a +or" reCuire target o&tion to re+erence a c(ic3ab(e co " " and. E0a" &(e be(o ' tri,,er btn2 ' hen enter 3ey i &re ed. <ote that an in&ut " u t have +ocu ed due to bro ' er nature.

```
<h:form id="form">

    <h:panelGrid columns="3" cellpadding="5">
        <h:outputLabel for="name" value="Name:" style="font-weight:bold"/>
        <p:inputText id="name" value="#{defaultCommandBean.text}" />
        <h:outputText value="#{defaultCommandBean.text}" id="display" />
    </h:panelGrid>

    <p:commandButton value="Button1" id="btn1" actionListener="#{bean.submit1}"
        ajax="false"/>
    <p:commandButton value="Button2" id="btn2" actionListener="#{bean.submit2}" />
    <h:commandButton value="Button3" id="btn3" actionListener="#{bean.submit3}" />
    <p:defaultCommand target="bt2" />

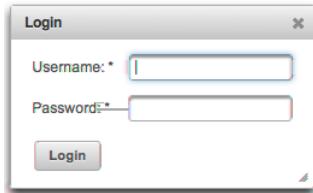
</h:form>
```

Scope

If you need to initialize default values and on a "use" the `coerce` attribute that refers to the ancestor or component owner of the target input.

3.32 Dialog

Dialog is a component that can overlay other elements on a page.



Info

9a,	dialog
*o " &onent *(a	org.primefaces.component.dialog.Dialog
*o " &onent 9y&e	org.primefaces.component.Dialog
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DialogRenderer
Renderer *(a	org.primefaces.component.dialog.DialogRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component
rendered	true	Boolean	Value indicating whether the component is rendered. If false, it is not rendered.
binding	nu((Object	An expression that binds to a server-side UI component in a backing bean.
' id , et>ar	nu((String,	<a href= the client id , et
header	nu((String,	Text of the header
draggable	true	Boolean	Specifies dragability
resizable	true	Boolean	Specifies resizability
" oda(+a(e	Boolean	Enables "oda(ity).
visible	+a(e	Boolean	When enabled dialog is visible by default.
' idth	auto	Integer	Width of the dialog,
height	auto	Integer	. height of the dialog,
" inWidth	150	Integer	Initial width of the dialog.

Name	Default	Type	Description
" in . ei, ht	0	Inte, er	8 ini " u " hei, ht o+ a re i=ab(e dia(o,..
ty(e	nu((Strin,	In(ine ty(e o+ the dia(o,..
ty(e*(a	nu((Strin,	Sty(e c(a o+ the dia(o,
ho ' E+ect	nu((Strin,	E+ect to u e ' hen ho ' in , the dia(o,
hideE+ect	nu((Strin,	E+ect to u e ' hen hidin , the dia(o,
&o ition	nu((Strin,	De+ine ' here the dia(o, hou(d be di &(ayed
c(o ab(e	true	1oo(ean	De+ine i+ c(o e icon hou(d be di &(ayed or not
onSho '	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen dia(o, i di &(ayed.
on . ide	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen dia(o, i hidden.
a&&end9o	nu((Strin,	A&&end the dia(o, to the e(e " ent de+ined by the ,iven earch e0&re ion.
ho ' . eader	true	1oo(ean	De+ine vi ibi(ity o+ the header content.
+ooter	nu((Strin,	9e0t o+ the +ooter.
dyna " ic	+a(e	1oo(ean	Enab(e (a=y (oadin, o+ the content ' ith a/a0.
" ini " i=ab(e	+a(e	1oo(ean	Whether a dia(o, i " ini " i=ab(e or not.
" a0i " i=ab(e	+a(e	1oo(ean	Whether a dia(o, i " a0i " i=ab(e or not.
c(o e7nE ca&e	+a(e	1oo(ean	De+ine i+ dia(o, hou(d c(o e on e ca&e 3ey.
dir	(tr	Strin,	De+ine te0t directionE va(id va(ue are <i>ltr</i> and <i>rtl</i> .
+ocu	nu((Strin,	De+ine ' hich co " &onent to a&&(y +ocu .
+it>ie ' &ort	+a(e	1oo(ean	Dia(o, i=e " i, ht e0ceeed vie ' &ort i+ content i bi,,er than vie ' &ort in ter " o+ hei,ht. +it>ie ' &ort o&tion auto " atica((y ad/u t hei,ht to +it dia(o, ' ithin the vie ' &ort.

Getting started with the Dialog

Dia(o, i a &ane(co " &onent containin, other co " &onent E note that by de+au(t dia(o, i not vi ib(e.

```
<p:dialog>
    <h:outputText value="Resistance to PrimeFaces is Futile!" />
    //Other content
</p:dialog>
```

Show and Hide

Sho ' in, and hidin, the dia(o, i ea y u in, the c(ient ide a&i.

```
<p:dialog header="Header Text" widgetVar="dlg">//Content</p:dialog>
<p:commandButton value="Show" type="button" onclick="PF('dlg').show()" />
<p:commandButton value="Hide" type="button" onclick="PF('dlg').hide()" />
```

Effects

There are various effect options to be used when displaying and closing the dialog. Use `showEffect` and `hideEffect` options to affect how the dialog bounces/closes/opens and adds/removes.

```
<p:dialog showEffect="fade" hideEffect="explode" ...>
    //...
</p:dialog>
```

Position

By default the dialog is positioned at center of the view port and `position` option is used to change the location of the dialog. The visible areas

- `String` position, value (like 'center', 'left', 'right', 'top', 'bottom') representing the position within the view port.
- `*Object` separated by and y coordinate value (like `200, 500`)
- `*Object` separated by position value (like '`top`', '`right`'. Use in combination with `center` and `auto` combination)

So "examples" are described below:

```
<p:dialog position="top" ...>
```

```
<p:dialog position="left,top" ...>
```

```
<p:dialog position="200,50" ...>
```

Focus

The dialog occupies the entire window by default which is overridden by the `focus` attribute. This attribute is not defined by the dialog itself, but by the `onblur` and `onfocus` events of the window object. The dialog has its own `onblur` and `onfocus` events, so it is necessary to use the `blur` and `focus` events of the window object to handle focus changes.

Ajax Behavior Events

`close` event is one of the ajax behavior events provided by the dialog, that is fired when the dialog is hidden. It is triggered when the dialog is closed by clicking the close button or pressing the escape key. The `CloseEvent` is an instance of `org.primefaces.event.CloseEvent`.

Example: adding a `ChangeListener` to the dialog to update the value of a component when the dialog is closed.

```
<p:dialog>
    <p:ajax event="close" listener="#{dialogBean.handleClose}" update="msg" />
    //Content
</p:dialog>

<p:messages id="msg" />
```

```
public class DialogBean {

    public void handleClose(CloseEvent event) {
        //Add facesmessage
    }
}
```

9 other a/a0 behavior event are **maximize** and **minimize** that are invoked when dialog is opened or closed.

Client Side Callbacks

Si " is ar to c(o e i tener onShow and onHide are handy ca((bac3 -or c(ient ide in case you need to execute cu to " /ava cri&t.

```
<p:dialog onShow="alert('Visible')" onHide="alert('Hidden')">
    //Content
</p:dialog>
```

Client Side API

Wid, et PrimeFaces.widget.Dialog

Method	Params	Return Type	Description
show()	B	void	Displays dialog.
hide()	B	void	Hides dialog.

Skinning

Dialog, re ide in a main container e(e " ent which styleClass option a&&(y. #o((o ' in, i the (i t o+ structure(ty(e c(a e K

Style Class	Applies
.ui-dialog,	* container element of dialog,
.ui-dialog-titlebar	Title bar
.ui-dialog-header	Header section
.ui-dialog-titlebar-icon	Icon
.ui-dialog-content	Dialog body

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or "ore in+or" ation.

Tips

- U e a&&end9o 'ith care a the &a,e de+inition and ht "(do " 'ou(d be di++erentE +or e0a " &(e i+ dia(o, i in ide an hN+or " co " &onent and a&&end9o1ody i enab(edE on the bro ' er dia(o, ' ou(d be out ide o+ +or" and "ay cau e une0&ected re u(t . In thi ca eE ne t a +or" in ide a dia(o,..
- Do not &(ace dia(o, in ide tab(e E container (i3e div ' ith re(ative &o itionin, or ' ith nonB vi ib(e over+(o ' de+inedE in ca e (i3e the e +unctiona(ity " i,ht be bro3en. 9hi i not a (i" itation but a re u(t o+ D 7 8 "ode(. #or e0a " &(e dia(o, in ide a (ayout unitE tabvie ' E accordion are a cou&(e o+ e0a " &(e . Sa " e a&&(ie to con+ir" Dia(o, a ' e().

3.33 Drag&Drop

Dra, : Dro& uti(itie o+ !ri " e#ace con i t o+ t ' o co " &onent K Dra, ,ab(e and Dro&&ab(e.

3.33.1 Draggable

Info

9a,	draggable
*o " &onent *(a	org.primefaces.component.dnd.Draggable
*o " &onent 9y&e	org.primefaces.component.Draggable
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DraggableRenderer
Renderer *(a	org.primefaces.component.dnd.DraggableRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent
rendered	true	boo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7 b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et
&ro0y	+a(e	1oo(ean	Di &(ay a &ro0y e(e " ent in tead o+ actua(e(e " ent.
dra, 7n(y	+a(e	1oo(ean	S&eci+ie dra, ,ab(e that can't be dro&&ed.
+or	nu((Strin,	Id o+ the co " &onent to add dra, ,ab(e behavior
di ab(ed	+a(e	1oo(ean	Di ab(e dra, ,ab(e behavior ' hen true.
a0i	nu((Strin,	S&eci+ie dra, a0i E va(id va(ue are '0' and 'y'.
contain " ent	nu((Strin,	*on traint dra, ,in, ' ithin the boundarie o+ contain " ent e(e " ent
he(&er	nu((Strin,	. e(&er e(e " ent to di &(ay ' hen dra, ,in,
revert	+a(e	1oo(ean	Revert dra, ,ab(e to it' ori, ina(&o ition ' hen not dro&&ed onto a va(id dro&&ab(e
na&	+a(e	1oo(ean	Dra, ,ab(e ' i((na& to ed,e o+ near e(e " ent
na& 8 ode	nu((Strin,	S&eci+ie the na& " ode. >a(id va(ue are 'both'E

Name	Default	Type	Description
			'inner' and 'outer'.
na&90(erance	%0	Inte,er	Di tance +ro " the na& e(e " ent in &i0e(to tri,,er na&.
=inde0	nu((Inte,er	UInde0 to a&&(y durin, dra,,in,..
hand(e	nu((Strin,	S&eci+ie a hand(e +or dra,,in,..
o&acity	1	Doub(e	De+ine the o&acity o+ the he(&er durin, dra,,in,..
tac3	nu((Strin,	In tac3 " ode dra,,ab(e over(a& i contro(ed auto " atica(y u in, the &rovided e(ector dra,,ed ite " a('ay over(ay other dra,,ab(e .
,rid	nu((Strin,	Dra,,in, ha&&en in every 0 and y &i0e(.
co&e	nu((Strin,	Sco&e 3ey to " atch dra,,ab(e and dro&&ab(e .
cur or	cro hair	Strin,	*SS cur or to di &(ay in dra,,in,..
da hboard	nu((Strin,	Id o+ the da hboard to connect ' ith.

Getting started with Draggable

Any co " &onent can be enhanced ' ith dra,,ab(e behavior ba ica(y thi i achieved by de+inin, the id o+ co " &onent u in, the *for* attribute o+ dra,,ab(e.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="This is actually a regular panel" />
</p:panel>

<p:draggable for="pnl"/>
```

If you o " it the *for* attributeE &arent co " &onent ' i((be eected a the dra,,ab(e tar,et.

```
<h:graphicImage id="campnou" value="/images/campnou.jpg">
    <p:draggable />
</h:graphicImage>
```

Handle

If de+au(t any &oint in dra,,ed co " &onent can be u ed a hand(eE i+ you need a &eci+ic hand(eE you can de+ine it ' ith hand(e o&tion. #o((o ' in, &ane(i dra,,ed u in, it' header on(y.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="I can only be dragged using my header" />
</p:panel>
<p:draggable for="pnl" handle="div.ui-panel-titlebar"/>
```

Drag Axis

Drag, , in, can be (i " ited to either hori=onta((y or vertica((y.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="I am dragged on an axis only" />
</p:panel>

<p:draggable for="pnl" axis="x or y"/>
```

Clone

1 y de+au(tE actua(co " &onent i u ed a the dra, indicatorE i+ you need to 3ee& the co " &onent at it' ori, ina((ocationE u e a c(one he(&er.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="I am cloned" />
</p:panel>

<p:draggable for="pnl" helper="clone"/>
```

Revert

When a dra, ,ab(e i not dro&&ed onto a " atchin, dro&&ab(e& revert o&tion enab(e the co " &onent to " ove bac3 to it' ori, ina(&o ition ' ith an ani " ation.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="I will be reverted back to my original position" />
</p:panel>

<p:draggable for="pnl" revert="true"/>
```

Opacity

Durin, dra, ,in,,E o&acity o&tion can be u ed to ,ive vi ua(+edbac3E he(&er o+ +o((o ' in, &ane(' o&acity i reduced in dra, ,in,,.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="My opacity is lower during dragging" />
</p:panel>

<p:draggable for="pnl" opacity="0.5"/>
```

Grid

De+inin, a ,rid enab(e dra, ,in, in &eci+ic &i0e(. .9hi va(ue ta3e a co " "a e&arated di " en ion in 0Ey +or " at.

```
<p:panel id=" pnl" header="Draggable Panel">
    <h:outputText value="I am dragged in grid mode" />
</p:panel>

<p:draggable for="pnl" grid="20,40"/>
```

Containment

A draggab(e can be restricted to a certain section on &a, eE +o((o ' in, dra, , ab(e cannot ,o out ide o+ it' &parent.

```
<p:outputPanel layout="block" style="width:400px;height:200px;">
    <p:panel id="conpnl" header="Restricted">
        <h:outputText value="I am restricted to my parent's boundaries" />
    </p:panel>
</p:outputPanel>

<p:draggable for="conpnl" containment="parent" />
```

3.33.2 Droppable

Info

9a,	droppable
*o " &onent *(a	org.primefaces.component.dnd.Droppable
*o " &onent 9y&e	org.primefaces.component.Droppable
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DroppableRenderer
Renderer *(a	org.primefaces.component.dnd.DroppableRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier of the component.
rendered	true	Boolean	Logical value to specify if the component is rendered. If false, it is not rendered.
binding	null	Object	An expression that refers to a server-side IDE UI component instance in a backing bean.
id,etc>ar	null	String,	Variable name of the client-side id.
for	null	String,	Id of the component to add droppable behavior.
disabled	false	Boolean	Disable or enable droppable behavior.
hoverStyle*(a	null	String,	Style class to apply when an element is over.
activeStyle*(a	null	String,	Style class to apply when an element is being over.
onDrop	null	String,	Script to execute when a drag-and-drop is dropped.
accept	null	String,	Selector to define the accepted drag-and-drop.
co&e	null	String,	Specify to attach drag-and-drop and droppable.
tolerance	null	String,	Specifies the interaction mode to accept a drag-and-drop.
dataSource	null	String,	Id of a UIData component to connect with.

Getting Started with Droppable

Una de las principales funcionalidades de PrimeFaces es la capacidad de agregar comportamiento de arrastrar y soltar a cualquier componente. Esto se logra mediante el atributo `for` que se aplica al componente que actuará como receptor.

```
<p:outputPanel id="slot" styleClass="slot" />
<p:droppable for="slot" />
```

Este código crea un panel de salida con la clase CSS "slot" y lo hace receptivo para arrastre y soltar, asociándolo al ID "slot".

```
<style type="text/css">
.slot {
    background:#FF9900;
    width:64px;
    height:96px;
    display:block;
}
</style>
```

El atributo `for` se aplica a los componentes que deseas permitir que sean arrastrados y soltados.

```
<p:outputPanel id="slot" styleClass="slot">
    <p:droppable />
</p:outputPanel>
```

Ajax Behavior Events

El evento `drop` es el único y más ampliamente utilizado de los comportamientos de arrastrar y soltar proporcionados por PrimeFaces. Permite ejecutar código JavaScript en el lado del cliente cuando se suelta un elemento en un destino. Puedes definir este evento en el controlador de arrastrar y soltar.

Este código muestra cómo activar el arrastrado y soltado en un componente `graphicImage` y configurar un escuchador Ajax para el evento `drop`.

```
<p:graphicImage id="messi" value="barca/messi_thumb.jpg" />
<p:draggable for="messi"/>

<p:outputPanel id="zone" styleClass="slot" />
<p:droppable for="zone">
    <p:ajax listener="#{ddController.onDrop}" />
</p:droppable>
```

```
public void onDrop(DragDropEvent ddEvent) {
    String draggedId = ddEvent.getDragId();           //Client id of dragged component
    String droppedId = ddEvent.getDropId();           //Client id of dropped component
    Object data = ddEvent.getData();                  //Model object of a datasource
}
```

onDrop

onDrop is a client-side callback that is invoked when a drag-and-drop event occurs and the object is dropped onto another element.

```
<p:outputPanel id="zone" styleClass="slot" />
<p:droppable for="zone" onDrop="handleDrop"/>
```

```
function handleDrop(event, ui) {
    var draggable = ui.draggable,      //draggable element, a jQuery object
        helper = ui.helper,           //helper element of draggable, a jQuery object
        position = ui.position,       //position of draggable helper
        offset = ui.offset;          //absolute position of draggable helper
}
```

DataSource

Drag-and-drop functionality requires data elements that extend the UIData interface. Data is passed in order to connect a droppable to accept data from a data component and define data source options such as the ID of the data component. Events are triggered when data is dragged onto a droppable or into a drag-and-drop action. Dragged cars are displayed with a data table.

```
public class TableBean {

    private List<Car> availableCars;
    private List<Car> droppedCars;

    public TableBean() {
        availableCars = //populate data
    }

    //getters and setters

    public void onCarDrop(DragDropEvent event) {
        Car car = ((Car) event.getData());
        droppedCars.add(car);
        availableCars.remove(car);
    }
}
```

```

<h:form id="carForm">
    <p:fieldset legend="AvailableCars">
        <p:dataGrid id="availableCars" var="car"
            value="#{tableBean.availableCars}" columns="3">
            <p:column>
                <p:panel id=" pnl" header="#{car.model}" style="text-align:center">
                    <p:graphicImage value="/images/cars/#{car.manufacturer}.jpg" />
                </p:panel>
                <p:draggable for=" pnl" revert="true" handle=".ui-panel-titlebar"
                    stack=".ui-panel"/>
            </p:column>
        </p:dataGrid>
    </p:fieldset>

    <p:fieldset id="selectedCars" legend="Selected Cars" style="margin-top:20px">
        <p:outputPanel id="dropArea">

            <h:outputText value="!!!Drop here!!!"
                rendered="#{empty tableBean.droppedCars}" style="font-size:24px;" />

            <p: dataTable var="car" value="#{tableBean.droppedCars}"
                rendered="#{not empty tableBean.droppedCars}">
                <p:column headerText="Model">
                    <h:outputText value="#{car.model}" />
                </p:column>
                <p:column headerText="Year">
                    <h:outputText value="#{car.year}" />
                </p:column>
                <p:column headerText="Manufacturer">
                    <h:outputText value="#{car.manufacturer}" />
                </p:column>
                <p:column headerText="Color">
                    <h:outputText value="#{car.color}" />
                </p:column>
            </p: dataTable>
        </p:outputPanel>
    </p:fieldset>

    <p:droppable for="selectedCars" tolerance="touch"
        activeStyleClass="ui-state-highlight" datasource="availableCars"
        onDrop="handleDrop"/>
        <p:ajax listener="#{tableBean.onCarDrop}" update="dropArea availableCars" />
    </p:droppable>

</h:form>

<script type="text/javascript">
    function handleDrop(event, ui) {
        ui.draggable.fadeOut('fast');           //fade out the dropped item
    }
</script>

```

Tolerance

There are four different tolerance modes that define the 'way of acceptance' a drag-and-drop.

Mode	Description
hit	drag-and-drop over the entire droppable area
intersect	drag-and-drop over the droppable area at least 50% intersected
pointer	drag-and-drop over the droppable area at any point
touch	drag-and-drop over the droppable area at any point

Acceptance

You can (in) which drag-and-drop can be dropped onto droppable using the `accept` attribute which defines the drag-and-drop area. #only in, e.g. "and" (or "or") a, e. Only if it is accepted by droppable.

```
<p:graphicImage id="messi" value="barca/messi_thumb.jpg" />
<p:draggable for="messi" scope="forward"/>

<p:graphicImage id="xavi" value="barca/xavi_thumb.jpg" />
<p:draggable for="xavi" scope="midfield"/>

<p:outputPanel id="forwardsonly" styleClass="slot" scope="forward" />
<p:droppable for="forwardsonly" />
```

Skinning

`hoverStyleClass` and `activeStyleClass` attributes are used to change the style of the droppable when interacting with a drag-and-drop.

3.34 Dock

Doc3 co " &onent " i " ic the ' e((3no ' n doc3 interface o+ 8 ac 7S F.



Info

9a,	dock
*o " &onent *(a	org.primefaces.component.dock.Dock
*o " &onent 9y&e	org.primefaces.component.Dock
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.DockRenderer
Renderer *(a	org.primefaces.component.dock.DockRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+i er o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
" ode(nu((8 enu 8 ode(8 enu 8 ode(in tance to create " enu &ro ,ra " " atica(y
&o ition	botto "	Strin ,	!o ition o+ the doc3E <i>bottom or top</i> .
ite " Width	-0	Inte ,er	Initia(' idth o+ ite " .
" a0Width	50	Inte ,er	8 a0i " u " ' idth o+ ite " .
&ro0i " ity	90	Inte ,er	Di tance to en(ar ,e.
ha(i ,n	center	Strin ,	. ori=onta(a(i ,n " entE
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et.

Getting started with the Dock

A doc3 i co " &o ed o+ " enuite " .

```
<p:dock>
    <p:menuItem value="Home" icon="/images/dock/home.png" url="#" />
    <p:menuItem value="Music" icon="/images/dock/music.png" url="#" />
    <p:menuItem value="Video" icon="/images/dock/video.png" url="#" />
    <p:menuItem value="Email" icon="/images/dock/email.png" url="#" />
    <p:menuItem value="Link" icon="/images/dock/link.png" url="#" />
    <p:menuItem value="RSS" icon="/images/dock/rss.png" url="#" />
    <p:menuItem value="History" icon="/images/dock/history.png" url="#" />
</p:dock>
```

Position

Doc3 can be located in t ' o (ocation *top* or *bottom* ?de+au(t@. #or a doc3 &o itioned at to& et &o ition to to&.

Dock Effect

When "ou e i over the doc3 ite " E icon are =oo " ed in. 9he con+i, uration o+ thi e+ect i done via the "a0Width and &ro0i " ity attribute .

Dynamic Menus

8 enu can be created &ro, ra " " atica((y a ' e((E ee the dyna " ic " enu &art in " enu co " &onent ection +or " ore in+or " ation and an e0a " &(e.

Skinning

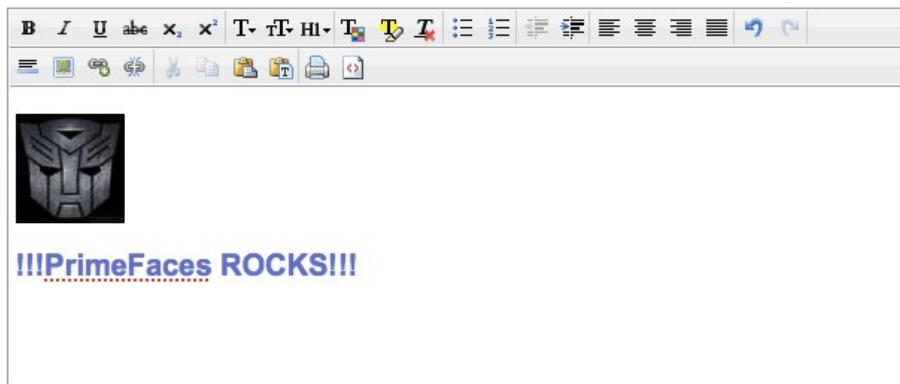
#o((o ' in, i the (i t o+ tructura(ty(e c(a e E V&o itonW can be *top* or *bottom*.

Style Class	Applies
.uiBdoc3BV&o itionW	8 ain container.
.uiBdoc3BcontainerBV&o itionW	8 enu ite " container.
.uiBdoc3Bite " BV&o itionW	Each " enu ite " .

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.35 Editor

Editor is an input component with rich text editing, capability.



Info

9a,	editor
*o " &onent *(a	org.primefaces.component.editor.Editor
*o " &onent 9y&e	org.primefaces.component.Editor
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.EditorRenderer
Renderer *(a	org.primefaces.component.editor.EditorRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Value to specify the rendering of the component.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
value	null	Object	A value of the component than can be either an Expression or a (iterable) object.
converter	null	Converter	An expression or a converter that defines a converter for the component. When it's an Expression it's resolved to a converter instance. In case it's a static object it's used refer to a converter id.
immediate	false	Boolean	When set true it performs validation (optional) immediately before the component.
required	false	Boolean	Specifies the component as required.

Name	Default	Type	Description
validator	null	Method	A " method expression that refers to a " method validation, the input.
valueChangeListener	null	Method	A " method expression that refers to a " method for handling, a valuechange event.
requiredMessage	null	String	Message to be displayed when required validation fails.
converterMessage	null	String	Message to be displayed if converter fails.
validatorMessage	null	String	Message to be displayed when validation fails.
clientId	null	String	<a " element of the client identifier.
control	null	String	Attribute to control to update when toolbar.
height	null	Integer	Height of the editor.
width	null	Integer	Width of the editor.
disabled	false	Boolean	Disabled editor.
type	null	String	Inline type of the editor container.
typeface	null	String	Style of the editor container.
onchange	null	String	Script code to execute when editor data changes.
placeholder	null	Integer	Placeholder " in, the Editor is added to the server in, value expression.

Getting started with the Editor

Rich Text entered in, the Editor is added to the server in, value expression.

```
public class Bean {
    private String text;
    //getter and setter
}
```

```
<p:editor value="#{bean.text}" />
```

Custom Toolbar

Toolbar of editor is ready to control in, controls option

```
<p:editor value="#{bean.text}" controls="bold italic underline strikethrough" />
```



Here is the full list of available controls:

G bo(d	G /u ti+y
G ita(ic	G undo
G under(ine	G redo
G tri3ethrou,h	G ru(e
G ub cri&t	G i "a,e
G u&er cri&t	G (in3
G +ont	G un(in3
G i=e	G cut
G ty(e	G co&y
G co(or	G &a te
G hi,h(i,ht	G &a tete0t
G bu((et	G &rint
G nu "berin,	G ource
G a(i,n(e+t	G outdent
G center	G indent
G a(i,nri,ht	G re " ove#or " at

Client Side API

Wid, etn *PrimeFaces.widget.Editor*

Method	Params	Return Type	Description
init?@	B	void	Initializes a new editor. Subsequent calls do not reinitialize the editor.
ave . 9 8 6?@	B	void	Save content (text in textarea) back to the textarea.
c(ear?@	B	void	Clears the text in editor.
enab(e?@	B	void	Enables editing.
di ab(e?@	B	void	Disables editing.
+ocu ?@	B	void	Add cursor or focus to edit area.
e(ectA((?@	B	void	Selects a text in editor.
,etSe(ected . 9 8 6?@	B	String	Returns selected text area.
,etSe(ected9e0t?@	B	String	Returns selected text in &(ain +or " at).

Skinning

#o((o ' in , i the (i t o+ tructura(ty(e c(a e .

Style Class	Applies
.uiBeditor	8 ain container.
.uiBeditorBtoo(bar	9oo(bar o+ editor.
.uiBeditorB,rou&	1utton ,rou& .
.uiBeditorBbutton	Each button.
.uiBeditorBdivider	Divider to e&arate button .
.uiBeditorBdi ab(ed	Di ab(ed editor contro(.
.uiBeditorBi t	Dro&do ' n (i t .
.uiBeditorBco(or	*o(or &ic3er.
.uiBeditorB&o&u&	!o&u& over(ay .
.uiBeditorB&ro " &t	7ver(ay to &rovide in&ut.
.uiBeditorB " e a,e	7ver(ay di &(ayin, a " e a,e.

Editor i not inte,rated ' ith 9he " eRo((er a there i on(y one icon et +or the contro(.

3.36 Effect

Effect component based on the /Dquery UI effect library.

Info

9a,	effect
9a, *(a	org.primefaces.component.effect.EffectTag
*o "&onent *(a	org.primefaces.component.effect.Effect
*o "&onent 9y&e	org.primefaces.component.Effect
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.EffectRenderer
Renderer *(a	org.primefaces.component.effect.EffectRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component
rendered	1	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	nu((Object	An expression that maps to a server-side UI component instance in a backing bean
effect	nu((String,	
event	nu((String,	Do an event to attach the event that executes the animation
type	nu((String,	Specifies the name of the animation
for	nu((String,	Component that is animated
speed	1000	Integer	Speed of the animation
delay	nu((Integer	Time to wait until running the effect.

Getting started with Effect

Effect component needs a trigger and target which inherit 'parent' by default. In addition, it can be triggered by running the effect on the target component.

```
<h:outputText value="#{bean.value}">
    <p:effect type="pulsate" event="click" />
</h:outputText>
```

Effect Target

Here "ay be ca e ' here you ' ant to di &(ay an effect on another target on the a"e &a,e ' hi(e 3ee&in, the &arent a the tri,,er. U e for o&tion to &eci+y a tar,et.

```
<h:outputLink id="lnk" value="#">
    <h:outputText value="Show the Barca Temple" />
    <p:effect type="appear" event="click" for="img" />
</h:outputLink>

<p:graphicImage id="img" value="/ui/barca/campnou.jpg" style="display:none"/>
```

With thi ettin,E out&ut6in3 beco "e the tri,,er +or the effect on ,ra&hicl "a,e. When the (in3 i c(ic3edE initia((y hidden ,ra&hicl "a,e co "e u& ' ith a +ade effect.

Note It' i " &ortant +or co " &onent that have the effect co "&onent a a chi(d to have an a i,ned id becau e o "e co " &onent do not render their c(ientId' i+ you don't ,ive the " an id e0&(icit(y.

List of Effects

#o((o ' in , i the (i t o+ effect %

- ↳ bind
- ↳ click
- ↳ drop
- ↳ fade
- ↳ float
- ↳ hide
- ↳ bounce
- ↳ highlight
- ↳ pulse
- ↳ shake
- ↳ move
- ↳ transition

Effect Configuration

Each effect has different parameters or animation (like color or duration and more). In order to change the configuration of the animation & provide the effect with the parameters, .

```
<h:outputText value="#{bean.value}">
    <p:effect type="scale" event="mouseover">
        <f:param name="percent" value="90"/>
    </p:effect>
</h:outputText>
```

It's important to provide them with the configuration parameters.

```
<h:outputText value="#{bean.value}">
    <p:effect type="blind" event="click">
        <f:param name="direction" value="'horizontal'" />
    </p:effect>
</h:outputText>
```

For the purpose of configuration, each effect has its own documentation.

<http://docs.jquery.com/UI/Effects>

Effect on Load

Effect can also be applied to any AS# component when it is loaded or triggered after an asynchronous request (like a click or a mouseover). The event name is `load` and the animation type is `pulse`.

```
<p:messages id="messages">
    <p:effect type="pulsate" event="load" delay="500">
        <f:param name="mode" value="'show'" />
    </p:effect>
</p:messages>

<p:commandButton value="Save" actionListener="#{bean.action}" update="messages"/>
```

3.37 FeedReader

#eedReader is used to display content from a feed.

Info

9a,	feedReader
*o "&onent *(a	org.primefaces.component.feedreader.FeedReader
*o "&onent 9y&e	org.primefaces.component.FeedReader
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.FeedReaderRenderer
Renderer *(a	org.primefaces.component.feedreader.FeedReaderRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
value	null	String	URL of the feed.
var	null	String	Iterator to refer each item in feed.
item	Unbound	Integer	Number of item to display.

Getting started with FeedReader

#eedReader requires a feed URL to display and render its content for each feed item.

```
<p:feedReader value="http://rss.news.yahoo.com/rss/sports" var="feed">
    <h:outputText value="#{feed.title}" style="font-weight: bold"/>
    <h:outputText value="#{feed.description.value}" escape="false"/>
    <p:separator />
    <f:facet name="error">
        Something went wrong.
    </f:facet>
</p:feedReader>
```

Note that you need the PrimeFaces library in your classpath to use the feedreader component.

3.38 Fieldset

#ie(d et i a ,rou&in, co " &onent a an e0ten ion to ht "(+ie(d et.



Info

9a,	fieldset
*o " &onent *(a	org.primefaces.component.fieldset.Fieldset
*o " &onent 9y&e	org.primefaces.component.Fieldset
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.FieldsetRenderer
Renderer *(a	org.primefaces.component.fieldset.FieldsetRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If false, the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a bean.
' id, et>ar	null	String	<a href="#" value of the client id attribute.
(e, end	null	String	Start and end tags.
type	null	String	Input type of the field.
type*(a	null	String	Style class of the field.
to, , (eab(e	+absolute	Boolean	Specifies content to be included with animation.
to, , (eS&eed	500	Integer	Specifies duration in milliseconds.
co((a& ed	+absolute	Boolean	Defines initial visibility state of content.

Getting started with Fieldset

#ie(d et i u ed a a container co " &onent +or it chi(dren.

```
<p:fieldset legend="Simple Fieldset">
    <h:panelGrid column="2">
        <p:graphicImage value="/images/godfather/1.jpg" />
        <h:outputText value="The story begins as Don Vito Corleone ..." />
    </h:panelGrid>
</p:fieldset>
```

Legend

6e, end can be de+ined in t ' o ' ay E ' ith (e, end attribute a in e0a " &(e above or u in, (e, end +acet. U e +acet ' ay i+ you need to &(ace cu to " ht " (other than i " &(e te0t.

```
<p:fieldset>
    <f:facet name="legend">
        </f:facet>

        //content
</p:fieldset>
```

Here is an example that adds a facelet element and update the component when it is toggled.

```
<p:growl id="messages" />

<p:fieldset legend="Toggleable Fieldset" toggleable="true"
    <p:ajax listener="#{bean.onToggle}" update="messages">
        //content
    </p:fieldset>
```

```
public void onToggle(ToggleEvent event) {
    Visibility visibility = event.getVisibility();
    FacesMessage msg = new FacesMessage();
    msg.setSummary("Fieldset " + event.getId() + " toggled");
    msg.setDetail("Visibility: " + visibility);

    FacesContext.getCurrentInstance().addMessage(null, msg);
}
```

Client Side API

Wid, et PrimeFaces.widget.Fieldset

Method	Params	Return Type	Description
to,,(e?)	B	void	9o,,(e +ie(d et content.

Skinning

style and styleClass option apply to the fieldset. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB+ie(d et	8 ain container
.uiB+ie(d etBto,,(eab(e	8 ain container ' hen +ie(d et i to,,(eab(e
.uiB+ie(d et .uiB+ie(d etB(e,end	6e,end o+ +ie(d et
.uiB+ie(d etBto,,(eab(e .uiB+ie(d etB(e,end	6e,end o+ +ie(d et ' hen +ie(d et i to,,(eab(e
.uiB+ie(d et .uiB+ie(d etBto,,(er	9o,,(e icon on +ie(d et

A skinning, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

Tips

G A co((a& ed +ie(d et ' i((re " ain co((a& ed a+ter a &o tbac3 ince +ie(d et 3ee& it to,,(e tate interna((yE you donZt need to " ana,e thi u in, to,,(e6i tener and co((a& ed o&tion.

3.39 FileDownload

The fileDownload component is used to write binary data to the client to write a server or a writer and treat the binary data. It is used to do the action.

Info

9a,	fileDownload
Action to perform	org.primefaces.component.filedownload.FileDownloadActionListener

Attributes

Name	Default	Type	Description
value	null	StreamedContent	A stream of content in binary format.
contentDisposition	attachment	String	Specifies the disposition type.

Getting started with FileDownload

A user component and action is required to trigger the download process. It can be attached to any component (like a button or a link) and it will be an *org.primefaces.model.StreamedContent* instance. We use, in the build-in *DefaultStreamedContent* implementation. It takes the content from the constructor in the binary format. The second is the file name and the third is the name of the file.

```
public class FileBean {

    private StreamedContent file;

    public FileDownloadController() {
        InputStream stream = this.getClass().getResourceAsStream("yourfile.pdf");
        file = new DefaultStreamedContent(stream, "application/pdf",
            "downloaded_file.pdf");
    }

    public StreamedContent getFile() {
        return this.file;
    }
}
```

Here the content is bound to the value of the download action.

```
<h:commandButton value="Download">
    <p:fileDownload value="#{fileBean.file}" />
</h:commandButton>
```

If you are here, it means the action has been triggered to use a component and it is a file.

```
<h:commandLink value="Download">
    <p:fileDownload value="#{fileBean.file}" />
    <h:graphicImage value="pdficon.gif" />
</h:commandLink>
```

If you'd like to use the "Download" button and co. and include a file download option as well, you can do something like this:

```
<p:commandButton value="Download" ajax="false">
    <p:fileDownload value="#{fileBean.file}" />
</p:commandButton>
```

```
<p:commandLink value="Download" ajax="false">
    <p:fileDownload value="#{fileBean.file}" />
    <h:graphicImage value="pdficon.gif" />
</p:commandLink>
```

ContentDisposition

If you want to download content inline and not as an attachment, you can do so by using the `Content-Disposition` header. Another alternative is to use the `inline` attribute in the `fileDownload` component.

Monitor Status

A modal dialog is non-blocking and cannot be closed. Instead, you can provide a feature to monitor the download via client-side `monitorDownload(startFunction, endFunction)` method. This allows you to display a progress bar while the download is in progress.

```
<script type="text/javascript">
    function showStatus() {
        PF('statusDialog').show();
    }

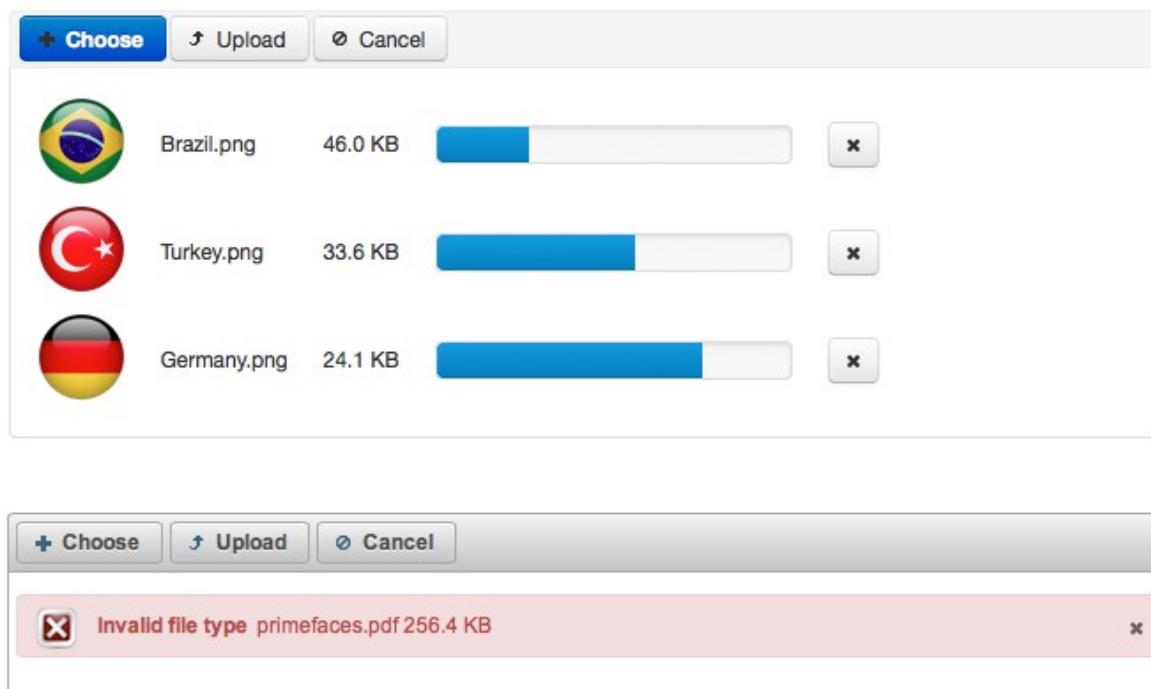
    function hideStatus() {
        PF('statusDialog').hide();
    }
</script>
```

```
<h:form>
    <p:dialog modal="true" widgetVar="statusDialog" header="Status" draggable="false"
        closable="false">
        <p:graphicImage value="/design/ajaxloadingbar.gif" />
    </p:dialog>
    <p:commandButton value="Download" ajax="false"
        onclick="PrimeFaces.monitorDownload(showStatus, hideStatus)">
        <p:fileDownload value="#{fileDownloadController.file}" />
    </p:commandButton>
</h:form>
```

*Note: This feature is currently disabled.

3.40 FileUpload

#i(eU&oad ,oe beyond the bro ' er in&ut ty&ePI+i(eI +unctiona(ity and +eature an . 9 8 65 &o ' ered rich o(ution ' ith ,race+u(de,radation +or (e,acy bro ' er .



Info

9a,	fileUpload
*o " &onent *(a	org.primefaces.component.fileupload.FileUpload
*o " &onent 9y&e	org.primefaces.component.FileUpload
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.FileUploadRenderer
Renderer *(a	org.primefaces.component.fileupload.FileUploadRenderer

Attributes

Name	Default		Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	boo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that "a& to a erver ide UI*o " &onent in tance in a bac3in , bean.

Name	Default		Description
value	null	object	>a(ue o+ the co " &onent than can be either an E6 e0&re ion o+ a (itera(te0t.
converter	null	*onverter (String,	An e(e0&re ion or a (itera(te0t that de-line a converter +or the co " &onent. When it' an E6 e0&re ion it' re o(ved to a converter in tance. In ca e it' a static te0tE it " u t re+er to a converter id.
i " " ediate	+a(e	boolean	When et trueE &roce va(idation (o, ic i e0ecuted at a&&(y reCue t va(ue &ha e +or thi co " &onent.
reCured	+a(e	boolean	8 ar3 co " &onent a reCured.
va(idator	null	8 methodE0&r	A " method e0&re ion that re+er to a " method va(idation , the in&ut.
va(ue*han,e6i tener	null	8 methodE0&r	A " method e0&re ion that re+er to a " method +or hand(in , a va(uchan ,eevent.
reCured 8 e a,e	null	String,	8 e a,e to be di &(ayed ' hen reCured +ie(d va(idation +ai(.
converter 8 e a,e	null	String,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator 8 e a,e	null	String,	8 e a,e to be di &(ayed ' hen va(idation +ai(.
' id,et>ar	null	String,	<a " e o+ the c(ient ide ' id,et.
u&date	null	String,	*o " &onent? @ to u&date a+ter +i(eu&(oad co " &(ete .
&roce	null	String,	*o " &onent? @ to &roce in +i(eu&(oad reCue t.
+i(eU&(oad6i tener	null	8 methodE0&r	8 method to invo3e ' hen a +i(e i u&(oaded.
" u(ti&(e	+a(e	boolean	A((o ' choo in , o+ " u(ti +i(e u&(oad +ro " native +i(e bro ' e dia(o,
auto	+a(e	boolean	When et to trueE e(ectin , a +i(e tart the u&(oad &roce i " &(icit(y).
(abe(*hoo e	String,	6abe(o+ the bro ' e button.
a((o ' 9y&e	null	String,	Re, u(ar e0&re ion +or acce&ted +i(e ty&e E e.,. (?:].^]@?, i+!/&e\$, ^&n, @_l
i=e6i " it	null	Integer	Individua(+i(e i=e (i " it in byte .
+i(e6i " it	null	Integer	8 a0i " u " nu " ber o+ +i(e a((o ' ed to u&(oad.
ty(e	null	String,	In(ine ty(e o+ the co " &onent.
ty(e*(a	null	String,	Sty(e c(a o+ the co " &onent.
" ode	advanced	String,	8 ode o+ the +i(eu&(oadE can be simple or advanced.
u&(oad6abe(U&(oad	String,	6abe(o+ the u&(oad button.
cance(6abe(*ance(String,	6abe(o+ the cancel button.

Name	Default		Description
inva(id\$=e 8 e a,e	nu()	Strin,	8 e a,e to di &(ay ' hen i=e (i " it e0ceed .
inva(id#i(e 8 e a,e	nu()	Strin,	8 e a,e to di &(ay ' hen +i(e i not acce&ted.
+i(e6i " it 8 e a,e	nu()	Strin,	8 e a,e to di &(ay ' hen +i(e (i " it e0ceed .
dra,Dro&Su&&ort	true	1 oo(ean	S&eci+ie dra,dro& ba ed +i(e e(ection +ro " +i(e y te " E de+au(t i true and ' or3 on(y on u&&orted bro ' er .
on tart	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen u&(oad be,in .
onerror	nu()	Strin,	*a((bac3 to e0ecute i+ +i(eu&(oad reCue t +ai(.
onco "&(ete	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen u&(oad end .
di ab(ed	+a(e	1 oo(ean	Di ab(e co " &onent ' hen et true.
" e a,e9e "&(ate	Vna " eW V i=eW	Strin,	8 e a,e te " &(ate to u e ' hen di &(ayin , +i(e va(idation error .
&revie ' Width	20	Inte,er	Width +or i " a,e &revie ' in &i0e(.

Getting started with FileUpload

#i(eU&(oad en,ine on the erver ide can either be erv(et).0 or co " " on +i(eu&(oad. !ri " e#ace e(ect the " o t a&&ro&riate u&(oader en,ine by detection and it i &o ib(e to +orce one or the other u i,n an **optional** con+i, l1

```

<filter>
    <filter-name>PrimeFaces FileUpload Filter</filter-name>
    <filter-class>
        org.primefaces.webapp.filter.FileUploadFilter
    </filter-class>
</filter>
<filter-mapping>
    <filter-name>PrimeFaces FileUpload Filter</filter-name>
    <servlet-name>Faces Servlet</servlet-name>
</filter-mapping>

```

<ote that the erv(etBna " e hou(d " atch the con+i,ured na " e o+ the AS# erv(et ' hich i #ace Serv(et in thi ca e. A(ternative(y you can do a con+i,uration ba ed on ur(B&attern a ' e(.

Simple File Upload

Si "&(e+i(e u&(oad "ode ' or3 in (e,acy "ode ' ith a +i(e in&ut ' ho e va(ue hou(d be an U&(oaded#i(e in tance. A/a0 u&(oad are not u&&orted in i " &(e u&(oad.

```

<h:form enctype="multipart/form-data">
    <p:fileUpload value="#{fileBean.file}" mode="simple" />
    <p:commandButton value="Submit" ajax="false"/>
</h:form>

```

```

import org.primefaces.modelUploadedFile;

public class FileBean {
    private UploadedFile file;
    //getter-setter
}

```

Advanced File Upload

#i(eU&(oad6i tener i the ' ay to acce the u&(oaded +i(e in thi "odeE ' hen a +i(e i u&(oaded de+ined +i(eU&(oad6i tener i &roce ed ' ith a #i(eU&(oadEvent a the &ara " eter.

```

<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}" />

```

```

public class FileBean {

    public void handleFileUpload(FileUploadEvent event) {
        UploadedFile file = event.getFile();
        //application code
    }
}

```

Multiple Uploads

The `multiple` attribute indicates that multiple files can be uploaded at once. If this attribute is set to `true`, the file input field will be a multiple file input field.

```
<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}" multiple="true" />
```

Auto Upload

The default behavior requires the user to click the upload button. You can change this by setting `auto` to `true`. Auto uploads are triggered upon a file selection or by clicking the upload button.

```
<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}" auto="true" />
```

Partial Page Update

After the file is uploaded, you can update the page using the `update` attribute to update any component on the page. #{} is used to refer to the file bean.

```
<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}" update="msg" />
<p:growl id="msg" />
```

```
public class FileBean {
    public void handleFileUpload(FileUploadEvent event) {
        //add facesmessage to display with growl
        //application code
    }
}
```

File Filters

File types can be restricted to only accept the file types you've configured. #{} is used to refer to the file bean.

```
<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}"
    allowTypes="/(\.|\\/)(gif|jpe?g|png)$/" />
```

Size Limit

The size limit is the maximum size of each file that needs to be uploaded. It is measured in bytes. #{} is used to refer to the file bean.

```
<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}" sizeLimit="1000" />
```

File Limit

The file upload component restricts the number of files that can be uploaded.

```
<p:fileUpload fileUploadListener="#{fileBean.handleFileUpload}" fileLimit="3" />
```

Validation Messages

The `invalidFileMessage`, `invalidSizeMessage` and `fileLimitMessage` options are provided to display validation messages to the user. If no message template is defined, the user message will be displayed as it is.

Skinning

Style classes are defined in a container element which `style` and `styleClass` options apply. A summary of the style classes available is as follows:

Class	Applies
.ui-fileupload	Span container element
.ui-fileupload-buttonbar	Button bar.
.ui-fileupload-button	Upload button.
.ui-fileupload-start	Upload button.
.ui-fileupload-cancel	Cancel button.
.ui-fileupload-content	Content container.

Browser Compatibility

Advanced features such as drag & drop, file filtering and file preview are supported in modern browsers. Older browsers that do not support these features (such as Internet Explorer 8 and earlier) will fall back to a standard file input field. It is recommended to use a modern browser for best results.

Filter Configuration

File filter configuration is required if you are using compression or decompression on the server side and the file type is not supported by the browser.

Parameter Name	Description
thresholdSize	The size of the file to be uploaded in bytes. It is recommended to set this to 3.
uploadDirectory	The directory path to the file that exceeds the threshold. It is recommended to use the default value of <code>/tmp</code> .

An example configuration below defines the threshold size to be 503b and upload directory to user temporary directory.

```
<filter>
    <filter-name>PrimeFaces FileUpload Filter</filter-name>
    <filter-class>
        org.primefaces.webapp.filter.FileUploadFilter
    </filter-class>
    <init-param>
        <param-name>thresholdSize</param-name>
        <param-value>51200</param-value>
    </init-param>
    <init-param>
        <param-name>uploadDirectory</param-name>
        <param-value>/Users/primefaces/temp</param-value>
    </init-param>
</filter>
```

Note that upload directory is used internally so you may need to inject the location to save the uploaded content into your bean.

3.41 Focus

#ocu i a uti(ity co " &onent that " a3e it ea y to " ana,e the e(e " ent +ocu on a AS# &a,e.

Info

9a,	focus
*o " &onent *(a	org.primefaces.component.focus.Focus
*o " &onent 9y&e	org.primefaces.component.Focus.FocusTag
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.FocusRenderer
Renderer *(a	org.primefaces.component.focus.FocusRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identifier o+ the co " &onent
rendered	true	boo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
+or	nu((Strin,	S&eci+ie the e0act co " &onent to et +ocu
conte0t	nu((Strin,	9he root co " &onent to tart +ir t in&ut earch.
" inSeverity	error	Strin,	8 ini " u " everity (eve(to be u ed ' hen +indin, the +ir t inva(id co " &onent

Getting started with Focus

1y deau(t +ocu ' i((+ind the first enabled and visible input component on &a,e and a&&(y +ocu . In&ut co " &onent can be any e(e " ent uch a in&utE te0tarea and e(ect.

```
<p:focus />

<p:inputText ... />
<h:inputText ... />
<h:selectOneMenu ... />
```

#o((o ' in, i a i" &(e e0a " &(eK

```

<h:form>
    <p:panel id="panel" header="Register">

        <p:focus />

        <p:messages />

        <h:panelGrid columns="3">
            <h:outputLabel for="firstname" value="Firstname: *" />
            <h:inputText id="firstname" value="#{pprBean.firstname}"
                required="true" label="Firstname" />
            <p:message for="firstname" />

            <h:outputLabel for="surname" value="Surname: *" />
            <h:inputText id="surname" value="#{pprBean.surname}"
                required="true" label="Surname"/>
            <p:message for="surname" />
        </h:panelGrid>

        <p:commandButton value="Submit" update="panel"
            actionListener="#{pprBean.savePerson}" />
    </p:panel>
</h:form>

```

When this code is initially run, it will focus on the first input field ('firstname'). If the user enters invalid data (e.g., numbers instead of letters) and submits the form, an error message will be displayed below the input field. The command button will also be disabled until valid data is entered.

Validation Aware

Another useful feature of PrimeFaces is validation. When a component is invalid (*first invalid component*), it receives focus. So if you review the code above, when you enter invalid data into the first input field ('firstname') and submit the form, an error message will appear below the field, and the command button will be disabled until valid data is entered.

Explicit Focus

Additionally, you can set the focus on an input component by using the `p:focus` attribute. For example:

```

<p:focus for="text"/>

<h:inputText id="text" value="#{bean.value}" />

```

3.42 Fragment

#ra, " ent co " &onent i u ed to de+ine auto " atica((y &artia((y &roce and u&date ection ' henever a/a0 reCue t i tri, ,ered by a de cendant co " &onent.

Info

9a,	fragment
*o " &onent *(a	org.primefaces.component.fragment.Fragment
*o " &onent 9y&e	org.primefaces.component.Fragment
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.FragmentRenderer
Renderer *(a	org.primefaces.component.fragment.FragmentRenderer

Attributes

Name	Default	Type	Description
id	null	String	UniCue identi+ier o+ the co " &onent
rendered	true	Boolean	1 oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	null	Object	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in, bean
autoU&date	+a(e	Boolean	U&date the +ra, " ent auto " atica((y.

Getting started with Fragment

In the +o((o ' in, ca eE reCued in&ut +ie(d out ide the +ra, " ent i i,nored and on(y the content o+ the +ra, " ent are &roce edBu&dated auto " atica((y on button c(ic3 ince button i in ide the +ra, " ent. #ra, " ent " a3e it ea y to de+ine &artia(a/a0 &roce and u&date ' ithout e0&(icit(y de+inin, co " &onent identi+ier .

Required: *

Name: **Submit**

```

<h:form>

    <h:panelGrid columns="2">
        <p:outputLabel for="ign" value="Required:" />
        <p:inputText id="ign" required="true" />
    </h:panelGrid>

    <p:fragment autoUpdate="true">
        <h:panelGrid columns="4" cellpadding="5">
            <h:outputLabel for="name" value="Name:" />
            <p:inputText id="name" value="#{pprBean.firstname}" />
            <p:commandButton value="Submit"/>
            <h:outputText value="#{pprBean.firstname}" />
        </h:panelGrid>
    </p:fragment>

</h:form>

```

AutoUpdate has different notion compared to autoUpdate of "e.g., role" (and out-of-bounds). The idea is to automatically update a component after an action like a click or the source is descendant. In other words, if there is no such restriction as they are updated for every action like a click or the source is updated on the source.

3.43 Galleria

Galleria is used to display a set of images.



Info

9a,	galleria
*o "&onent *(a	org.primefaces.component.galleria.Galleria
*o "&onent 9y&e	org.primefaces.component.Galleria
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.GalleriaRenderer
Renderer *(a	org.primefaces.component.galleria.GalleriaRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	Unique identifier of the component.
rendered	true	booleean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin,	nu((7b/ect	An element indicating that it is to be rendered in a bean.
' id,et>ar	nu((Strin,	<a " e o+ the client id, et.
va(ue	nu((*o((ection	*o((ection o+ data to display.
var	nu((Strin,	<a " e o+ variable to access in the component.
ty(e	nu((Strin,	Inline style of the container element.
ty(e*(a	nu((Strin,	Style class of the container element.
effect	+ade	Strin,	<a " e o+ animation to use.

Name	Default	Type	Description
effectSpeed	500	Integer	Duration of animation in milliseconds.
laneWidth	400	Integer	Width of the view port.
laneHeight	-100	Integer	Height of the view port.
innerWidth	40	Integer	Width of the inner area.
innerHeight	-0	Integer	Height of the inner area.
horizontalGrid	true	Boolean	Defines visibility of horizontal grid.
horizontalCaption	false	Boolean	Defines visibility of horizontal caption.
transitionInterval	-1000	Integer	Defines interval of hide transition.
autoPlay	true	Boolean	Indicates if slides are displayed in automatic mode.

Getting Started with Galleria

Slides to display are defined as children of a galleria element:

```
<p:galleria effect="slide" effectDuration="1000">
    <p:graphicImage value="/images/image1.jpg" title="image1" alt="image1 desc" />
    <p:graphicImage value="/images/image2.jpg" title="image1" alt="image2 desc" />
    <p:graphicImage value="/images/image3.jpg" title="image1" alt="image3 desc" />
    <p:graphicImage value="/images/image4.jpg" title="image1" alt="image4 desc" />
</p:galleria>
```

Galleria displays the details of each slide in an overlay which is displayed by clicking, the previous or next icon. It retrieves the image from the title attribute and description from the alt attribute of the image element, providing the title attribute.

Dynamic Collection

So to use you would need to define a dynamic set of slides rather than defining each slide directly. You can use built-in data iteration features:

```
<p:galleria value="#{galleriaBean.images}" var="image" >
    <p:graphicImage value="#{image.path}"
                    title="#{image.title}" alt="#{image.description}" />
</p:galleria>
```

Effects

There are various effect options to be used in transition like bounce, slide, drop, fade, etc. You can also define your own effects.

You can also tune the effect speed option to tune this.

```
<p:galleria effect="slide" effectSpeed="1000">
    //images
</p:galleria>
```

Skinning

Galleria resides in a main container element which *style* and *styleClass* option apply. A 3innin, type class are, obviously, the main the "in, action or " ore in-or " ation. #o((o ' in, i the (i t o+ tructura(type class e

Style Class	Applies
.uiGalleria	*ontainer e(e " ent +or ,a((eria.
.uiGalleriaBane(B ' ra&&er	*ontainer o+ &ane(.
.uiGalleriaBane(*ontainer o+ each i " a,e.
.uiGalleriaBca&tion	*a&tion e(e " ent.
.uiGalleriaBnavB&revE .uiGalleriaBnavBne0t	<avi,ator o+ +i(" tri&.
.uiGalleriaB+i(" tri&B ' ra&&er	*ontainer o+ +i(" tri&.
.uiGalleriaB+i(" tri&	#i(" tri& e(e " ent.
.uiGalleriaB+ra " e	#ra " e e(e " ent in a +i(" tri&.
.uiGalleriaB+ra " eBcontent	*ontent o+ a +ra " e.
.uiGalleriaB+ra " eBi " a,e	9hu " bnai(i " a,e.

3.44 GMap

G 8 a& i a "a& co " &onent inte , rated ' ith Goo ,(e 8 a& A! I >).



Info

9a,	gmap
*o " &onent *(a	org.primefaces.component.gmap.GMap
*o " &onent 9y&e	org.primefaces.component.Gmap
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.GmapRenderer
Renderer *(a	org.primefaces.component.gmap.GmapRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1 oo(ean	1 oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean.
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et.
" ode(nu((8 a& 8 ode(An or ,.&ri " e+ace . " ode(. 8 a& 8 ode(in tance.

Name	Default	Type	Description
ty(e	nu((Strin,	In(ine ty(e o+ the " a& container.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the " a& container.
ty&e	nu((Strin,	9y&e o+ the " a&.
center	nu((Strin,	*enter &oint o+ the " a&.
=oo "	2	Inte ,er	De+ine the initia(=oo " (eve(.
treet>ie '	+a(e	1 oo(ean	*ontro(treet vie ' u&&ort.
di ab(eDe+au(tUI	+a(e	1 oo(ean	Di ab(e de+au(t UI contro(
navi ,ation*ontro(true	1 oo(ean	De+ine vi ibi(it y o+ navi ,ation contro(.
" a&9y&e*ontro(true	1 oo(ean	De+ine vi ibi(it y o+ " a& ty&e contro(.
dra , ,ab(e	true	1 oo(ean	De+ine dra , ,abi(it y o+ " a&.
di ab(edDoub(e*(ic3Uoo "	+a(e	1 oo(ean	Di ab(e =oo " in , on " ou e doub(e c(ic3.
on!oint*(ic3	nu((Strin,	Aava cri&t ca((bac3 to e0ecute ' hen a &oint on " a& i c(ic3ed.
+it1ound	true	1 oo(ean	De+ine i+ center and =oo " hou(d be ca(cu(ated auto " atica(y to contain a((" ar3er on the " a&.

Getting started with GMap

#ir t thin, to do i &(acin, >) o+ the Goo,(e 8 a& A !I that the G 8 a& i ba ed on. Idea((ocation i the head ection o+ your &a,e.

```
<script src="http://maps.google.com/maps/api/js?sensor=true/false"
       type="text/javascript"></script>
```

A Goo,(e 8 a& a&i tate E " andatory en or &ara" eter i u ed to &eci+y i+ your a&&(ication reCuire a en or (i3e G !S (ocator. #our o&tion are reCuired to &(ace a , " a& on a &a,eE the e are center=oo " E ty&e and ty(e.

```
<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
       style="width:600px;height:400px" />
```

centerN *enter o+ the " a& in (atE (n , +or " at
 zoomN Uoo " (eve(o+ the " a&
 typeN 9y&e o+ " a&E va(id va(ue areE IhybridIE I ate((iteIE IhybridI and IterrainI.
 styleN Di " en ion o+ the " a&.

MapModel

G 8 a& i bac3ed by an *org.primefaces.model.map.MapModel* in tanceE !ri " e#ace &rovide *org.primefaces.model.map.DefaultMapModel* a the de+au(t i " &(e " entation. A !I Doc o+ a((G 8 a& re(ated " ode(c(a e are avai(ab(e at the end o+ G 8 a& ection and a(o at /avadoc o+ !ri " e#ace .

Markers

A " ar3er i re&re ented by *org.primefaces.model.map.Marker*.

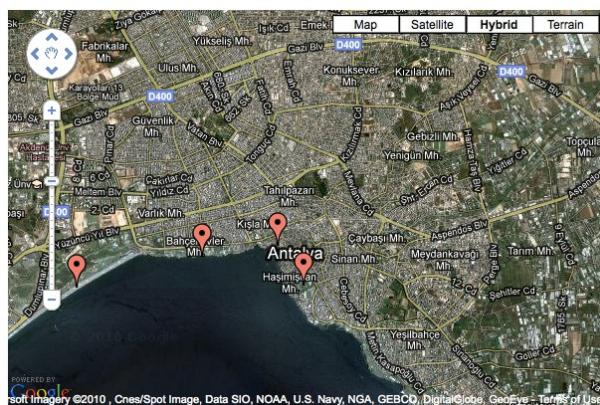
```
<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}"/>
```

```
public class MapBean {

    private MapModel model = new DefaultMapModel();

    public MapBean() {
        model.addOverlay(new Marker(new LatLng(36.879466, 30.667648), "M1"));
        //more overlays
    }

    public MapModel getModel() { return this.model; }
}
```



Polylines

A &o(yine i re&re ented by *org.primefaces.model.map.Polyline*.

```
<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}"/>
```

```

public class MapBean {

    private MapModel model;

    public MapBean() {
        model = new DefaultMapModel();

        Polyline polyline = new Polyline();
        polyline.getPaths().add(new LatLng(36.879466, 30.667648));
        polyline.getPaths().add(new LatLng(36.883707, 30.689216));
        polyline.getPaths().add(new LatLng(36.879703, 30.706707));
        polyline.getPaths().add(new LatLng(36.885233, 37.702323));

        model.addOverlay(polyline);
    }

    public MapModel getModel() { return this.model; }
}

```

Polygons

A polygon is represented by `org.primefaces.model.map.Polygon`.

```
<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}"/>
```

```

public class MapBean {

    private MapModel model;

    public MapBean() {
        model = new DefaultMapModel();

        Polygon polygon = new Polygon();
        polyline.getPaths().add(new LatLng(36.879466, 30.667648));
        polyline.getPaths().add(new LatLng(36.883707, 30.689216));
        polyline.getPaths().add(new LatLng(36.879703, 30.706707));

        model.addOverlay(polygon);
    }

    public MapModel getModel() { return this.model; }
}

```

Circles

A circle is represented by `org.primefaces.model.map.Circle`.

```
<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}"/>
```

```

public class MapBean {

    private MapModel model;

    public MapBean() {
        model = new DefaultMapModel();

        Circle circle = new Circle(new LatLng(36.879466, 30.667648), 500);

        model.addOverlay(circle);
    }

    public MapModel getModel() { return this.model; }
}

```

Rectangles

A rectangle is represented by `org.primefaces.model.map.Rectangle`.

```
<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}"/>
```

```

public class MapBean {

    private MapModel model;

    public MapBean() {
        model = new DefaultMapModel();
        LatLng coord1 = new LatLng(36.879466, 30.667648);
        LatLng coord2 = new LatLng(36.883707, 30.689216);

        Rectangle rectangle = new Rectangle(coord1, coord2);

        model.addOverlay(rectangle);
    }

    public MapModel getModel() { return this.model; }
}

```

Ajax Behavior Events

Grid 8 also provides "any custom" ajax behavior event or you to hook into various features.

Event	Listener Parameter	Fired
over(ajaxEvent)	or, .-ri "e+ace .event." a&.7ver(ajaxEvent)	When an over(ay is detected.
state*han,e	or, .&ri "e+ace .event." a&.State*han,eEvent	When "a& state change .
&ointSe(ect	or, .&ri "e+ace .event." a&.!ointSe(ectEvent	When an e "&ty &oint is detected.
"ar3erDra,	or, .&ri "e+ace .event." a&.8ar3erDra,Event	When a "ar3er is drawn, ,ed.

#0((0 ' in, e0a " &(e di &(ay a #ace 8 e a,e about the e(eected " ar3er ' ith ,ro ' (co " &onent.

```
<h:form>
    <p:growl id="growl" />

    <p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}">
        <p:ajax event="overlaySelect" listener="#{mapBean.onMarkerSelect}"
            update="growl" />
    </p:gmap>
</h:form>
```

```
public class MapBean {

    private MapModel model;

    public MapBean() {
        model = new DefaultMapModel();
        //add markers
    }

    public MapModel getModel() {
        return model;
    }

    public void onMarkerSelect(OverlaySelectEvent event) {
        Marker selectedMarker = (Marker) event.getOverlay();
        //add facesmessage
    }
}
```

InfoWindow

A `co` " "on u e ca e i di &(ayin, an in+o ' indo ' ' hen a " ar3er i e(eected. `gmapInfoWindow` i u ed to i " &(e ent thi &ecia(u e ca e. #o((o ' in, e0a" &(eE di &(ay an in+o ' indo ' that contain an i " a,e o+ the e(eected " ar3er data.

```
<h:form>
    <p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
        style="width:600px;height:400px" model="#{mapBean.model}">
        <p:ajax event="overlaySelect" listener="#{mapBean.onMarkerSelect}" />
        <p:gmapInfoWindow>
            <p:graphicImage value="/images/#{mapBean.marker.data.image}" />
            <h:outputText value="#{mapBean.marker.data.title}" />
        </p:gmapInfoWindow>
    </p:gmap>
</h:form>
```

```

public class MapBean {

    private MapModel model;

    private Marker marker;

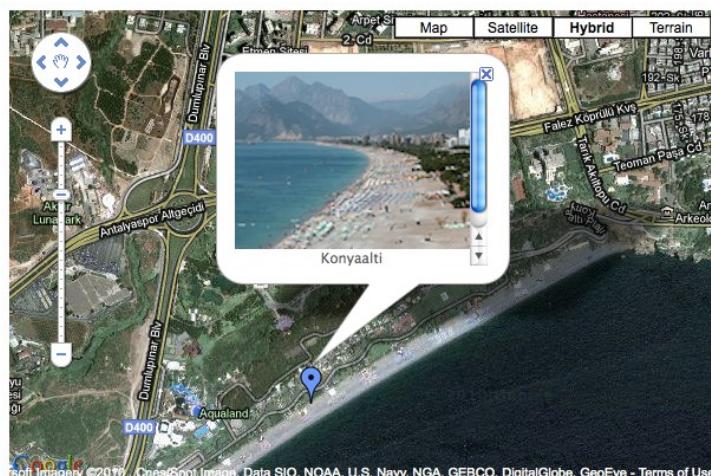
    public MapBean() {
        model = new DefaultMapModel();
        //add markers
    }

    public MapModel getModel() { return model; }

    public Marker getMarker() { return marker; }

    public void onMarkerSelect(OverlaySelectEvent event) {
        this.marker = (Marker) event.getOverlay();
    }
}

```



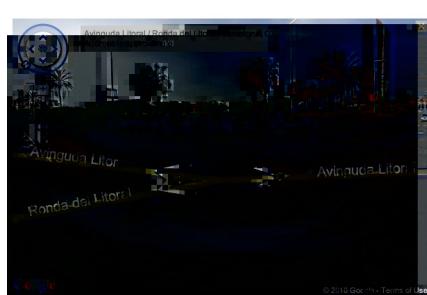
Street View

StreetView is enabled by setting the `streetView` option to true.

```

<p:gmap center="41.381542, 2.122893" zoom="15" type="hybrid"
       style="width:600px;height:400px" streetView="true" />

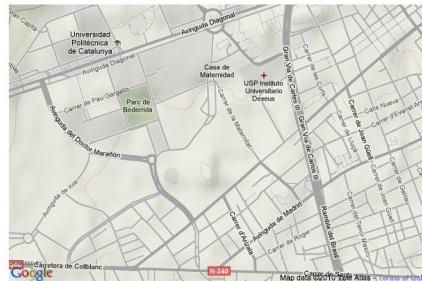
```



Map Controls

*ontro(on " a& can be cu to " i=ed via attribute (i3e *navigationControl* and *mapTypeControl*. A(ternative(y ettin, *disableDefaultUI* to true ' i((re " ove a((contro(at once.

```
<p:gmap center="41.381542, 2.122893" zoom="15" type="terrain"  
        style="width:600px;height:400px"  
        mapTypeControl="false" navigationControl="false" />
```



Native Google Maps API

In case you need to access native code with Java script, you provided `getMap()` method.

```
var gmap = PF('yourWidgetVar').getMap();
//gmap is a google.maps.Map instance
```

#u((" a& a&i i &rovided atK

<http://www.00code.e.co/a&i/a&0docu/entation/ava cri&t/re+erence.htm>

GMap API

`org.primefaces.model.map.MapModel`?`org.primefaces.model.map.DefaultMapModel` i the de+au(t
i " &(e " entation@

Method	Description
add7ver(ay?7ver(ay over(ay@	Add an over(ay to " a&
6i tS 8ar3erT ,et 8ar3er ?@	Return the (i to+ " ar3er
6i tS !o(y(ineT ,et !o(y(ine ?@	Return the (i to+ &o(y(ine
6i tS !o(y, onT ,et !o(y, on ?@	Return the (i to+ &o(y, on
6i tS*irc(eT ,et *irc(e ?@	Return the (i to+ circ(e
6i tSRectan,(eT ,etRectan,(e ?@	Return the (i to+ rectan,(e .
7ver(ay +ind 7ver(ay?Strin , id@	#ind an over(ay by it' uniCue id

org.primefaces.model.map.Overlay

Property	Default	Type	Description
id	null	String	Id of the overlay, generated and used internally
data	null	Object	Data represented in "array"
=index	null	Integer	UBIndex of the overlay

org.primefaces.model.map.Marker extends *org.primefaces.model.map.Overlay*

Property	Default	Type	Description
title	null	String	Title to display on hovering
lat	null	Float	Location of the marker
icon	null	String	Icon of the marker, round
shadow	null	String	Shadow of the marker
cursor	pointer	String	Cursor or to display on hovering
draggable	0	Boolean	Define if the marker can be dragged
clickable	1	Boolean	Define if the marker can be clicked
enable	0	Boolean	Enable shadow if it is not displayed
visible	1	Boolean	Define visibility of the marker

org.primefaces.model.map.Polyline extends *org.primefaces.model.map.Overlay*

Property	Default	Type	Description
&ath	null	float	Width of coordinate
strokeColor	null	String	Color of a line
strokeDash	1	Double	Thickness of a line
strokeWidth	1	Integer	Width of a line

org.primefaces.model.map.Polygon extends *org.primefaces.model.map.Overlay*

Property	Default	Type	Description
&ath	null	float	Width of coordinate
strokeColor	null	String	Color of a line
strokeDash	1	Double	Thickness of a line

Property	Default	Type	Description
strokeWidth	1	Integer	Width of the line
fillColor	null	String	Color of the fill
strokeColor	1	Double	Color of the stroke

org.primefaces.model.map.Circle extends *org.primefaces.model.map.Overlay*

Property	Default	Type	Description
center	null	Point	Center of the circle
radius	null	Double	Radius of the circle.
strokeColor	null	String	Stroke color of the circle.
strokeWidth	1	Double	Stroke width of the circle.
strokeDash	1	Integer	Stroke dash of the circle.
fillColor	null	String	Fill color of the circle.
strokeDash	1	Double	Stroke dash of the circle.

org.primefaces.model.map.Rectangle extends *org.primefaces.model.map.Overlay*

Property	Default	Type	Description
bound	null	Point, Point	Boundary of the rectangle.
strokeColor	null	String	Stroke color of the rectangle.
strokeWidth	1	Double	Stroke width of the rectangle.
strokeDash	1	Integer	Stroke dash of the rectangle.
fillColor	null	String	Fill color of the rectangle.
strokeDash	1	Double	Stroke dash of the rectangle.

org.primefaces.model.map.LatLng

Property	Default	Type	Description
lat	null	double	Latitude of the coordinate
lon	null	double	Longitude of the coordinate

org.primefaces.model.map.LatLngBounds

Property	Default	Type	Description
center	nu((6at6n ,	*enter coordinate o+ the boundary
northEast	nu((6at6n ,	<orthEast coordinate o+ the boundary
southWest	nu((6at6n ,	SouthWest coordinate o+ the boundary

GMap Event API

A((c(a e in event a&i e0tend +ro " javax.faces.event.FacesEvent.

org.primefaces.event.map.MarkerDragEvent

Property	Default	Type	Description
"ar3er	nu((8 ar3er	Dra , ,ed "ar3er in tance

org.primefaces.event.map.OverlaySelectEvent

Property	Default	Type	Description
over(ay	nu((7ver(ay	Se(ected over(ay in tance

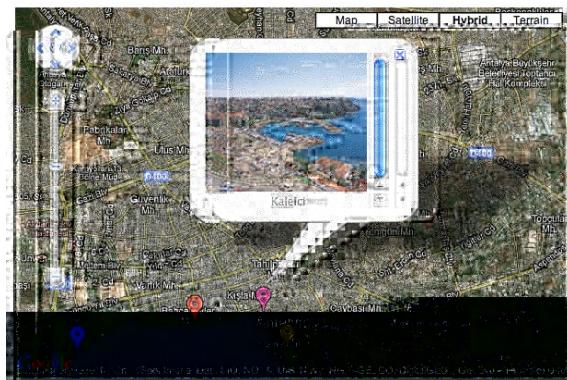
org.primefaces.event.map.PointSelectEvent

Property	Default	Type	Description
(at6n ,	nu((6at6n ,	*oordinate o+ the e(elected &oint

org.primefaces.event.map.StateChangeEvent

3.45 GMapInfoWindow

GMapInfoWindow is used with GMap component to open a window when an overlay is selected.



Info

9a,	gmapInfoWindow
9a, *(a	org.primefaces.component.gmap.GMapInfoWindowTag
*o "&onent *(a	org.primefaces.component.gmap.GMapInfoWindow
*o "&onent 9y&e	org.primefaces.component.GMapInfoWindow
*o "&onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean
width	null	Integer	Specifies the width of the info window

Getting started with GMapInfoWindow

See [Getting started with GMap](#) section for more information about how to use GMapInfoWindow.

3.46 GraphicImage

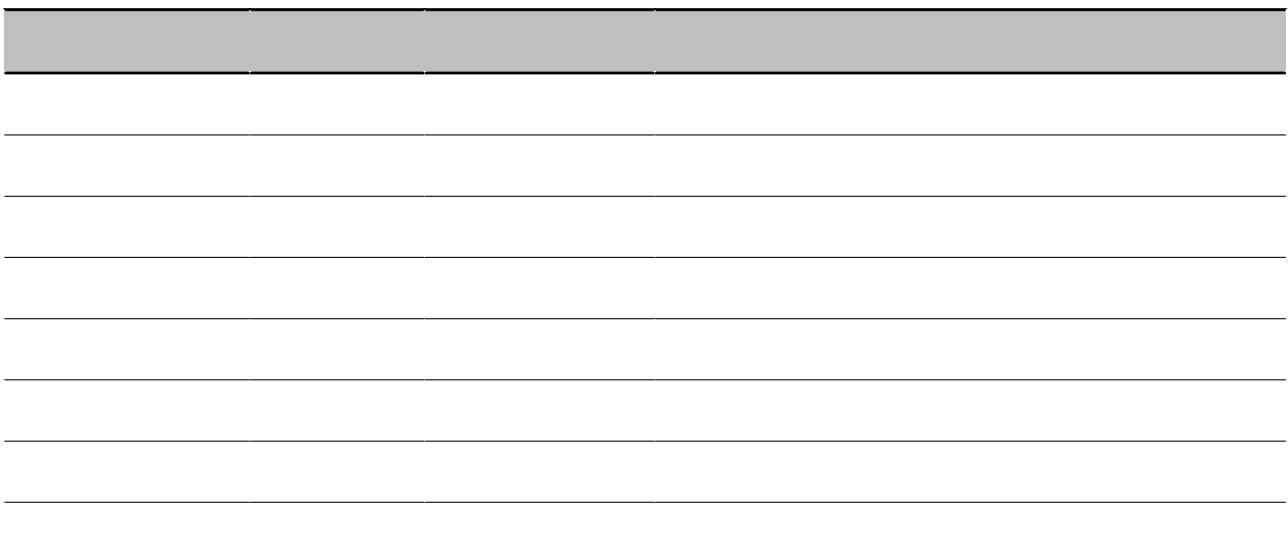
The `GraphicImage` component provides standard support for displaying binary data stored in the database or on the server. It is similar to the `Image` component, but it uses the `GraphicImage` class instead of the `Image` class.

Info

9a,	graphicImage
*o "&onent *(a	org.primefaces.component.graphicimage.GraphicImage
*o "&onent 9y&e	org.primefaces.component.GraphicImage
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.GraphicImageRenderer
Renderer *(a	org.primefaces.component.graphicimage.GraphicImageRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	boolean	Value indicating whether the component is rendered. If set to false, the component will not be rendered.
binding	null	Object	An expression that binds the component to a server-side UI component in a backing bean.
value	null	Object	Binary data to store or convert to a string.
alt	null	String	Alternate text for the image.
uri	null	String	URL to value attribute.
width	null	String	Width of the image.
height	null	String	.height of the image.
title	null	String	Title of the image.
dir	null	String	Direction of the text displayed.
align	null	String	Alignment code.
isAbsolute	false	boolean	Specifies if the URL is absolute.
useAbsolute	null	String	
type	null	String	Style of the image.



Displaying Charts with JFreeChart

See static injection at chart component or a `BackingBean` with #reechart.

Displaying a Barcode

If you want to generate a barcode can be generated a 'high' `BarcodeImageHandler` or the barcode A!

```
<p:graphicImage value="#{backingBean.barcode}" />
```

```
public class BarcodeBean {

    private StreamedContent barcode;

    public BackingBean() {
        try {
            File barcodeFile = new File("dynamicbarcode");
            BarcodeImageHandler.saveJPEG(
                BarcodeFactory.createCode128("PRIMEFACES"), barcodeFile);
            barcode = new DefaultStreamedContent(
                new FileInputStream(barcodeFile), "image/jpeg");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public BarcodeBean getBarcode() {
        return this.barcode;
    }
}
```



Displaying Regular Images

A `GraphicImage` extend standard `Image` component it can also display regular non dynamic images / use standard `Image` component with, name and optional library.

```
<p:graphicImage name="barcalogo.jpg" library="yourapp" />
```

How It Works

Dynamically encode & decode a stream.

Decompress the value stream to generate a key.

The key is appended to the URL that points to the resource handler.

Use the `ResourceHandler` to decompress the stream to the original content and then invoke the instance of `StreamedContent` bean and treat its content to client.

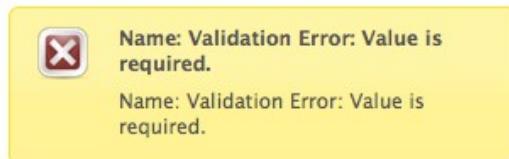
A re u(t there ' i((be % reCue t to di &(ay an i " a,eE+ir t bro' er ' i((" a3e a reCue t to (oad the &a,e and then another one to the dyna " ic i " a,e ur(that &oint to AS# re ource hand(er. !(ea e note that you cannot u e vie ' co&e bean a they are not avai(ab(e in re ource (oadin, reCue t.

Passing Parameters and Data Iteration

You can &a reCue t &ara " eter to the ,ra&hicl " a,e via +N&ara " ta, E a a re u(t the actua(reCue t renderin, the i " a,e can have acce to the e value . 9hi i e0tre " e(y handy to di &(ay dyna " ic i " a,e i+ your i " a,e i in a data iteration co " &onent (i3e datab(e or ui&re&eat.

3.47 Growl

Gro' (i ba ed on the 8 ac' , ro' (notification ' id, et and u ed to di &(ay #ace 8 e a,e in an over(ay.



Info

9a,	growl
*o " &onent *(a	org.primefaces.component.growl.Growl
*o " &onent 9y&e	org.primefaces.component.Growl
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.GrowlRenderer
Renderer *(a	org.primefaces.component.growl.GrowlRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in , bean
tic3y	+a(e	1oo(ean	S&eci+ie i+ the " e a,e hou(d tay in tead o+ hidden auto " atica((y.
ho ' Su " " ary	true	1oo(ean	S&eci+ie i+ the u " " ary o+ " e a,e hou(d be di &(ayed.
ho ' Detai(+a(e	1oo(ean	S&eci+ie i+ the detai(o+ " e a,e hou(d be di &(ayed.
,(oba(7n(y	+a(e	1oo(ean	When trueE on(y +ace " e a,e ' ithout c(ientid are di &(ayed.
(i)e	4000	Inte ,er	Duration in " i((i econd to di &(ay nonB tic3y " e a,e .

Name	Default	Type	Description
autoUpdate	+a(e	Boolean	S&eci+ie auto u&date " ode.
ready &(ay	true	Boolean	De+ine i+ a(ready rendered " e a,ed hou(d be di &(ayed.
+or	null	String,	<a " e o+ a ocated 3eyE ta3e &recedence ' hen u ed ' ith ,(oba(7n(y.
e ca&e	true	Boolean	De+ine ' hether ht" (' ou(d be e ca&ed or not.
everity	null	String,	*o" "a e&arated (i t o+ everitie to di &(ay on(y.

Getting Started with Growl

Gro ' (u a,e i i" i(ar to standard hN" e a,e co " &onent. Si " &(y &(ace ,ro ' (any ' here on your &a,eE ince " e a,e are di &(ayed a an over(ayE the (ocation o+ ,ro ' (in AS# &a,e doe not " atter.

```
<p:growl />
```

Lifetime of messages

1y de+au(t each " e a,e ' i((be di &(ayed +or 4000 " and then hidden. A " e a,e can be " ade tic3y " eanin, it'((never be hidden auto " atica((y.

```
<p:growl sticky="true" />
```

I+ ,ro ' (i not ' or3in, in tic3y " odel it' a(o &o ib(e to tune the duration o+ di &(ayin, " e a,e . #o(o(in, ,ro ' (' i((di &(ay the " e a,e +or 5 econd and then +adeBout.

```
<p:growl life="5000" />
```

Growl with Ajax Updates

I+ you need to di &(ay " e a,e ' ith ,ro ' (a+ter an a/a0 reCue t you /u t need to u&date it. <ote that i+ you enab(e autoU&dateE ,ro ' (' i((be u&dated auto " atica((y ' ith each a/a0 reCue t any ' ay.

```
<p:growl id="messages"/>
<p:commandButton value="Submit" update="messages" />
```

Positioning

Gro ' (i &o itioned at to& ri,ht corner by de+au(tE &o ition can be contro((ed ' ith a *SS eector ca((ed ui-growl. With the be(o ' ettin, ,ro ' (' i((be (ocated at to& (e-t corner.

```
.ui-growl {
    left:20px;
}
```

Targetable Messages

Here you need to target one or more messages to a specific component. You have two options: using the `for` attribute to associate a message with a specific component, or using the `onMessage` attribute to associate a message with a specific component.

```
<p:messages for="somekey" />
<p:growl for="anotherkey" />
```

```
FacesContext context = FacesContext.getCurrentInstance();

context.addMessage("somekey", facesMessage1);
context.addMessage("somekey", facesMessage2);

context.addMessage("anotherkey", facesMessage3);
```

In addition to targeting specific components, you can also target the entire page using the `for` attribute. This allows you to display messages across multiple components on the same page.

Severity Levels

Using the `severity` attribute, you can define which severities can be displayed by the component. For example, you can configure it to only display informational and warning messages.

```
<p:growl severity="info, warn" />
```

Escaping

If you want to include content in the message, you need to escape it via the `escape` attribute.

```
<p:growl escape="false" />
```

Skinning

The skinning of the growl component is based on the PrimeFaces CSS classes.

Style Class	Applies
.ui-growl	Main container element
.ui-growl .ui-growl-content	Content container
.ui-growl .ui-growl-message	Message container
.ui-growl .ui-growl-icon	Icon container
.ui-growl .ui-growl-title	Title container
.ui-growl .ui-growl-detail	Detail container
.ui-growl .ui-growl-severity	Severity icon

Style Class	Applies
.ui-error-block	Info severity icon
.ui-warning-block	Warning severity icon
.ui-error-block	Error severity icon
.ui-success-block	Success severity icon

A warning, type case are, (obviously the "main the" in, action or "more in or" action.

3.48 HotKey

This component is a generic key binding component that can bind any action or key to a specific event handler or action.

Info

9a,	hotkey
*o " &onent *(a	org.primefaces.component.hotkey.HotKey
*o " &onent 9y&e	org.primefaces.component.HotKey
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.HotKeyRenderer
Renderer *(a	org.primefaces.component.hotkey.HotKeyRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify if the component is rendered.
bindin,	null	Boolean	An expression that adds to a server-side UI component in a bean.
bind	null	String	Key binding.
hand(er	null	String	Java script event handler to be executed when the key binding is triggered.
action	null	Method	A method expression that's being evaluated in the script context caused by user input.
actionListener	null	Method	An action listener that's being evaluated in the script context caused by user input.
i " " ediate	!empty	Boolean	Boolean value that determines if the handler is true if the action is triggered at a specific time.
async	!empty	Boolean	When set to true, events are not queued.
&roce	null	String	*o " &onent id? @ to &roce &artia((y instead of ' href').
update	null	String	*(ident id of the component? @ to be updated after a sync &artia((ub) it's rendered).

Name	Default	Type	Description
onstart	null	String	A Java script handler to execute before a/a0 reCue t i be,in .
onco "&(ete	null	String	A Java script handler to execute ' hen a/a0 reCue t i co "&(eted.
onucces	null	String	A Java script handler to execute ' hen a/a0 reCue t ucceed .
onerror	null	String	A Java script handler to execute ' hen a/a0 reCue t +ai(.
,(oba(true	Boolean	Globally a/a0 reCue t are (i tened by a/a0Statu co "&onentE ettin, ,(oba(to +a(e ' i((not tri, ,er a/a0Statu .
de(ay	null	String	Indicates whether delay " i((i econnd e(a& e bet' een ca((to request() on(y the " o t recent one i ent and a((other reCue t are di carded. If thi o&tion i not &eciatedE or i+ the va(ue o+ delay i the (itera trin, (none) ' ithout the Cuote E no de(ay i u ed.
&artia(Sub " it	+a(e	Boolean	Enables eria(i=ation o+ va(ue be(on, in, to the &artia((y &roce ed co "&onent on(y.
re et>a(ue	+a(e	Boolean	If trueE (oca(va(ue o+ in&ut co "&onent to be u&dated ' ithin the a/a0 reCue t ' ou(d be re et.
i,noreAutoU&date	+a(e	Boolean	If trueE co "&onent ' hich autoU&datePItrueI ' i((not be u&dated for thi reCue t. If not &eciatedE or the va(ue i +a(eE no uch indication i " ade.

Getting Started with HotKey

Hotkeys are used in two ways either on client side with the event handler or with a/a0 u&&ort. So to use them you need to bind them to an element.

```
<p:hotkey bind="a" handler="alert('Pressed a');" />
```

When this hotkey is pressed, it will alert the message 'Pressed a'.

Key combinations

For some key combinations you need to use the combination key instead of the key itself.

```
<p:hotkey bind="ctrl+s" handler="alert('Pressed ctrl+s');" />
<p:hotkey bind="ctrl+shift+s" handler="alert('Pressed ctrl+shift+s')"/>
```

Integration

Here's an example of how to integrate hotkeys with a client-side API. In this example, pressing 'ctrl+shift+s' will trigger the 'onctrlshifts' event.

```
<p:hotkey bind="left" handler="PF('switcher').previous();" />
<p:hotkey bind="right" handler="PF('switcher').next();" />

<p:imageSwitch widgetVar="switcher">
    //content
</p:imageSwitch>
```

Ajax Support

Also if a built-in feature of hotkey "beanin, you can do also calling with key combination. #000 in, +or " can be used with the *ctrl+shift+s* combination.

```
<h:form>

    <p:hotkey bind="ctrl+shift+s" update="display" />

    <h:panelGrid columns="2">
        <h:outputLabel for="name" value="Name:" />
        <h:inputText id="name" value="#{bean.name}" />
    </h:panelGrid>

    <h:outputText id="display" value="Hello: #{bean.firstname}" />

</h:form>
```

Note that hotkey ' will not be triggered if there is a focused input element.

3.49 IdleMonitor

Idle monitor attaches user action on a component and notify callback when they are idle or active again.

Info

Value	idleMonitor
onIdle="&onIdle"(a)	org.primefaces.component.idlemonitor.IdleMonitor
onActive="&onActive"(y&e)	org.primefaces.component.IdleMonitor
onBind="&onBind"(a)"i(y)	org.primefaces.component
Renderer="&renderer"*(y&e)	org.primefaces.component.IdleMonitorRenderer
Renderer="&renderer"*(a)	org.primefaces.component.idlemonitor.IdleMonitor

Attributes

Name	Default	Type	Description
id	None	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component when set to false the component will not be rendered.
bindId	None	Object	An object representation that maps to a server-side UI component instance in a backing bean.
idleTime	00000	Integer	The time to wait in seconds until deciding whether the user is idle. Default is 5 seconds.
onIdle	None	String	Script code to execute when the user becomes idle.
onActive	None	String	Script code to execute when the user becomes active again.
onBind	None	String	Script code to execute when the component is bound.

Getting Started with IdleMonitor

To begin, you can hook up to client-side events that are called when a user becomes idle or becomes active again. Example: to trigger visibility of a dialog, around the event.

```
<p:idleMonitor onidle="PF('idleDialog').show();"
               onactive="PF('idleDialog').hide();"/>

<p:dialog header="What's happening?" widgetVar="idleDialog" modal="true">
    <h:outputText value="Dude, are you there?" />
</p:dialog>
```

Controlling Timeout

By default the monitor waits for 5 minutes (50000 ms) until triggered, the timeout event. You can control this duration with the timeout attribute.

Ajax Behavior Events

The monitor provides two ajax behavior events which are *idle* and *active* that are fired according to the timer changes. Each event

```
<p:idleMonitor timeout="5000" update="messages">
    <p:ajax event="idle" listener="#{bean.idleListener}" update="msg" />
    <p:ajax event="active" listener="#{bean.activeListener}" update="msg" />
</p:idleMonitor>

<p:growl id="msg"/>
```

```
public class Bean {

    public void idleListener() {
        //Add facesmessage
    }

    public void idle() {
        //Add facesmessage
    }
}
```

3.50 ImageCompare

The `ImageCompare` component provides a user interface to compare two images.



Info

9a,	imageCompare
*o "&onent *(a	org.primefaces.component.imagecompare.ImageCompare
*o "&onent 9y&e	org.primefaces.component.ImageCompare
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ImageCompareRenderer
Renderer *(a	org.primefaces.component.imagecompare.ImageCompareRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	Unique identifier of the component
rendered	true	booleean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin,	nu((7b/ect	An expression that maps to a server-side UI component in a backing bean
' id, et>ar	nu((Strin,	<a " e of the client side ' id, et.

Name	Default	Type	Description
(e+tl " a,e	nu((Strin,	Source o+ the i " a,e &(aced on the (e+tl ide
ri,htl " a,e	nu((Strin,	Source o+ the i " a,e &(aced on the ri,htl ide
' idth	nu((Strin,	Width o+ the i " a,e
hei,ht	nu((Strin,	. ei,ht o+ the i " a,e
ty(e	nu((Strin,	In(ine ty(e o+ the container e(e " ent
ty(e*(a	nu((Strin,	Sty(e c(a o+ the container e(e " ent

Getting started with ImageCompare

I " a,e*o " &are i created ' ith t ' o i " a,e ' ith a " e hei,ht and ' idth. It i reCuired to et ' idth and hei,ht o+ the i " a,e a ' e(.

```
<p:imageCompare leftImage="xbox.png" rightImage="ps3.png"
width="438" height="246"/>
```

Skinning

1 oth i " a,e are &(aced in ide a div container e(e " ent. *style* and *styleClass* attribute a&&(y to thi e(e " ent.

3.51 ImageCropper

The `ImageCropper` component allows you to crop an image, a certain region of an image. A new image is created containing the cropped area and is saved to a designated folder on the server side.



Info

9a,	imageCropper
*o " &onent *(a	org.primefaces.component.imagecropper.ImageCropper
*o " &onent 9y&e	org.primefaces.component.ImageCropper
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ImageCropperRenderer
Renderer *(a	org.primefaces.component.imagecropper.ImageCropperRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	nu((Object	An expression that refers to a server-side UI component instance in a backing bean
value	nu((Object	Value of the component than can be either an Expression or a (iterable) object
converter	nu((Converter	An expression or a (iterable) object that defines a converter for the component. When it's an Expression, it's resolved to a converter instance. In case it's a static object, it's used to refer to a converter id
immediate	false	Boolean	When set to true, the validation (object) is executed at the time of the value change and the component.
required	false	Boolean	Specifies if the component is required

Name	Default	Type	Description
validator	null	String	A method binding expression that refers to a method validation, the input
valueChangeListener	null	String	A method binding expression that refers to a method or handler, a valuechange event
requiredMessage	null	String	Message to be displayed when required validation failed.
converterMessage	null	String	Message to be displayed when conversion failed.
validatorMessage	null	String	Message to be displayed when validation failed.
clientId	null	String	<a href="#" id="the client id of the element.
immediate	null	String	*onetime(reative &ath to the immediate attribute.
alt	null	String	Alternate text of the image element.
aspectRatio	null	Double	Aspect ratio of the crop&er area.
innerSize	null	String	Inner size of the crop&er area.
outerSize	null	String	Outer size of the crop&er area.
backgroundColor	#000000	String	Background color of the container.
backgroundOpacity	0.4	Double	Background opacity of the container
initialCoordinate	null	String	Initial coordinate of the crop&er area.

Getting started with the ImageCropper

In addition to an input component and an attribute to be cropped provided via the `image` attribute. The cropped area of the original image is used to create a new image. This image can be accessed on the `Cropper` bean by setting the `value` attribute of the `ImageCropper`. A `value` in, the `ImageCropper` at RWE1A!!LR779R0ca "nou./&,

```
<p:imageCropper value="#{cropper.croppedImage}" image="/campnou.jpg" />
```

```
public class Cropper {
    private CroppedImage croppedImage;

    //getter and setter
}
```

`org.primefaces.model.CroppedImage` bean, a !ri " e#ace A !I and contain handy in-or " ation about the cro& &roce . #o((o ' in, tab(e de crie *ro&&edl " a, e &ro&ertie .

Property	Type	Description
originalImage	String	<a href="#" id="the original image that's cropped
byte	byte[]	*ontent of the cropped area as a byte array

Property	Type	Description
(e+ht	int	6e+ht coordinate
ri , ht	int	Ri , ht coordinate
' idth	int	Width o+ the cro&&ed i " a,e
hei , ht	int	. ei , ht o+ the cro&&ed i " a,e

External Images

I " a,e*ro&&er ha the abi(ity to cro& e0terna(i " a,e a ' e(.

```
<p:imageCropper value="#{cropper.croppedImage}"  
image="http://primefaces.prime.com.tr/en/images/schema.png">  
</p:imageCropper>
```

Context Relative Path

#or (oca(i " a,e E I " a,e*ro&&er a(' ay reCuire the i " a,e &ath to be conte0t re(ative. So to acco "&(i h thi i " &(y /u t add (a h ?I&ath&to(i " a,e.&n,I@ and i " a,ecro&&er ' i((reco, ni=e it at RWE1A ! !LR779R&ath&to(i " a,e.&n,. Action ur(re(ative (oca(i " a,e are not u&&orted.

Initial Coordinates

1y de+au(tE u er action i nece ary to initiate the cro&&er area on an i " a,e you can &eci+y an initia(area to di &(ay on &a,e (oad u in, *initialCoords* o&tion in *x,y,w,h* +or " at.

```
<p:imageCropper value="#{cropper.croppedImage}" image="/campnou.jpg"  
initialCoords="225,75,300,125"/>
```

Boundaries

" inSi=e and " a0Si=e attribute are contro(to (i " it the i=e o+ the area to cro&.

```
<p:imageCropper value="#{cropper.croppedImage}" image="/campnou.jpg"  
minSize="50,100" maxSize="150,200"/>
```

Saving Images

1e(o ' i an e0a " &(e to ave the cro&&ed i " a,e to +i(e y te " .

```
<p:imageCropper value="#{cropper.croppedImage}" image="/campnou.jpg" />  
<p:commandButton value="Crop" actionListener="#{myBean.crop}" />
```

```
public class Cropper {  
  
    private CroppedImage croppedImage;  
  
    //getter and setter  
  
    public String crop() {  
        ServletContext servletContext = (ServletContext)  
FacesContext.getCurrentInstance().getExternalContext().getContext();  
        String newFileName = servletContext.getRealPath("") + File.separator +  
"ui" + File.separator + "barca" + File.separator+ croppedImage.getOriginalFileName()  
+ "cropped.jpg";  
  
        FileImageOutputStream imageOutput;  
        try {  
            imageOutput = new FileImageOutputStream(new File(newFileName));  
            imageOutput.write(croppedImage.getBytes(), 0,  
croppedImage.getBytes().length);  
            imageOutput.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        return null;  
    }  
}
```

3.52 ImageSwitch

I " a,e ' itch co " &onent i a i" &(e i " a,e ,a((ery co " &onent.

Previous Next



Info

9a,	imageSwitch
*o " &onent *(a	org.primefaces.component.imageswitch.ImageSwitch
*o " &onent 9y&e	org.primefaces.component.ImageSwitch
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ImageSwitchRenderer
Renderer *(a	org.primefaces.component.imageswitch.ImageSwitchRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
e++ect	nu((Strin,	<a " e o+ the e++ect +or tran ition.
&eed	500	Inte,er	S&eed o+ the e++ect in " i((i econd .
(ide ho ' S&eed)000	Inte,er	S(ide ho ' &eed in " i((i econd .
(ide ho ' Auto	true	1oo(ean	Start (ide ho ' auto " atica((y on &a,e (oad.
ty(e	nu((Strin,	Sty(e o+ the " ain container.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the " ain container.

Getting Started with ImageSwitch

I " a,eS ' itch co " &onent need a et o+ i " a,e to di &(ay. !rovide the i " a,e co((ection a a et o+ chi(dren co " &onent .

```
<p:imageSwitch effect="FlyIn">
    <p:graphicImage value="/images/nature1.jpg" />
    <p:graphicImage value="/images/nature2.jpg" />
    <p:graphicImage value="/images/nature3.jpg" />
    <p:graphicImage value="/images/nature4.jpg" />
</p:imageSwitch>
```

8 o to+ the ti " eE i " a,e cou(d be dyna " icE uiNre&eat i u&&orted to i " &(e " ent thi ca e.

```
<p:imageSwitch>
    <ui:repeat value="#{bean.images}" var="image">
        <p:graphicImage value="#{image}" />
    </ui:repeat>
</p:imageSwitch>
```

Slideshow or Manual

I " a,eS ' itch i in (ideSho ' " ode by de+au(tE i+ you'd (i3e " anua(tran ition di ab(e (ide ho ' and u e c(ient ide a&i to create contro(.

```
<p:imageSwitch effect="FlyIn" widgetVar="imageswitch">
    //images
</p:imageSwitch>

<span onclick="PF('imageswitch').previous();">Previous</span>
<span onclick="PF('imageswitch').next();">Next</span>
```

Client Side API

Wid, et\ PrimeFaces.widget.ImageSwitch

Method	Params	Return Type	Description
startSlide(@?)	B	void	Start (ide ho ' " ode.
stopSlide(@?)	B	void	Stop (ide ho ' " ode.
to, ,(@)slide(@?)	B	void	Go , ,(e (ide ho ' " ode.
&autoSlide(@?)	B	void	!au e (ide ho ' " ode.
nextSlide()	B	void	S ' itche to ne0t i " a,e.
&reviewSlide()	B	void	S ' itche to &review i " a,e.
' itch90?index0()	inde0	void	Di &(ay i " a,e ' ith ,iven inde0.

Effect Speed

The speed is considered in terms of the second and specified via the speed attribute.

```
<p:imageSwitch effect="FlipOut" speed="150">
    //set of images
</p:imageSwitch>
```

List of Effects

The following list contains all available transition effects. Note that values are case sensitive.

- ↳ blindF
- ↳ blindM
- ↳ blindU
- ↳ cover
- ↳ curtainF
- ↳ curtainM
- ↳ fade
- ↳ fadeUoo "
- ↳ flipF
- ↳ flipM
- ↳ none
- ↳ crossU&
- ↳ crossD ' n
- ↳ cross6e+t
- ↳ crossRi,ht
- ↳ crossGert
- ↳ hue(e
- ↳ hideF
- ↳ hideM
- ↳ to
- ↳ turnU&
- ↳ turnDo ' n
- ↳ turn6e+t
- ↳ turnRi,ht
- ↳ uncover
- ↳ ' i&e
- ↳ =oo "

3.53 Inplace

Inplace provides a way to edit content directly. It contains the initial content and the hidden content that is displayed when the edit button is clicked.

Basic Input: Edit Me

Basic Input:

Info

9a,	inplace
*o "&onent *(a	org.primefaces.component.inplace.Inplace
*o "&onent 9y&e	org.primefaces.component.Inplace
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.InplaceRenderer
Renderer *(a	org.primefaces.component.inplace.InplaceRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify whether the component is rendered or not.
binding	nu((Object	An expression that maps to a server-side UI component instance in a backing bean.
' id, et>ar	nu((String,	ar
(abe(nu((String,	Value to be bound in the attribute.
e "&ty6abe(nu((String,	Value to be bound in the attribute.
effect	+ade	String,	Effect to be used when transitioning.
effectSeed	nor " a(String,	Seed of the effect.
disabled	+a(e	Boolean	Prevent hidden content to be bound.
type	nu((String,	Type of the main container element.
type*(a	nu((String,	Style class of the main container element.
editor	+a(e	Boolean	Specifies the editor mode.

Name	Default	Type	Description
ave6abe(Save	String,	9oo(ti& te0t o+ ave buttin in editor " ode.
cance6abe(* ance(String,	9oo(ti& te0t o+ cance(buttin in editor " ode.
event	c(ic3	String,	<a " e o+ the c(ient ide event to di &(ay in(ine content.
to,,(eab(e	true	Boolean	De+ine i+ in&(ace i to,,(eab(e or not.

Getting Started with Inplace

The in(ine co " &onent need to be a chi(d o+ in&(ace.

```
<p:inplace>
    <h:inputText value="Edit me" />
</p:inplace>
```

Custom Labels

1y de+au(t in&(ace di &(ay it +ir t chi(d va(ue a the (abe(E you can cu to " i=e it via the (abe(attribute.

```
<h:outputText value="Select One:" />

<p:inplace label="Cities">
    <h:selectOneMenu>
        <f:selectItem itemLabel="Istanbul" itemValue="Istanbul" />
        <f:selectItem itemLabel="Ankara" itemValue="Ankara" />
    </h:selectOneMenu>
</p:inplace>
```

Select One: Cities

Select One: Istanbul ▾

Facets

#or advanced cu to " i=ationE output and input +acet are &rovided.

```
<p:inplace id="checkboxInplace">
    <f:facet name="output">
        Yes or No
    </f:facet>
    <f:facet name="input">
        <h:selectBooleanCheckbox />
    </f:facet>
</p:inplace>
```

Effects

Default effect is *fade* and other options like effect is *slide*. Any effect speed can be tuned with value *slow*, *normal* and *fast*.

```
<p:inplace label="Show Image" effect="slide" effectSpeed="fast">
    <p:graphicImage value="/images/nature1.jpg" />
</p:inplace>
```

Editor

Inline editing is enabled via the *editor* option.

```
public class InplaceBean {
    private String text;
    //getter-setter
}
```

```
<p:inplace editor="true">
    <h:inputText value="#{inplaceBean.text}" />
</p:inplace>
```



save and *cancel* are two provided ajax behavior events you can use to hook in the editing process.

```
public class InplaceBean {
    private String text;
    public void handleSave() {
        //add faces message with update text value
    }
    //getter-setter
}
```

```
<p:inplace editor="true">
    <p:ajax event="save" listener="#{inplaceBean.handleSave}" update="msgs" />
    <h:inputText value="#{inplaceBean.text}" />
</p:inplace>

<p:growl id="msgs" />
```

Client Side API

Wid, et^N PrimeFaces.widget.Inplace

Method	Params	Return Type	Description
show()	B	void	Show content and hide dialog element.
hide()	B	void	Show dialog element and hide content.
toogle()	B	void	Show visibility of content and hide dialog element.
refresh()	B	void	Refreshes the content to process inline updates.
cancel()	B	void	Refreshes the content to revert inline updates.

Skinning

In addition to the standard style and styleClass options available in the input structure type, there are specific styles for the Inplace component.

Style Class	Applies
.ui-inplace	On container element.
.ui-inplace-above	On container element when dialog is above.
.ui-inplace->	On dialog element.
.ui-inplace-content	On content element.
.ui-inplace-editor	Editor control container.
.ui-inplace-save	Save button.
.ui-inplace-cancel	Cancel button.

A skinning, type class are, (obviously) on the "main" in, action or "more info" action.

3.54 InputMask

Input 8 a 3 force an input to fit in a defined " a 3 te " &(ate.

Date:	<input type="text" value="11/12/2010"/>
Phone:	<input type="text" value="(523) 453-4253"/>
Phone with Ext:	<input type="text" value="(234) 532-4524 x35254"/>
taxId:	<input type="text" value="52-3434234"/>
SSN:	<input type="text" value="234-52-3452"/>
Product Key:	<input type="text" value="____-____-_____"/>

Info

9a,	inputMask
*o " &onent *(a	org.primefaces.component.inputmask.InputMask
*o " &onent 9y&e	org.primefaces.component.InputMask
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.InputMaskRenderer
Renderer *(a	org.primefaces.component.inputmask.InputMaskRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component
rendered	true	Boolean	True (value to identify the rendering, or the component) 'hen it is true the component ' i((not be rendered.
binding	null	Object	An expression that "a& to a server side UI *o " &onent in tance in a bac3in, bean
" a 3	null	String	8 a 3 te " &(ate
&(ace . o(der	null	String	!ace . o(der in " a 3 te " &(ate.
value	null	Object	>a(ue or the component than can be either an E6 expression or a (itera(te0t

Name	Default	Type	Description
converter	nu((*onverter)Strin,	An e(e0&re ion or a (itera te0t that define a converter +or the co " &onent. When it' an E6 e0&re ion it' re oved to a converter in tance. In ca e it' a static te0t it " u t re+er to a converter id
i " " ediate	+a(e	1oo(ean	When et trueE &roce va(idation (o, ic i e0ecuted at a&(y reCue t va(ue &ha e +or thi co " &onent.
reCuired	+a(e	1oo(ean	8 ar3 co " &onent a reCuired
va(idator	nu((8 ethodE0&r	A " ethod bindin, e0&re ion that re+er to a " ethod va(idation , the in&ut
va(ue*han,e6i tener	nu((8 ethodE0&r	A " ethod bindin, e0&re ion that re+er to a " ethod +or hand(in , a va(uchan,eevent
reCuired 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen reCuired +ie(d va(idation +ai(.
converter 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen va(idation +ie(d .
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
acce 3ey	nu((Strin,	Acce 3ey that ' hen &re ed tran +ocu to the in&ut e(e " ent.
a(t	nu((Strin,	A(ternate te0tua(de cri&tion o+ the in&ut +ie(d.
autoco " &(ete	nu((Strin,	*ontro(bro ' er autoco " &(ete behavior.
dir	nu((Strin,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
di ab(ed	+a(e	1oo(ean	Di ab(e in&ut +ie(d
(abe(nu((Strin,	A (oca(i=ed u er &re entab(e na " e.
(an,	nu((Strin,	*ode de cribin, the (an,ua,e u ed in the ,enerated " ar3u&+or thi co " &onent.
" a0(en,th	nu((Inte ,er	8 a0i " u " nu " ber o+ character that " ay be entered in thi +ie(d.
onb(ur	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu .
onchan,e	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu and it va(ue ha been " odified ince ,ainin, +ocu .
onc(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i c(ic3ed.
ondb(c(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i doub(e c(ic3ed.

Name	Default	Type	Description
on+ocu	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent receive +ocu .
on3eydo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over in&ut e(e " ent.
on3ey&re	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over in&ut e(e " ent.
on3eyu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over in&ut e(e " ent.
on " ou edo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over in&ut e(e " ent
on " ou e " ove	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin in&ut e(e " ent.
on " ou eout	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " in&ut e(e " ent.
on " ou eover	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on " ou eu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over in&ut e(e " ent.
on e(ect	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
&(aceho(der	nu((Strin,	S&eci+ie a hort hint.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
i=e	nu((Inte,er	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
ty(e	nu((Strin,	In(ine ty(e o+ the in&ut e(e " ent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the in&ut e(e " ent.
tabinde0	nu((Inte,er	!o ition o+ the in&ut e(e " ent in the tabbin , order.
tit(e	nu((Strin,	Advi ory too(ti& in+or " aton.

Getting Started with InputMask

In&ut 8 a 3 be(o ' en+orce in&ut to be in 99/99/9999 date +or " at.

```
<p:inputMask value="#{bean.field}" mask="99/99/9999" />
```

Mask Samples

Here are some examples based on different types of masks:

```
<h:outputText value="Phone: " />
<p:inputMask value="#{bean.phone}" mask="(999) 999-9999"/>

<h:outputText value="Phone with Ext: " />
<p:inputMask value="#{bean.phoneExt}" mask="(999) 999-9999? x99999"/>

<h:outputText value="SSN: " />
<p:inputMask value="#{bean.ssn}" mask="999-99-9999"/>

<h:outputText value="Product Key: " />
<p:inputMask value="#{bean.productKey}" mask="a*-999-a999"/>
```

Skinning

The `style` and `styleClass` options apply to the input field itself. A skinning, typically achieved by overriding the "input" class, can be applied to the entire component.

3.55 InputText

In this section we will learn how to standard input component with binding, validation, and conversion.

Info

9a,	inputText
*o " &onent *(a	org.primefaces.component.inputtext.InputText
*o " &onent 9y&e	org.primefaces.component.InputText
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.InputTextRenderer
Renderer *(a	org.primefaces.component.inputtext.InputTextRender

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify whether the component is rendered or not.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
value	null	Object	>a value of the component than can be either an E6 expression or a (itera) teot
converter	null	Converter	An expression or a (itera) teot that defines a converter for the component. When it's an E6 expression, it refers to a converter instance. In case it's a static teot, it refers to a converter id.
immediate	false	Boolean	When set true, validation (binding) is executed at a specific time and for this component.
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method binding, expression that refers to a " method validation, the input
valueChangeListener	null	Method	A method binding, expression that refers to a " method for handing, a valuechangeevent
requiredMessage	null	String	The message to be displayed if the value is required.

Name	Default	Type	Description
converter	none	String	A converter that converts the value to a String.
validator	none	String	A validator that validates the value.
id	none	String	The ID of the component.
accesskey	none	String	An access key that is used to identify the component.
alt	none	String	An alternate text for the component.
autoComplete	none	String	An auto-complete behavior.
dir	none	String	Direction indication for the component.
disabled	false	Boolean	Whether the component is disabled.
(abnormal)	none	String	An abnormal behavior.
(an,)	none	String	An annotation.
"a0(en,th)	none	Integer	The character that should be entered in this field.
onb(ur)	none	String	An onblur event.
onchan,e	none	String	An onchange event.
onc(ic3)	none	String	An onclick event.
ondb(c(ic3)	none	String	An ondblclick event.
on+ocu	none	String	An onfocus event.
on3eydo 'n	none	String	An onmousedown event.
on3ey&re	none	String	An onmousemove event.
on3eyu&	none	String	An onmouseup event.
on " ou edo ' n	none	String	An onmouseover event.
on " ou e " ove	none	String	An onmouseout event.
on " ou eout	none	String	An onmousemove event.

Name	Default	Type	Description
on " ou eover	null	String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on " ou eu&	null	String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over in&ut e(e " ent.
on e(ect	null	String,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
&(aceho(der	null	String,	S&eci+ie a hort hint.
readon(y	0	Boolean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
i=e	null	Integer	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
ty(e	null	String,	In(ine ty(e o+ the in&ut e(e " ent.
ty(e *(a	null	String,	Sty(e c(a o+ the in&ut e(e " ent.
tabinde0	null	Integer	!o ition o+ the in&ut e(e " ent in the tabbin , order.
tit(e	null	String,	Advi ory too(ti& in+or " aton.
ty&e	true	String,	In&ut +ie(d ty&e.

Getting Started with InputText

In&ut9e0t u a,e i a " e a standard in&ut9e0t

```
<p:inputText value="#{bean.propertyName}" />
```

```
public class Bean {
    private String propertyName;
    //getter and setter
}
```

Client Side API

Wid, et\ PrimeFaces.widget.InputText

Method	Params	Return Type	Description
enable()	Boolean	void	Enab(e the in&ut +ie(d.
disable()	Boolean	void	Di ab(e the in&ut +ie(d.

3.56 InputTextarea

InputTextarea is an extension to standard textarea with autoresize and character counter and the "min" feature.



Info

9a,	inputTextarea
*o " &onent *(a	org.primefaces.component.inputtextarea.InputTextarea
*o " &onent 9y&e	org.primefaces.component.InputTextarea
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.InputTextareaRenderer
Renderer *(a	org.primefaces.component.inputtextarea.InputTextareaRender

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression that binds to a server-side UI component in a backing bean
value	null	Object	The value of the component than can be either an Expression or a converter
converter	null	Converter	An expression or a converter that defines a converter for the component. When it's an Expression, it's resolved to a converter in advance. In case it's a static text, it's used to refer to a converter id
immediate	false	Boolean	When set true, validation (optional) is executed at the same time as the component's value is set.
required	false	Boolean	Specifies if the component is required

Name	Default	Type	Description
validator	null	String	A " method binding expression that refers to a " method validation, the input
valueChangeListener	null	String	A " method binding expression that refers to a " method for hand(in, a valuechange event
required	null	String	String to be displayed when required +ie(d validation +ai(.
converter	null	String	String to be displayed when conversion +ai(.
validator	null	String	String to be displayed when validation +ie(d .
id,et>ar	null	String	<a " e o+ the client id,et.
acce 3ey	null	String	Access key that ' hen &re ed translation +ocu to the input element.
alt	null	String	Alternate text for the creation of the input +ie(d.
autoComplete	null	String	*ontrols browser autocomplete behavior.
dir	null	String	Direction indication +or teot that does not inherit directionality. >a(id value are R69 and R96.
disabled	+able	Boolean	Disabled input +ie(d
(abe)	null	String	A (ocused user &re entab(e na " e.
(an,	null	String	*ode de cribin, the (an,ua,e used in the ,enerated " ar3u& +or thi co " &onent.
onblur	null	String	*(ient ide ca((bac3 to execute ' hen input element (o e +ocu .
onchan,e	null	String	*(ient ide ca((bac3 to execute ' hen input element (o e +ocu and it value has been " odi+ied since ,ainin, +ocu .
onclick	null	String	*(ient ide ca((bac3 to execute ' hen input element (o e +ocu .
ondoubleclick	null	String	*(ient ide ca((bac3 to execute ' hen input element (o e +ocu .
onfocus	null	String	*(ient ide ca((bac3 to execute ' hen input element (o e +ocu .
on3eydo ' n	null	String	*(ient ide ca((bac3 to execute ' hen a 3ey is &re ed do ' n over input element (o e +ent.
on3ey&re	null	String	*(ient ide ca((bac3 to execute ' hen a 3ey is &re ed and re(ea ed over input element (o e +ent.
on3eyu&	null	String	*(ient ide ca((bac3 to execute ' hen a 3ey is re(ea ed over input element (o e +ent.
on " ou edo ' n	null	String	*(ient ide ca((bac3 to execute ' hen a &ointer button is

Name	Default	Type	Description
			&re ed do ' n over in&ut e(e " ent
on " ou e " ove	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin in&ut e(e " ent.
on " ou eout	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " in&ut e(e " ent.
on " ou eover	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on " ou eu&	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over in&ut e(e " ent.
on e(ect	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
i=e	nu((Inte ,er	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
ty(e	nu((Strin ,	In(in e ty(e o+ the in&ut e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the in&ut e(e " ent.
tabinde0	nu((Inte ,er	!o ition o+ the in&ut e(e " ent in the tabbin , order.
tit(e	nu((Strin ,	Advi ory too(ti& in+or " aton.
autoRe i=e	true	1oo(ean	S&eci+ie auto ,ro ' in , ' hen bein, ty&ed.
" a0(en ,th	nu((Inte ,er	8 a0i " u " nu " ber o+ character that " ay be entered in thi +ie(d.
counter	nu((Strin ,	Id o+ the out&ut co " &onent to di &(ay re " ainin , char .
counter9e " &(ate	VOW	Strin ,	9e " &(ate te0t to di &(ay in counter.
co " &(ete 8 ethod	nu((8 ethod E0&r	8 ethod to &rovide u , ,e tion .
" iDuery6en ,th)	Inte ,er	<u " ber o+ character to be ty&ed to run a Cuery.
CueryDe(ay	500	Inte ,er	De(ay in " be+ore endin , each Cuery.
cro((. ei ,ht	nu((Inte ,er	. ei ,ht o+ the vie ' &ort +or autoco " &(ete u , ,e tion .

Getting Started with InputTextarea

In & ut9e0tarea u a, e i a " e a standard in & ut9e0tarea

```
<p:inputTextarea value="#{bean.propertyName}" />
```

AutoResize

Whi(e te0tarea i bein, ty&edE i+ content hei , ht e0ceed the a((ocated &aceE te0tarea can , ro ' auto " atica((y. U e autoRe i=e o&tion to turn on/off thi +eature.

```
<p:inputTextarea value="#{bean.propertyName}" autoResize="true|false"/>
```

Remaining Characters

In & ut9e0tarea can (i " it the " a0i " u " a((o ' ed character ' ith " a06en, th o&tion and di &(ay the re " ainin, character count a ' e(.

```
<p:inputTextarea value="#{bean.propertyName}" counter="display"
    maxlength="20" counterTemplate="{0} characters remaining" />
<h:outputText id="display" />
```

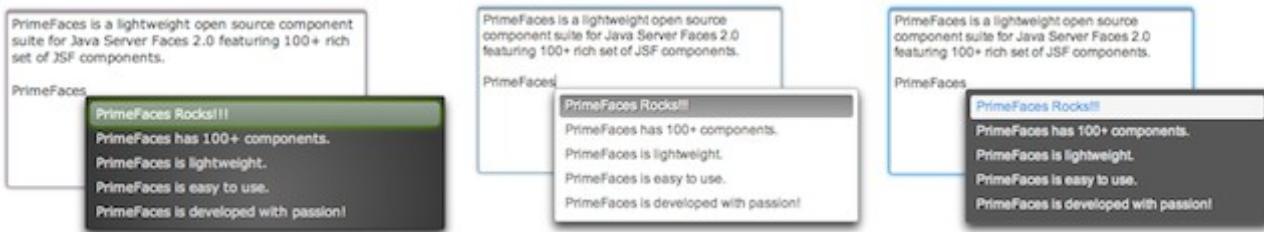
AutoComplete

In & ut9e0tarea u&&ort a/a0 autoco " &(ete +unctiona(ity a ' e(. Mou need to &rovide a co " &(ete 8 ethod to &rovide the u,,e tion to u e thi +eature. In a " &(e be(o ' E co " &(eteArea " ethod ' i((be inv03ed ' ith the Cuery bein, the +our character be0re the caret.

```
public class AutoCompleteBean {
    public List<String> completeArea(String query) {
        List<String> results = new ArrayList<String>();

        if(query.equals("PrimeFaces")) {
            results.add("PrimeFaces Rocks!!!");
            results.add("PrimeFaces has 100+ components.");
            results.add("PrimeFaces is lightweight.");
            results.add("PrimeFaces is easy to use.");
            results.add("PrimeFaces is developed with passion!");
        }
        else {
            for(int i = 0; i < 10; i++) {
                results.add(query + i);
            }
        }
        return results;
    }
}
```

```
<p:inputTextarea rows="10" cols="50" minQueryLength="4"
    completeMethod="#{autoCompleteBean.completeArea}" />
```



Skinning

In addition to the standard style and styleClass options available in the inputTextarea structure, there are several additional style classes that can be used to further customize the appearance of the component.

Style Class	Applies
ui-autocomplete	Autocomplete dropdown.
ui-autocomplete-item	Autocomplete item.
.ui-autocomplete .list	Autocomplete list.
.ui-autocomplete .list-item	Autocomplete list item.
.ui-autocomplete .list-item.ui-state-highlight	Autocomplete list item currently selected.

A summary of the available style classes is provided in the table below.

3.57 Keyboard

A keyboard is an input component that uses a virtual keyboard to provide the input. Features include the ability to change layout and switch between numeric, alphanumeric, and special character modes.



Info

9a,	keyboard
*o " &onent *	



Name	Default	Type	Description
va(idator	null	String	A " method binding, returning a reference to a " method va(idation, the input
va(ue*han, e6i tener	null	String	A " method binding, returning a reference to a " method +or hand(in, a va(uchan, event
reCured 8 e a,e	null	String,	8 e a,e to be displayed when required instead of va(idation +ai(.
converter 8 e a,e	null	String,	8 e a,e to be displayed instead of conversion +ai(.
va(idator 8 e a,e	null	String,	8 e a,e to be displayed instead of va(idation +ai(d .
&a ' ord	+a(e	Boolean	8 a3e the input a &a ' ord instead of d.
ho ' 8 ode	+ocu	String,	Specifies the how ' 8 odes +ocu 'E 'button' 'both'
buttonI " a,e	null	String,	I " a,e +or the button.
buttonI " a,e7n(y	+a(e	Boolean	When set to true on(y i " a,e or the button ' ou(d be displayed instead of a y.
e+ect	+adeln	String,	E+ect of the display action.
e+ectDuration	null	String,	Length of the display action.
(ayout	C ' erty	String,	Layout of the keyboard.
(ayout9e " &(ate	null	String,	9e " &(ate of the current layout.
3ey&ad 7n(y	+ocu	Boolean	Specifies display of a keyboard instead of a keyboard.
&ro " &t6abe(null	String,	6abe(or the &ro " &t te0t.
c(o e6abe(null	String,	6abe(or the c(o e 3ey.
c(ear6abe(null	String,	6abe(or the c(ear 3ey.
bac3 &ace6abe(null	String,	6abe(or the bac3 &ace 3ey.
acce 3ey	null	String,	Access 3ey that when &re ed translates +ocu to the input element.
a(t	null	String,	Alternate text for description of the input +ie(d.
autoco " &(ete	null	String,	*ontrols browser's auto " &(ete behavior.
dir	null	String,	Direction indication +or te0t that does not inherit directionality. >aid va(ue are 69R and R96.
di ab(ed	+a(e	Boolean	Disable input +ie(d
(abe(null	String,	Allows(ed user &re entab(ed na " e.
(an,	null	String,	*ode de cribin, the (an, ua, e u ed in the generated " ar3u& +or thi co " &onent.

Name	Default	Type	Description
" a0(en,th	nu((Inte,er	8 a0i " u " nu " ber o+ character that " ay be entered in thi +ie(d.
onb(ur	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu .
onchan,e	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu and it va(ue ha been " odi+ied ince ,ainin, +ocu .
onc(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i c(ic3ed.
ondb(c(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i doub(e c(ic3ed.
on+ocu	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent receive +ocu .
on3eydo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over in&ut e(e " ent.
on3ey&re	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over in&ut e(e " ent.
on3eyu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over in&ut e(e " ent.
on " ou edo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over in&ut e(e " ent
on " ou e " ove	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin in&ut e(e " ent.
on " ou eout	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " in&ut e(e " ent.
on " ou eover	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on " ou eu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over in&ut e(e " ent.
on e(ect	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
&(aceho(der	nu((Strin,	S&eci+ie a hort hint.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
i=e	nu((Inte,er	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
ty(e	nu((Strin,	In(ine ty(e o+ the in&ut e(e " ent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the in&ut e(e " ent.

Name	Default	Type	Description
tabindex	null	Integer	The position of the input element in the tabbing order.
title	null	String	Alternative text for the input element.
clientId	null	String	The client-side ID of the input element.

Getting Started with Keyboard

The keyboard is used to type text and is initialized by default when the input field gains focus. A keyboard is displayed when the input field has focus.

```
<p:keyboard value="#{bean.value}" />
```

Built-in Layouts

There are a couple built-in keyboard layouts: 'Cyrillic' and 'Alphabetic'. For example, the keyboard below has the Alphabetic layout.

```
<p:keyboard value="#{bean.value}" layout="alphabetic"/>
```



Custom Layouts

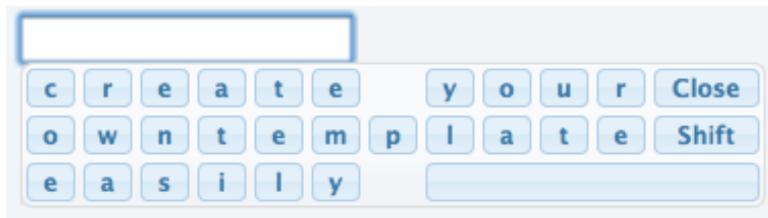
The keyboard has a very flexible layout mechanism allowing you to change its layout according to your own needs.

```
<p:keyboard value="#{bean.value}"
    layout="custom"
    layoutTemplate="prime-back,faces-clear,rocks-close"/>
```



Another example:

```
<p:keyboard value="#{bean.value}"
    layout="custom"
    layoutTemplate="create-space-your-close,owntemplate-shift,easily-space-spacebar"/>
```



A (ayout te " &(ate ba ica((y con i t o+ bui(tBn 3ey and your o ' n 3ey . #o((o ' in, i the (i t o+ a((bui(tBn 3ey .

- G bac3
- G c(ear
- G c(o e
- G hi+t
- G &acebar
- G &ace
- G ha(+) &ace

A((other te0t in a (ayout i rea(i=ed a e&erate 3ey o I&ri "eI ' ou(d create 5 3ey a I&I IrI IiI I " I IeI. U e da h to e&erate each " e " ber in (ayout and u e co " " a to create a ne ' ro '.

Keypad

1y de+au(t 3eyboard di &(ay ' ho(e 3ey E i+ you on(y need the nu"ber u e the 3ey&ad " ode.

```
<p:keyboard value="#{bean.value}" keypadOnly="true"/>
```

ShowMode

9here're a cou&(e o+ di+erent 'ay to di &(ay the 3eyboard by de+au(t 3eyboard i ho 'n once in&ut +ie(d receive the +ocu . 9hi i cu to " i=ed u in, the ho ' 8 ode +eature 'hich acce&t va(ue '+ocu E 'button' E 'both'. ;eyboard be(o ' di &(ay a button ne0t to the in&ut +ie(dE 'hen the button i c(ic3ed the 3eyboard i ho 'n.

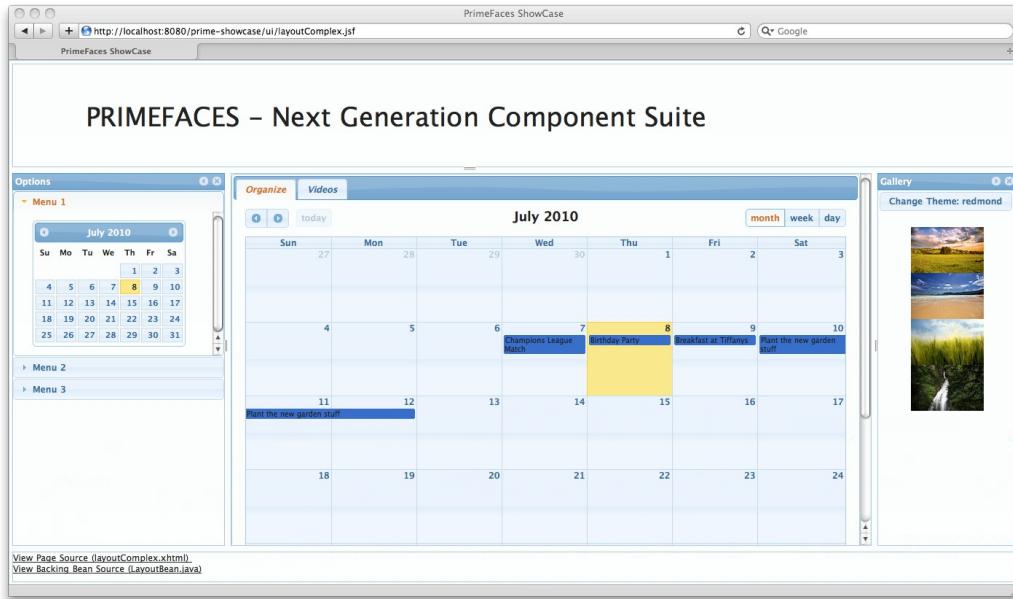
```
<p:keyboard value="#{bean.value}" showMode="button"/>
```



1utton can a(o be cu to " i=ed u in, the *buttonImage* and *buttonImageOnly* attribute .

3.58 Layout

Layout component feature a hierarchical layout mode, it very easy to create complex layout even if you're not familiar with 'eb de in.



Info

9a,	layout
*o " &onent *(a	org.primefaces.component.layout.Layout
*o " &onent 9y&e	org.primefaces.component.Layout
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.LayoutRenderer
Renderer *(a	org.primefaces.component.layout.LayoutRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Logical value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An object reference that maps to a server-side UI component instance in a backing bean.
' id, et>ar	null	String	The ID of the client-side element.
+u(!a,e	false	Boolean	Specifies whether the layout should be rendered or not.

Name	Default	Type	Description
ty(e	nu((String,	Style to apply to container element. This is only applied if the element has a style class.
ty(e*(a	nu((String,	Style class to apply to container element. This is only applied if the element has a style class.
onRe <i>e</i> =e	nu((String,	*(ident)ide call back to execute when a button is clicked.
on*(o <i>e</i>	nu((String,	*(ident)ide call back to execute when a button is clicked.
on9o,,(e	nu((String,	*(ident)ide call back to execute when a button is clicked.
re <i>e</i> =9it(e	nu((String,	9it(e) (abnormal) for the re <i>e</i> button.
co((a&e9it(e	nu((String,	9it(e) (abnormal) for the co((a&e button.
e0&and9it(e	nu((String,	9it(e) (abnormal) for the e0&and button.
c(o <i>e</i> 9it(e	nu((String,	9it(e) (abnormal) for the c(o <i>e</i> button.

Getting started with Layout

Layout is based on a border6layout "ode that contains 5 different layout units which are top, center, left, right and bottom". Each "ode" is visible in the source code as follows:



Full Page Layout

Layout has two "ode" you can either use it or a +u((&a, e (layout or +or a &ecific region in your &a, e. This is often used with the +u((!a, e attribute which is +a(e by default.

The regions in a (layout are defined by (layoutUnit +o((o 'in, i a i" &(e +u((&a, e (layout 'ith a((&o ib(e unit . Note that you can place any content in each (layout unit.

```
<p:layout fullPage="true">
    <p:layoutUnit position="north" size="50">
        <h:outputText value="Top content." />
    </p:layoutUnit>
    <p:layoutUnit position="south" size="100">
        <h:outputText value="Bottom content." />
    </p:layoutUnit>
    <p:layoutUnit position="west" size="300">
        <h:outputText value="Left content" />
    </p:layoutUnit>
    <p:layoutUnit position="east" size="200">
        <h:outputText value="Right Content" />
    </p:layoutUnit>
    <p:layoutUnit position="center">
        <h:outputText value="Center Content" />
    </p:layoutUnit>
</p:layout>
```

Forms in Full Page Layout

When 'or3in, 'ith +or" and +u((&a,e (ayout avoid u in, a +or" that contain (ayoutunit a ,enerated do " "ay not be the a" e. So +o((o ' in, i **invalid**.

```
<p:layout fullPage="true">
    <h:form>
        <p:layoutUnit position="west" size="100">
            <h:outputText value="Left Pane" />
        </p:layoutUnit>
        <p:layoutUnit position="center">
            <h:outputText value="Right Pane" />
        </p:layoutUnit>
    </h:form>
</p:layout>
```

A (ayout unit "u t have it' o 'n +or" in teadE a(o avoid tryin, to u&date (ayout unit becau e o+ a" e rea onE u&date it' content in tead.

Dimensions

E0ce&t center (ayoutUnit other (ayout unit **must** have di " en ion de+ined via *size* o&tion.

Element based layout

Another u e ca e o+ (ayout i the e(e" ent ba ed (ayout. 9hi i the de+au(t ca e actua((y o /u t i,nore +u((!a,e attribute or et it to +a(e. 6ayout e0a "&(e be(o ' de " on trate creatin, a &(it &ane(i " &(e " entation.

```
<p:layout style="width:400px;height:200px">

    <p:layoutUnit position="west" size="100">
        <h:outputText value="Left Pane" />
    </p:layoutUnit>

    <p:layoutUnit position="center">
        <h:outputText value="Right Pane" />
    </p:layoutUnit>

    //more layout units

</p:layout>
```

Ajax Behavior Events

Layout provides a way to attach behavior events to each layout unit.

Event	Listener Parameter	Fired
to, , (e)	or, .&ri "e+ace .event.9o, ,(eEvent	When a unit is expanded or collapsed.
close (e)	or, .&ri "e+ace .event.*((o eEvent	When a unit is closed.
re size (e)	or, .&ri "e+ace .event.Re size(eEvent	When a unit is resized.

Stateful Layout

Layout units can be updated once you create your data to store the user reference. You can update this data using a behavior event (injected by layout). #or e0a "&(e i+ a layout unit is collapsed) you can move and set its position. By binding, this user is tied in position to the collapsed attribute of the layout unit (layout ' will be rendered as the user sets it (at time).

Client Side API

Widget *PrimeFaces.widget.Layout*

Method	Params	Return Type	Description
to, , (e)?&option	&option	void	9o, , (e) (layout unit).
show ?&option	&option	void	Show (layout unit).
hide?unit	&option	void	. hide (layout unit).

Skinning

#o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB(layout	8 ain ' ra&&er container e(e " ent
.uiB(layoutBdoc	6ayout container
.uiB(layoutBunit	Each (ayout unit container
.uiB(layoutBV&o itionW	!o ition ba ed (ayout unit
.uiB(layoutBunitBheader	6ayout unit header
.uiB(layoutBunitBcontent	6ayout unit body

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or " ore in+or" ation.

3.59 LayoutUnit

LayoutUnit represents a representation in the border layout "order of the layout component".

Info

9a,	layoutUnit
*o "&onent *(a	org.primefaces.component.layout.LayoutUnit
*o "&onent 9y&e	org.primefaces.component.LayoutUnit
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.LayoutUnitRenderer
Renderer *(a	org.primefaces.component.layout.LayoutUnitRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An object representing the association to a server-side UI component instance in a backing bean.
action	null	String	Action of the unit.
style	null	String	Style of the unit.
styleClass	false	String	Style class of the unit.
component	false	Boolean	Style of the unit component.
dataProvider	false	Object	Provider of the unit data.
header	null	String	Text of the header.
footer	null	String	Text of the footer.
initialIndex	null	Integer	Initial index of the item.
lastIndex	null	Integer	Last index of the item.
utter	-&0	String	Gutter style of the layout unit.
visible	true	Boolean	Specifies visibility of the visible property.
dataProvider	false	Object	Provider of the data.
dataProvider	false	Object	Provider of the data.

Name	Default	Type	Description
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.
e+ect	nu((Strin ,	E+ect na " e o+ the (ayout tran ition.
e+ectS&eed	nu((Strin ,	E+ect &eed o+ the (ayout tran ition.

Getting started with LayoutUnit

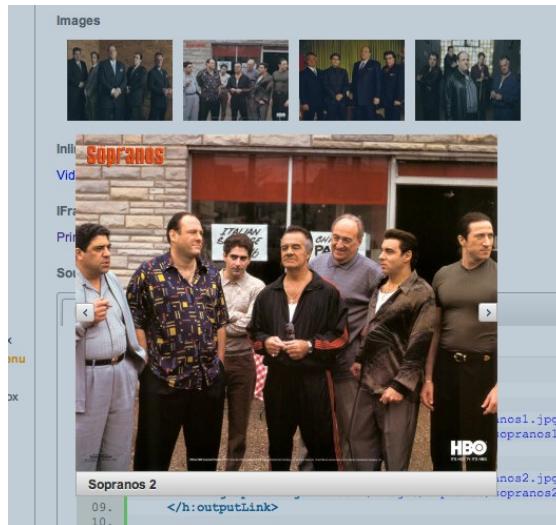
See (ayout co " &onent docu " entation +or " ore in+or " ation re ,ardin , the u a,e o+ (ayoutUnit .

Facets

In addition to *header* and *footer* attribute to di &(ay te0t at the e (ocation E+acet are a(o &rovided ' ith the a " e na " e to di &(ay cu to " content.

3.60 LightBox

6i, htbo0 i a &o ' er+u(over(ay that can di &(ay i " a,e E " u(ti " edia content cu to " content and e0terna(ur(.



Info

9a,	lightbox
*o " &onent *(a	org.primefaces.component.lightbox.LightBox
*o " &onent 9y&e	org.primefaces.component.LightBox
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.LightBoxRenderer
Renderer *(a	org.primefaces.component.lightbox.LightBoxRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onent ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
ty(e	nu((Strin ,	Sty(e o+ the container e(e " ent not the over(ay e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the container e(e " ent not the over(ay e(e " ent.
' idth	nu((Strin ,	Width o+ the over(ay in i+ra " e " ode.

Name	Default	Type	Description
hei, ht	nu((Strin,	. ei , ht o+ the over(ay in i+ra " e " ode.
i+ra " e	+a(e	1oo(ean	S&eci+ie an i+ra " e to di &(ay an e0terna(ur(in over(ay.
i+ra " e9it(e	nu((Strin,	9it(e o+ the i+ra " e e(e " ent.
vi ib(e	+a(e	1oo(ean	Di &(ay (i,htbo0 ' ithout reCuirin , any u er interaction by de+au(t.
on . ide	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen (i,htbo0 i di &(ayed.
onSho '	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen (i,htbo0 i hidden.

Images

9he i "a,e di &(ayed in the (i,ht1o0 need to be ne ted a chi(d out&ut6in3 co " &onent . #o((o ' in , (i,ht1o0 i di &(ayed ' hen any o+ the (in3 are c(ic3ed.

```
<p:lightbox>
    <h:outputLink value="sopranos/sopranos1.jpg" title="Sopranos 1">
        <h:graphicImage value="sopranos/sopranos1_small.jpg"/>
    </h:outputLink>

    <h:outputLink value="sopranos/sopranos2.jpg" title="Sopranos 2">
        <h:graphicImage value="sopranos/sopranos2_small.jpg"/>
    </h:outputLink>

    <h:outputLink value="sopranos/sopranos3.jpg" title="Sopranos 3">
        <h:graphicImage value="sopranos/sopranos3_small.jpg"/>
    </h:outputLink>

    //more
</p:lightbox>
```

IFrame Mode

6i,ht1o0 a(o ha the abi(ity to di &(ay i+ra " e in ide the &a,e over(ayE +o((o ' in , (i,htbo0 di &(ay the !ri " e#ace ho " e&a,e ' hen the (in3 in ide i c(ic3ed.

```
<p:lightbox iframe="true">
    <h:outputLink value="http://www.primefaces.org" title="PrimeFaces HomePage">
        <h:outputText value="PrimeFaces HomePage"/>
    </h:outputLink>
</p:lightbox>
```

*(ic3in , the out&ut6in3 ' i((di &(ay !ri " e#ace ho " e&a,e ' ithin an i+ra " e.

Inline Mode

In(ine " ode act (i3e a " oda(dia(o,E you can di &(ay other AS# content on the &a,e u in , the (i,htbo0 over(ay. Si " &(y &(ace your over(ay content in the Iin(ineI +acet. *(ic3in , the (in3 in the e0a " &(e be(o ' ' i((di &(ay the &ane(Grid content in over(ay.

```
<p:lightBox>
    <h:outputLink value="#" title="Leo Messi" >
        <h:outputText value="The Messiah"/>
    </h:outputLink>
    <f:facet name="inline">
        //content here
    </f:facet>
</p:lightBox>
```

6i, htbo0 in(ine "ode doe n't u&&ort advanced content (i3e co " &(e0 ' id, et . U e a dia(o, in tead +or advanced ca e invo(vin, cu to " content.

Client Side API

Wid, et PrimeFaces.widget.LightBox

Method	Params	Return Type	Description
ho ' ?@	B	void	Di &(ay (i, htbo0.
hide?@	B	void	. ide (i, htbo0.
ho ' UR6?o&t@	o&t	void	Di &(ay a UR6 in a i+ra " e. o&t &ara " eter ha three variab(e . ' idth and hei, ht +or i+ra " e di " en ion and rc +or the &a, e ur(.

Skinning

6i, htbo0 re ide in a "ain container e(e " ent ' hich *style* and *styleClass* o&tion a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB(i, htbo0	8 ain container e(e " ent.
.uiB(i, htbo0BcontentB ' ra&&er	*ontent ' ra&&er e(e " ent.
.uiB(i, htbo0Bcontent	*ontent container.
.uiB(i, htbo0BnavBri, ht	<e0t i " a, e navi, ator.
.uiB(i, htbo0BnavB(e+t	!reviou i " a, e navi, ator.
.uiB(i, htbo0B(oadin,	6oadin, i " a, e.
.uiB(i, htbo0Bca&tion	*a&tion e(e " ent.

3.61 Link

Links are often used to standard hrefs or components.

Info

Value	link
* @ component *(a)	org.primefaces.component.link.Link
* @ component 9y&e	org.primefaces.component.Link
* @ component #a " i(y)	org.primefaces.component
Renderer 9y&e	org.primefaces.component.LinkRenderer
Renderer *(a)	org.primefaces.component.link.LinkRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
value	null	Object	>a value of the component than can be either an Expression or a (iterable) object.
outcome	null	String	Used to resolve navigation case.
includeUrl	false	Boolean	Whether to include &url parameter in target URI.
target	null	String	Identifier of the target element which should be created to.
disabled	false	Boolean	Disabled button.
disabled*(parentWindow)	false	Boolean	Disabled attribute, the *parentWindow on the renderer, or this element's parent.
accessKey	null	String	Access key that is rendered after the button.
charset	null	String	Character encoding of the resource defined by this hyperlink.
coord	null	String	Location and handle of the hot spot on the screen for client use in "x,y" format.
dir	null	String	Direction indication for text that does not inherit directionality. >aid value are 69R and R96.
hreflang	null	String	Language code of the resource defined by the link.

Name	Default	Type	Description
re(nu((String,	Re(ation hi&+ro " the current docu " ent to the anchor &eci+ied by the (in3E va(ue are &rovided by a &aceB e&arated (i t o+ (in3 ty&e .
rev	nu((String,	A rever e (in3 +ro " the anchor &eci+ied by thi (in3 to the current docu " entE va(ue are &rovided by a &aceB e&arated (i t o+ (in3 ty&e .
ha&e	nu((String,	Sha&e o+ hot &ot on the creenE va(id va(ue are de+au(tE rectE circ(e and &o(y.
tabinde0	nu((String,	!o ition o+ the e(e " ent in the tabbin , order.
tar, et	nu((String,	<a " e o+ a +ra " e ' here the re ource tar,eted by thi (in3 ' i((be di &(ayed.
tit(e	nu((String,	Advi ory too(ti& in+or " aton.
ty&e	nu((String,	9y&e o+ re ource re+erenced by the (in3.
ty(e	nu((String,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((String,	Sty(e c(a o+ the co " &onent.
onb(ur	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen button (o e +ocu .
onc(ic3	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen button i c(ic3ed.
ondb(c(ic3	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen button i doub(e c(ic3ed.
on+ocu	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen button receive +ocu .
on3eydo ' n	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over button.
on3ey&re	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over button.
on3eyu&	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over button.
on " ou edo ' n	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over button.
on " ou e " ove	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin button
on " ou eout	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay+ro " button.
on " ou eover	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto button.
on " ou eu&	nu((String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over button.

Name	Default	Type	Description
href	null	String	Defines the URL to which the button links.
exact	true	Boolean	Defines if the button is "exact" (i.e. only links to the current page).

Getting Started with Link

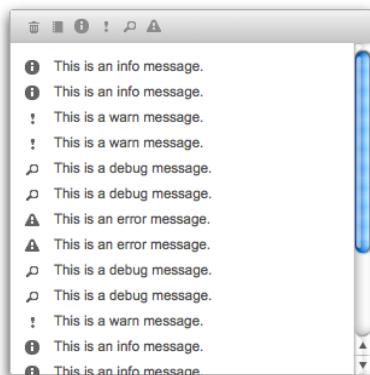
<?xml version="1.0" encoding="UTF-8"?>
 <p:link outcome="target" value="Navigate"/>

So navigate without outcome attribute based approach uses href attribute.

```
<p:link href="http://www.primefaces.org" value="Navigate"/>
```

3.62 Log

60, co " &onent i a vi ua(con o(e to di &(ay (o, on AS# &a,e .



Info

9a,	log
*o " &onent *(a	org.primefaces.component.log.Log
*o " &onent 9y&e	org.primefaces.component.Log
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.LogRenderer
Renderer *(a	org.primefaces.component.log.LogRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean

Getting started with Log

60, co " &onent i u ed i " &(y a addin, the co " &onent to the &a,e.

```
<p:log />
```

Log API

!ri "e#ace u e c(ient ide (o, a&i interna(yE +or e0a" &(e you can u e (o, co " &onent to ee detail(o+ an a/a0 reCue t. 60, A !I i a(o avai(ab(e via ,(oba(!ri "e#ace ob/ect in ca e you'd (i3e to u e the (o, co " &onent to di &(ay your (o, .

```
<script type="text/javascript">
    PrimeFaces.info('Info message');
    PrimeFaces.debug('Debug message');
    PrimeFaces.warn('Warning message');
    PrimeFaces.error('Error message');
</script>
```

Client Side API

Wid, et\ PrimeFaces.widget.Log

Method	Params	Return Type	Description
show?	B	void	Show the container element.
hide?	B	void	. hide the container element.

3.63 Media

8 edia co " &onent i u ed +or e " beddin , " u(ti " edia content.

Info

9a,	media
*o " &onent *(a	org.primefaces.component.media.Media
*o " &onent 9y&e	org.primefaces.component.Media
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MediaRenderer
Renderer *(a	org.primefaces.component.media.MediaRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	True (ean value to specify the rendering of the component) when set to false the component will not be rendered.
bindin,	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
value	null	String	Media source to display.
&ayer	null	String	Style of the player (video player are ICuic3ti " eIEI " indo ' IEI+(a hIEIrea(I and I&d+I).
' idth	null	String	Width of the player.
hei , ht	null	String	. height of the player.
ty(e	null	String	Style of the player.
ty(e*(a	null	String	Style*(a of the player.
cache	true	Boolean	Control browser caching, " false of the resource.

Getting started with Media

In it i " &(e t +or " " edia co " &onent require a source to &(ay

```
<p:media value="/media/ria_with_primefaces.mov" />
```

Player Types

There are several types of players supported by PrimeFaces:

```
<p:media value="http://www.youtube.com/v/ABCDEFGH" player="flash"/>
```

#0((o ' in , i the u&&orted &(ayer and +i(e ty&e .

Player	Types
' indo '	a O&a +E aviE ' " aE ' " v
Cuic3ti " e	ai+E ai++E aacE auE b "&E , " E " ovE " idE " idE " &,E " &-E " -aE & dE CtE Cti+E Ci+E CtiE ndE ti+E ti+E ' avE),%E)&,
+a h	+(vE " &)E ' +
rea(raE ra " E r " E r& " E rvE " iE " i(
&d+	&d+

Parameters

Different proprietary players might have different configuration parameters. The ones can be specified in, +N&ara " ta, .

```
<p:media value="/media/ria_with_primefaces.mov">
    <f:param name="param1" value="value1" />
</p:media>
```

StreamedContent Support

A media component can also display binary media content like audio or video files. In order to do this, it binds a StreamedContent object.

```
<p:media value="#{mediaBean.media}" width="250" height="225" player="quicktime"/>
```

```
public class MediaBean {
    private StreamedContent media;

    public MediaController() {
        InputStream stream = //Create binary stream from database
        media = new DefaultStreamedContent(stream, "video/quicktime");
    }
    public StreamedContent getMedia() { return media; }
}
```

3.64 MegaMenu

A mega menu is a horizontal navigation component that displays submenus together.



Info

9a,	megaMenu
*o "&onent *(a	org.primefaces.component.megamenu.MegaMenu
*o "&onent 9y&e	org.primefaces.component.MegaMenu
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MegaMenuRenderer
Renderer *(a	org.primefaces.component.megamenu.MegaMenuRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	nu((Object	An expression that maps to a server-side UI component instance in a backing bean.
' id, et>ar	nu((String,	<a> element of the client side ' id, et
"ode(nu((String	String attribute in advance to create " enu &ro, ra " statica(y
ty(e	nu((String,	Inline style of the component.
ty(e*(a	nu((String,	Style class of the component.

Name	Default	Type	Description
autoDi &(ay	true	Boolean	Define whether to "enu" i((be di &(ayed on "ou eover or not. When set to +a(eE c(ic3 event is required to di &(ay.
activeIndex0	null	Integer	Index of the active root "enu to di &(ay a hi, h(i, hted. 1y de+au(t no root i hi, h(i, hted.

Getting Started with MegaMenu

Layout of a menu item, rid ba ed and root ite " reCuire co(u " n a chi(dren to de+ine each ection in a ,rid.

```
<p:menu>
    <p:submenu label="TV" icon="ui-icon-check">
        <p:column>
            <p:submenu label="TV.1">
                <p:menuitem value="TV.1.1" url="#" />
                <p:menuitem value="TV.1.2" url="#" />
            </p:submenu>
            <p:submenu label="TV.2">
                <p:menuitem value="TV.2.1" url="#" />
                <p:menuitem value="TV.2.2" url="#" />
                <p:menuitem value="TV.2.3" url="#" />
            </p:submenu>
            <p:submenu label="TV.3">
                <p:menuitem value="TV.3.1" url="#" />
                <p:menuitem value="TV.3.2" url="#" />
            </p:submenu>
        </p:column>

        <p:column>
            <p:submenu label="TV.4">
                <p:menuitem value="TV.4.1" url="#" />
                <p:menuitem value="TV.4.2" url="#" />
            </p:submenu>
            <p:submenu label="TV.5">
                <p:menuitem value="TV.5.1" url="#" />
                <p:menuitem value="TV.5.2" url="#" />
                <p:menuitem value="TV.5.3" url="#" />
            </p:submenu>
            <p:submenu label="TV.6">
                <p:menuitem value="TV.6.1" url="#" />
                <p:menuitem value="TV.6.2" url="#" />
                <p:menuitem value="TV.6.3" url="#" />
            </p:submenu>
        </p:column>
    </p:submenu>
    //more root items
</p:menu>
```

Custom Content

Any content can be placed inside a column.

```
<p:column>
    <strong>Sopranos</strong>
    <p:graphicImage value="/images/sopranos/sopranos1.jpg" width="200"/>
</p:column>
```

Root MenuItem

A root menu item is a root item in a menu option. It can be used to execute an action or to open the menu.

```
<p:menu>
    //submenus
    <p:menuItem label="Logout" action="#{bean.logout}" />
</p:menu>
```

Dynamic Menus

Dynamic menus can be created to represent dynamic content and contain actions or links.

Skinning

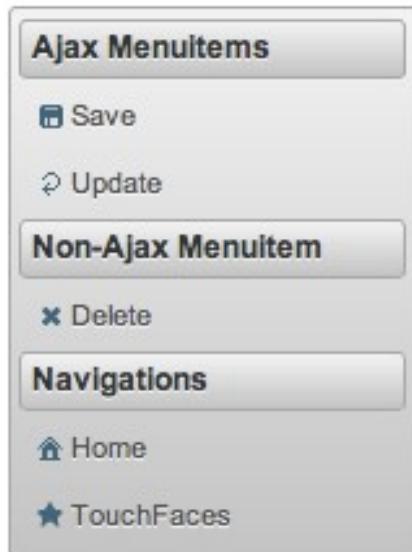
Style classes can be applied to a menu container to change its appearance.

Style Class	Applies
.ui-menu	Menu container element.
.ui-menu-item	Each menu item.
.ui-menu-item-link	Anchor element in a menu item.
.ui-menu-item-text	Text element in a menu item.

A skinning style class can be applied to a menu item.

3.65 Menu

8 enu i a navi ,ation co " &onent ' ith ub " enu and " enuite " .



Info

9a,	menu
*o " &onent *(a	org.primefaces.component.menu.Menu
*o " &onent 9y&e	org.primefaces.component.Menu
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MenuRenderer
Renderer *(a	org.primefaces.component.menu.MenuRenderer

Attributes

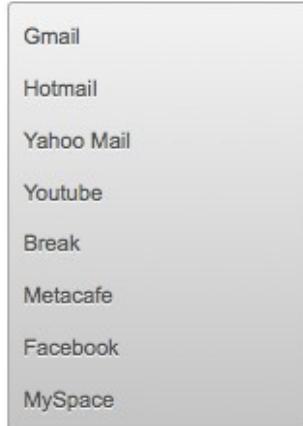
Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that "a& to a erver ide UI*o " &onent in tance in a bac3in , bean.
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et.
" ode(nu((8 enu 8 ode(A " enu " ode(in tance to create " enu &ro ,ra " " atica((y.
tri , ,er	nu((Strin ,	9ar ,et co " &onent to attach the over(ay " enu.

Name	Default	Type	Description
"y	nu((Strin ,	*orner o+ " enu to a(i,n ' ith tri,,er e(e " ent.
at	nu((Strin ,	*orner o+ tri,,er to a(i,n ' ith " enu e(e " ent.
over(ay	+a(e	1oo(ean	De+ine &o itionin, ty&e o+ " enuE either static or over(ay.
ty(e	nu((Strin ,	In(ine ty(e o+ the " ain container e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the " ain container e(e " ent.
tri,,erEvent	c(ic3	Strin ,	Event to ho ' the dyna " ic &o itioned " enu.

Getting started with the Menu

A " enu i co "&o ed o+ ub " enu and " enuite " .

```
<p:menu>
    <p:menuitem value="Gmail" url="http://www.google.com" />
    <p:menuitem value="Hotmail" url="http://www.hotmail.com" />
    <p:menuitem value="Yahoo Mail" url="http://mail.yahoo.com" />
    <p:menuitem value="Youtube" url="http://www.youtube.com" />
    <p:menuitem value="Break" url="http://www.break.com" />
    <p:menuitem value="Metacafe" url="http://www.metacafe.com" />
    <p:menuitem value="Facebook" url="http://www.facebook.com" />
    <p:menuitem value="MySpace" url="http://www.myspace.com" />
</p:menu>
```



Sub " enu are u ed to ,rou& " enuite " K

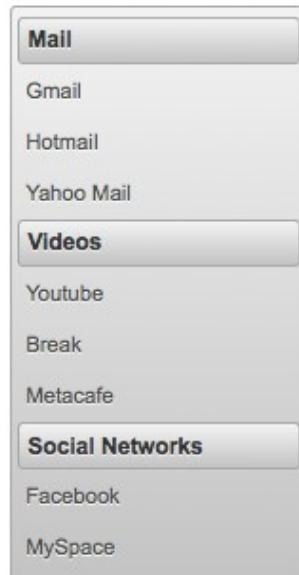
```

<p:menu>
    <p:submenu label="Mail">
        <p:menuitem value="Gmail" url="http://www.google.com" />
        <p:menuitem value="Hotmail" url="http://www.hotmail.com" />
        <p:menuitem value="Yahoo Mail" url="http://mail.yahoo.com" />
    </p:submenu>

    <p:submenu label="Videos">
        <p:menuitem value="Youtube" url="http://www.youtube.com" />
        <p:menuitem value="Break" url="http://www.break.com" />
        <p:menuitem value="Metacafe" url="http://www.metacafe.com" />
    </p:submenu>

    <p:submenu label="Social Networks">
        <p:menuitem value="Facebook" url="http://www.facebook.com" />
        <p:menuitem value="MySpace" url="http://www.myspace.com" />
    </p:submenu>
</p:menu>

```



Overlay Menu

A menu can be positioned on a page in three ways: static and dynamic. By default it is static, meaning the menu is located in the header or footer of the page. In contrast, dynamic menus are not located on the header or footer, but are overlaid on other elements.

A dynamic menu is created by setting the `overlay` option to true and defining a target element. The location of the menu is relative to the target element and is defined by "y" and "x" options that take combinations of our values:

- G (e+)
 - G ri, ht
 - G botto "
 - G to&

#or e0a " &(eE c(ic3in, the button be(o ' ' i((di &(ay the " enu ' ho e to& (e+t corner i a(i, ned ' ith botto " (e+t corner o+ button.

```
<p:menu overlay="true" trigger="btn" my="left top" at="bottom left">
    ...submenus and menuitems
</p:menu>

<p:commandButton id="btn" value="Show Menu" type="button"/>
```

Ajax and Non-Ajax Actions

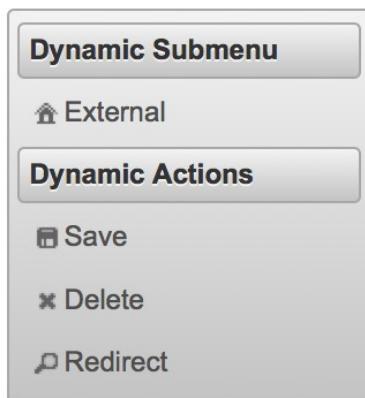
A "enu u e " enuite " E it i ea y to invo3e action ' ith or ' ithout a/a0 a ' e((a navi ,ation. See " enuite " docu " entation +or " ore in+or " ation about the ca&abi(itie .

```
<p:menu>
    <p:submenu label="Options">
        <p:menuItem value="Save" actionListener="#{bean.save}" update="comp"/>
        <p:menuItem value="Update" actionListener="#{bean.update}" ajax="false"/>
        <p:menuItem value="Navigate" url="http://www.primefaces.org"/>
    </p:submenu>
</p:menu>
```

Dynamic Menus

8 enu can be created &ro,ra " " atica((y a ' e((thi i " ore +(e0ib(e co " &red to the dec(arative a&&roach. 8 enu " etadata i de+ined u in, an *org.primefaces.model.MenuModel* in tanceE !ri " e#ace &rovide the bui(tbin *org.primefaces.model.DefaultMenuModel* i " &(e " entation.

#or +urther cu to " i=ation you can a(o create and bind your o ' n 8 enu 8 ode(i " &(e " entation. ?e.,. 7ne ' ith A!A b Entity annotation to ab(e ab(e to &er i t to a databa e@.



```
<p:menu model="#{menuBean.model}" />
```

```

public class MenuBean {

    private MenuModel model;

    public MenuBean() {
        model = new DefaultMenuModel();

        //First submenu
        DefaultSubMenu first_submenu = new DefaultSubMenu("Dynamic Submenu");

        DefaultMenuItem item = new DefaultMenuItem("External");
        item.setUrl("http://www.primefaces.org");
        item.setIcon("ui-icon-home");
        first_submenu.addElement(item);

        model.addElement(first_submenu);

        //Second submenu
        DefaultSubMenu second_submenu = new DefaultSubMenu("Dynamic Actions");

        item = new DefaultMenuItem("Save");
        item.setIcon("ui-icon-disk");
        item.setCommand("#{menuBean.save}");
        item.setUpdate("messages");
        second_submenu.addElement(item);

        item = new DefaultMenuItem("Delete");
        item.setIcon("ui-icon-close");
        item.setCommand("#{menuBean.delete}");
        item.setAjax(false);
        second_submenu.addElement(item);

        item = new DefaultMenuItem("Redirect");
        item.setIcon("ui-icon-search");
        item.setCommand("#{menuBean.redirect}");
        second_submenu.addElement(item);

        model.addElement(second_submenu);
    }

    public MenuModel getModel() { return model; }
}

```

#or a((UI co " &onent counter&art uch a &N " enuite " E &N ub " enue &N e&arator a corre &ondin ,
inter+ace ' ith a de+au(t i " &(e " entation e0i t in 8 enu 8 ode(A !I. Re ,ardin , action E i+ you need to
&a &ara " eter in a/a0 or nonBa/a0 co " " and E u e et!ara " ?3eyE va(ue@ " ethod . '

8 enu 8 ode(A !I i u&&orted by a((" enu co " &onent that have " ode(attribute.

Skinning

8 enu re ide in a " ain container e(e " ent ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB " enu	*ontainer e(e " ent o+ " enu
.uiB " enuB(i t	6i t container
.uiB " enuite "	Each " enu ite "
.uiB " enuite " B(in3	Anchor e(e " ent in a (in3 ite "
.uiB " enuite " Bte0t	9e0t e(e " ent in an ite "
.uiB " enuB (idin,	*ontainer o+ i&od (i3e (idin, " enu

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.66 Menubar

A menubar is a horizontal navigation component.



Info

9a,	menubar
*o "&onent *(a	org.primefaces.component.menubar.MenuBar
*o "&onent 9y&e	org.primefaces.component.MenuBar
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MenuBarRenderer
Renderer *(a	org.primefaces.component.menubar.MenuBarRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to true, the component will be rendered.
binding	nu((Object	An expression that maps to a server-side UI component instance in a backing bean.
' id , et>ar	nu((String ,	ar
" ode(nu((String	String value in quotes to create " encode and replace it with its corresponding character.
ty(e	nu((String ,	Inline style of the menubar
ty(e*(a	nu((String ,	Style class of the menubar
autoDi &(ay	+a(e	Boolean	Defines whether the timer (every 100ms) is triggered or not. When set to true, it is required to define the event.

Getting started with Menubar

Submenu and submenu items are required to create the menubar.

```
<p:menubar>
    <p:submenu label="Mail">
        <p:menuitem value="Gmail" url="http://www.google.com" />
        <p:menuitem value="Hotmail" url="http://www.hotmail.com" />
        <p:menuitem value="Yahoo Mail" url="http://mail.yahoo.com" />
    </p:submenu>
    <p:submenu label="Videos">
        <p:menuitem value="Youtube" url="http://www.youtube.com" />
        <p:menuitem value="Break" url="http://www.break.com" />
    </p:submenu>
</p:menubar>
```

Nested Menus

To create a menubar with a hierarchical structure, embed one menu in a parent menu.

```
<p:menubar>
    <p:submenu label="File">
        <p:submenu label="New">
            <p:menuitem value="Project" url="#" />
            <p:menuitem value="Other" url="#" />
        </p:submenu>
        <p:menuitem value="Open" url="#" /></p:menuitem>
        <p:menuitem value="Quit" url="#" /></p:menuitem>
    </p:submenu>
    <p:submenu label="Edit">
        <p:menuitem value="Undo" url="#" /></p:menuitem>
        <p:menuitem value="Redo" url="#" /></p:menuitem>
    </p:submenu>
    <p:submenu label="Help">
        <p:menuitem label="Contents" url="#" />
        <p:submenu label="Search">
            <p:submenu label="Text">
                <p:menuitem value="Workspace" url="#" />
            </p:submenu>
            <p:menuitem value="File" url="#" />
        </p:submenu>
    </p:submenu>
</p:menubar>
```

Root MenuItem

A menubar without a root menu option is empty.

```
<p:menubar>
    <p:menuitem label="Logout" action="#{bean.logout}" />
</p:menubar>
```

Ajax and Non-Ajax Actions

A "enu u e " enuite" È it i ea y to invo3e action ' ith or ' ithout a/a0 a ' e((a navi ,ation. See "enuite" docu " entation +or " ore in+or " ation about the ca&abi(itie .

```
<p:menubar>
    <p:submenu label="Options">
        <p:menuitem value="Save" actionListener="#{bean.save}" update="comp"/>
        <p:menuitem value="Update" actionListener="#{bean.update}" ajax="false"/>
        <p:menuitem value="Navigate" url="http://www.primefaces.org"/>
    </p:submenu>
</p:menubar>
```

Dynamic Menus

8 enu can be created &ro, ra " " atica((y a ' e((E ee the dyna " ic " enu &art in " enu co " &onent ection +or " ore in+or " ation and an e0a " &(e.

Skinning

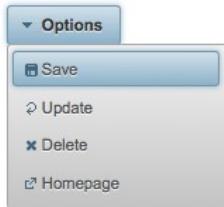
8 enubar re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in , i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB " enubar	*ontainer e(e " ent o+ " enubar.
.uiB " enuB(i t	6i t container
.uiB " enuite "	Each " enu ite "
.uiB " enuite " B(in3	Anchor e(e " ent in a (in3 ite "
.uiB " enuite " Bte0t	9e0t e(e " ent in an ite "

A 3innin , ty(e c(a e are ,(oba(E ee the " ain the " in , ection +or " ore in+or " ation.

3.67 MenuButton

8 enu 1 utton di &(ay di+erent co " " and in a &o&u& " enu.



Info

9a,	menuButton
*o " &onent *(a	org.primefaces.component menubutton.MenuButton
*o " &onent 9y&e	org.primefaces.component.MenuButton
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MenuButtonRenderer
Renderer *(a	org.primefaces.component menubutton.MenuButtonRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+iy the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in , bean.
va(ue	nu((Strin ,	6abe(o+ the button
ty(e	nu((Strin ,	Sty(e o+ the " ain container e(e " ent
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the " ain container e(e " ent
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et
" ode(nu((8 enu 8 ode(8 enu 8 ode(in tance to create " enu &ro ,ra " " atica((y
di ab(ed	+a(e	1oo(ean	Di ab(e or enab(e the button.
icon !o	(e+t	Strin ,	!o ition o+ the iconE va(id va(ue are (e+t and ri ,ht.
a&&end9o	nu((Strin ,	A&&end the over(ay to the e(e " ent de+ined by earch e0&re ion. De+au(t to docu " ent body.

Getting started with the MenuButton

8 enu1utton con i t o+ one ore "ore "enuite" . #o((o 'in, "enubutton e0a" &(e ha three "enuite" E+ir t one i u ed tri,,er an action 'ith a/a0E econd one doe the i" i(ar but ' ithout a/a0 and third one i u ed +or redirect &ur&o e .

```
<p:menuButton value="Options">
    <p:menuItem value="Save" actionListener="#{bean.save}" update="comp" />
    <p:menuItem value="Update" actionListener="#{bean.update}" ajax="false" />
    <p:menuItem value="Go Home" url="/home.jsf" />
</p:menuButton>
```

Dynamic Menus

8 enu can be created &ro,ra " "atrica((y a ' e((E ee the dyna" ic "enu &art in "enu co" &onent ection +or "ore in+or" ation and an e0a" &(e.

Skinning

8 enu1utton re ide in a "ain container ' hich *style* and *styleClass* attribute a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the "ain the "in, ection +or "ore in+or" ation. #o((o 'in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB "enu	*ontainer e(e " ent o+ "enu.
.uiB "enuB(i t	6i t container
.uiB "enuite "	Each "enu ite"
.uiB "enuite " B(in3	Anchor e(e " ent in a (in3 ite "
.uiB "enuite " Bte0t	9e0t e(e " ent in an ite "
.uiBbutton	1utton e(e " ent
.uiBbuttonBte0t	6abe(o+ button

3.68 MenuItem

8 enulte " i u ed by variou " enu co " &onent .

Info

9a,	menuItem
9a, *(a	org.primefaces.component.menuitem.MenuItemTag
*o "&onent *(a	org.primefaces.component.menuitem.MenuItem
*o "&onent 9y&e	org.primefaces.component.MenuItem
*o "&onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7 b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean.
va(ue	nu((Strin ,	6abe(o+ the " enuite "
action6i tener	nu((8 ethod E0&r	Action (i tener to be invo3ed ' hen " enuite " i c(ic3ed.
action	nu((8 ethod E0&r	Action to be invo3ed ' hen " enuite " i c(ic3ed.
i " " ediate	+a(e	1oo(ean	When trueE action o+ thi " enuite " i &roce ed a+ter a&&(y reCue t &ha e.
ur(nu((Strin ,	Ur(to be navi ,ated ' hen " enuite " i c(ic3ed
tar , et	nu((Strin ,	9ar ,et ty&e o+ ur(navi ,ation
ty(e	nu((Strin ,	Sty(e o+ the " enuite " (abe(
ty(e*(a	nu((Strin ,	Sty(e*(a o+ the " enuite " (abe(
onc(ic3	nu((Strin ,	Aava cri&t event hand(er +or c(ic3 event
a ync	+a(e	1oo(ean	When et to trueE a/a0 reCue t are not Cueued.
&roce	nu((Strin ,	*o "&onent to &roce &artia((y in tead o+ ' ho(e vie ' .
u&date	nu((Strin ,	*o "&onent to u&date a+ter a/a0 reCue t .

Name	Default	Type	Description
di ab(ed	+a(e	1oo(ean	Di ab(e the " enuite ".
on tart	nu((Strin,	Aava cri&t hand(er to e0ecute be+ore a/a0 reCue t i be, in .
onco " &(ete	nu((Strin,	Aava cri&t hand(er to e0ecute ' hen a/a0 reCue t i co " &(eted.
on ucce	nu((Strin,	Aava cri&t hand(er to e0ecute ' hen a/a0 reCue t ucceed .
onerror	nu((Strin,	Aava cri&t hand(er to e0ecute ' hen a/a0 reCue t +ai(.
,(oba(true	1oo(ean	G(oba(a/a0 reCue t are (i tened by a/a0Statu co " &onentE ettin, ,(oba(to +a(e ' i((not tri, ,er a/a0Statu .
de(ay	nu((Strin,	I+ (e than <i>delay</i> " i(((i econd e(a& e bet ' een ca((to <i>request()</i> on(y the " o t recent one i ent and a((other reCue t are di carded. I+ thi o&tion i not &eci+iiedE or i+ the va(ue o+ <i>delay</i> i the (itera(trin, lnone) ' ithout the Cuote E no de(ay i u ed.
&artia(Sub " it	+a(e	1oo(ean	Enab(e eria(i=ation o+ va(ue be(on, in, to the &artia((y &roce ed co " &onent on(y.
re et>a(ue	+a(e	1oo(ean	I+ trueE (oca(va(ue o+ in&ut co " &onent to be u&dated ' ithin the a/a0 reCue t ' ou(d be re et.
i ,noreAutoU&date	+a(e	1oo(ean	I+ trueE co " &onent ' hich autoU&datePItrueI ' i((not be u&dated +or thi reCue t. I+ not &eci+iiedE or the va(ue i +a(eE no uch indication i " ade.
a/a0	true	1oo(ean	S&eci+ie ub " it " ode.
icon	nu((Strin,	!ath o+ the " enuite " i " a,e.
tit(e	nu((Strin,	Advi ory too(ti& in+or " ation.
outco " e	nu((Strin,	<avi,ation ca e outco " e.
inc(ude>ie ' !ara "	+a(e	1oo(ean	De+ine i+ &a,e &ara " eter hou(d be in tar, et URI.
+ra, " ent	nu((Strin,	Ident+ier o+ the tar, et &a,e e(e " ent to cro((to.
di ab(e*(ientWindo '	+a(e	1oo(ean	Di ab(e a&&endin, the *(ientWindo ' on the renderin, o+ thi e(e " ent.
containerSty(e	nu((Strin,	In(ine ty(e o+ the " enuite " container.
containerSty(e*(a	nu((Strin,	Sty(e c(a o+ the " enuite " container.

Getting started with MenuItem

8 enulte " i a ,eneric co " &onent u ed by the +o((o ' in, co " &onent .

- G 8 enu
- G 8 enu1ar
- G 8 e,a 8 enu
- G 1readcru " b
- G Doc3
- G Stac3
- G 8 enu1utton
- G S&(it1utton
- G !ane(8 enu
- G 9ab 8 enu
- G S(ide 8 enu
- G 9iered 8 enu

<ote that o " e attribute o+ " enuite " " i, ht not be u&&orted by the e " enu co " &onent . Re+er to the &eci+ic co " &onent docu " entation +or " ore in+or " ation.

Navigation vs Action

8 enuite " ha t' o u e ca e E direct(y navi,atin, to a ur(' ith GE9 or doin, a !7S9 to e0ecute an action. 9hi i decided by ur(or outco " e attribute E i+ either one i &re ent " enuite " doe a GE9 reCue tE i+ not &arent +or " i &o ted ' ith or ' ithout a/a0 decided by ajax attribute.

Icons

9here are t' o 'ay to &eci+y an icon o+ a " enuite " E you can either u e bund(ed icon ' ithin !ri " e#ace or &rovide your o ' n via c .

9he " eRo((er Icon

```
<p:menuItem icon="ui-icon-disk" ... />
```

*u to " Icon

```
<p:menuItem icon="barca" ... />
```

```
.barca {
    background: url(barca_logo.png) no-repeat;
    height:16px;
    width:16px;
}
```

3.69 Message

A standard JSF component that displays messages.



Info

9a,	message
*o " &onent *(a	org.primefaces.component.message.Message
*o " &onent 9y&e	org.primefaces.component.Message
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MessageRenderer
Renderer *(a	org.primefaces.component.message.MessageRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	Unique identifier of the component.
rendered	true	1oo(ean	Boolean value to specify whether the component is rendered or not.
bindin,	nu((7b/ect	An expression that refers to a server-side UI component instance in a backing bean.
ho ' Su " " ary	+a(e	1oo(ean	Specifies the user action or the #ace of the component.
ho ' Detai(true	1oo(ean	Specifies the detail of the component.
+or	nu((Strin ,	Id of the component whose state is to be displayed.
redi &(ay	true	1oo(ean	Defines if a ready rendered component is displayed.
di &(ay	both	Strin ,	Defines the display mode.
e ca&e	true	1oo(ean	Defines whether the component is displayed or not.
everity	nu((Strin ,	Specifies the severity level of the message.
ty(e	nu((Strin ,	Defines the type of the component.
ty(e*(a	nu((Strin ,	Specifies the category of the component.

Getting started with Message

8 e a,e u a,e i e0act(y a"e a standard "e a,e.

```
<h:inputText id="txt" value="#{bean.text}" />
<p:message for="txt" />
```

Display Mode

8 e a,e co " &onent ha three di+erent di &(ay " ode K

G te0t N 7n(y "e a,e te0t i di &(ayed.

G icon N 7n(y "e a,e everity i di &(ayed and "e a,e te0t i vi ib(e a a too(ti&.

G both ?de+au(t@ N 1oth icon and te0t are di &(ayed.

Severity Levels

U in, everity attribute you can de+ine ' hich everitie can be di &(ayed by the co " &onent. #or in tance you can con+i,ure " e a,e to on(y di &(ay in+o and ' arnin, .

```
<p:message severity="info, warn" for="txt"/>
```

Escaping

*o " &onent e ca&e ht "(content in " e a,e by de+au(t@ in ca e you need to di &(ay ht " (di ab(e e ca&e o&tion.

```
<p:message escape="false" for="txt" />
```

Skinning

#u(((i t o+ *SS e(ector o+ " e a,e i a +o((o' K

Style Class	Applies
uiB " e a,eBV everityW	*ontainer e(e " ent o+ the " e a,e
uiB " e a,eBV everityWB u " " ary	Su " " ary te0t
uiB " e a,eBV everityWBdetai(Detai(te0t

V everityW can be 'in+o' 'error' ' ' arn' and error.

3.70 Messages

8 e a,e i a &reB 3inned e0tended ver ion o+ the standard AS# " e a,e co " &onent.



Info

9a,	messages
*o " &onent *(a	org.primefaces.component.messages.Messages
*o " &onent 9y&e	org.primefaces.component.Messages
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MessagesRenderer
Renderer *(a	org.primefaces.component.messages.MessagesRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier o+ the co " &onent.
rendered	true	Boolean	True(ean va(ue to &eciy the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	null	Object	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean.
ho ' Su " " ary	true	Boolean	S&eci+ie i+ the u " " ary o+ the #ace 8 e a,e hou(d be di &(ayed.
ho ' Detai(+a(e	Boolean	S&eci+ie i+ the detai(o+ the #ace 8 e a,e hou(d be di &(ayed.
,(oba(7n(y	+a(e	String	When trueE on(y +ace " e a,e ' ith no c(ientId are di &(ayed.
redi &(ay	true	Boolean	De+ine i+ a(ready rendered " e a,e hou(d be di &(ayed
autoU&date	+a(e	Boolean	Enab(e auto u&date " ode i+ et true.
+or	null	String	<a " e o+ a ociated 3eyta3e &recidence ' hen u ed ' ith ,(oba(7n(y.

Name	Default	Type	Description
severity	true	Boolean	Defines whether it's possible to hide the message.
severityIcon	null	String	*o " "a separated (into severity to display only).
showIcon	false	Boolean	Add a close icon to hide the message.
styleClass	null	String	Defines style class for the component.
icon	true	Boolean	Defines if severity icon is displayed.

Getting started with Messages

8 messages are available standard message types.

```
<p:messages />
```

AutoUpdate

When auto update is enabled, the component is updated with each arriving message automatically.

Targetable Messages

Here "ay be ti"e "here you need to target one or "ore "e a,e to a &ecentric "e a,e co" &onent or each "e a,e you have ,ro'(and "e a,e on a"e &a,e and you need to di &(ay o"e "e a,e on ,ro'(and o"e on "e a,e . Use target attribute to associate "e a,e with &ecentric co" &onent .

```
<p:messages for="somekey" />
<p:growl for="anotherkey" />
```

```
FacesContext context = FacesContext.getCurrentInstance();

context.addMessage("somekey", facesMessage1);
context.addMessage("somekey", facesMessage2);

context.addMessage("anotherkey", facesMessage3);
```

In a "&(e above, "e a,e " i((di &(ay +ir t and second "e a,e and ,ro'(' i((on(y di &(ay the)rd "e a,e.

Severity Levels

Using the severity attribute you can define which severity can be displayed by the component. #or in advance you can configure "e a,e to on(y di &(ay info and warning, .

```
<p:messages severity="info,warning" />
```

Escaping

By default, the content of a message is escaped (content in "e a,e e ca&e ht" (content in "e a,e E di ab(e e ca&e o&tion to di &(ay content a ht " (.

```
<p:messages escape="false" />
```

Skinning

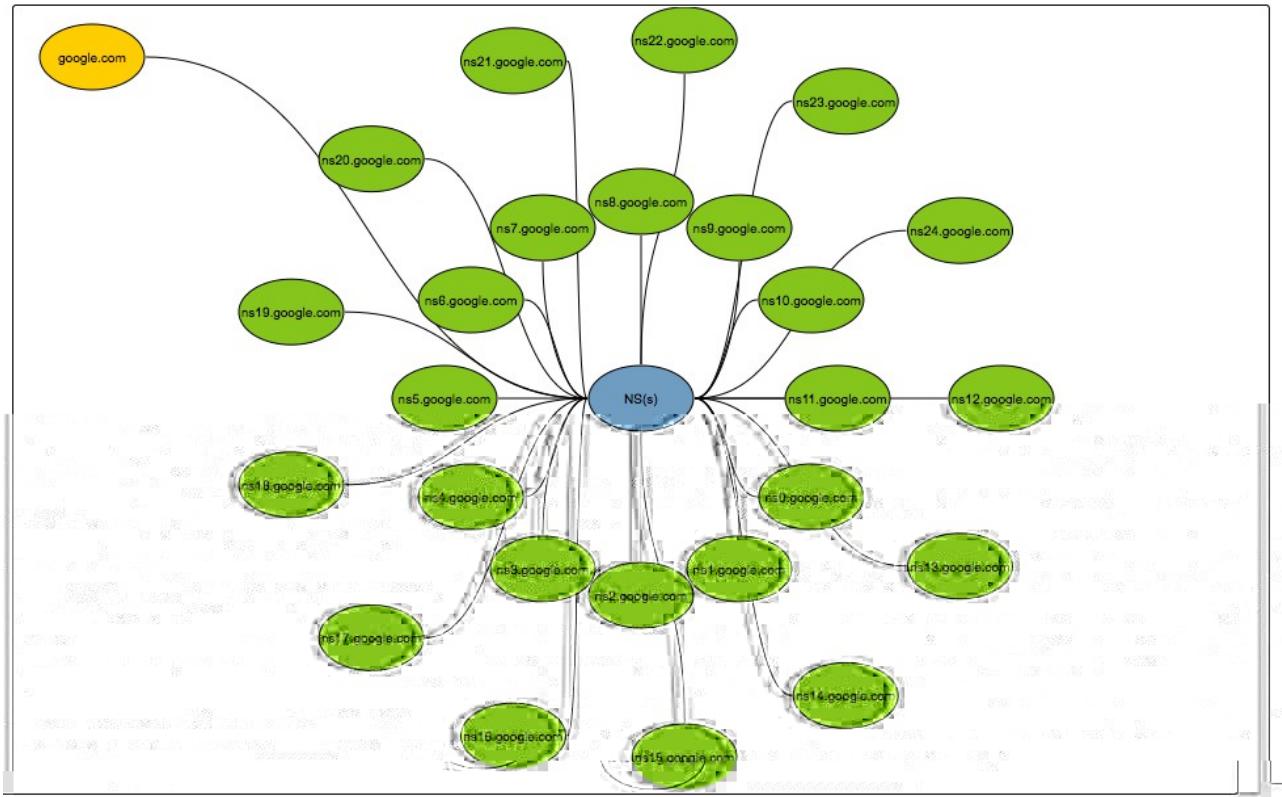
#u(((i t o+ *SS e(ector o+ "e a,e i a +o((o ' K

Style Class	Applies
uiB "e a,e BV everityW	*ontainer e(e " ent o+ the "e a,e
uiB "e a,e BV everityWB u" " ary	Su " " ary te0t
uiB "e a,e BV everityWBdetai(Detai(te0t
uiB "e a,e BV everityWBicon	Icon o+ the "e a,e.

Value of everityW can be 'info' 'error' 'warn' and 'error'.

3.71 Mindmap

8 ind " a& i an interactive too(to vi ua(i=e " ind " a& data +eaturin, (a=y (oadin, E ca((bac3 E ani " ation and " ore.



Info

9a,	mindmap
*o "&onent *(a	org.primefaces.component.mindmap.Mindmap
*o "&onent 9y&e	org.primefaces.component.Mindmap
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MindmapRenderer
Renderer *(a	org.primefaces.component.mindmap.MindmapRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Value indicating whether the component is rendered. If false, it will not be rendered.

Name	Default	Type	Description
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
va(ue	nu((8 ind " a&<ode	8 enu 8 ode(in tance to bui(d " enu dyna " ica((y.
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e *(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.
e+ectS&eed)00	Inte ,er	S&eed o+ the ani " ation in " i((i econd .
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.

Getting started with Mindmap

8 ind " a& reCuire an in tance o+ *org.primefaces.model.mindmap.MindmapNode* a the root. Due to it' (a=y nature a ect a/a0 behavior " u t be &rovided to (oad chi(dren o+ e(ected node on the +y' ith a/a0.

```
public class MindmapBean {

    private MindmapNode root;

    public MindmapBean() {
        root = new DefaultMindmapNode("google.com", "Google", "FFCC00", false);

        MindmapNode ips = new DefaultMindmapNode("IPs", "IP Nos", "6e9ebf", true);
        MindmapNode ns = new DefaultMindmapNode("NS(s)", "Names", "6e9ebf", true);
        MindmapNode mw = new DefaultMindmapNode("Mw", "Malicious ", "6e9ebf", true);

        root.addNode(ips);
        root.addNode(ns);
        root.addNode(malware);
    }

    public MindmapNode getRoot() {
        return root;
    }

    public void onNodeSelect(SelectEvent event) {
        MindmapNode node = (MindmapNode) event.getObject();
        //load children of select node and add via node.addNode(childNode);
    }
}
```

```
<p:mindmap value="#{mindmapBean.root}" style="width:100%;height:600px">
    <p:ajax event="select" listener="#{mindmapBean.onNodeSelect}" />
</p:mindmap>
```

DoubleClick Behavior

Selecting a node with a double click leads to a detailed dialog showing the details of the selected node.

```
<p:mindmap value="#{mindmapBean.root}" style="width:100%;height:600px;">
    <p:ajax event="select" listener="#{mindmapBean.onNodeSelect}" />
    <p:ajax event="dblselect" listener="#{mindmapBean.onNodeDblselect}"
        update="output" oncomplete="PF('details').show()"/>
</p:mindmap>

<p:dialog widgetVar="details" header="Node Details" resizable="false" modal="true"
    showEffect="fade" hideEffect="fade">
    <h:outputText id="output" value="#{mindmapBean.selectedNode.data}" />
</p:dialog>
```

```
public void onNodeDblselect(SelectEvent event) {
    this.selectedNode = (MindmapNode) event.getObject();
}
```

MindmapNode API

org.primefaces.model.mindmap.MindmapNode

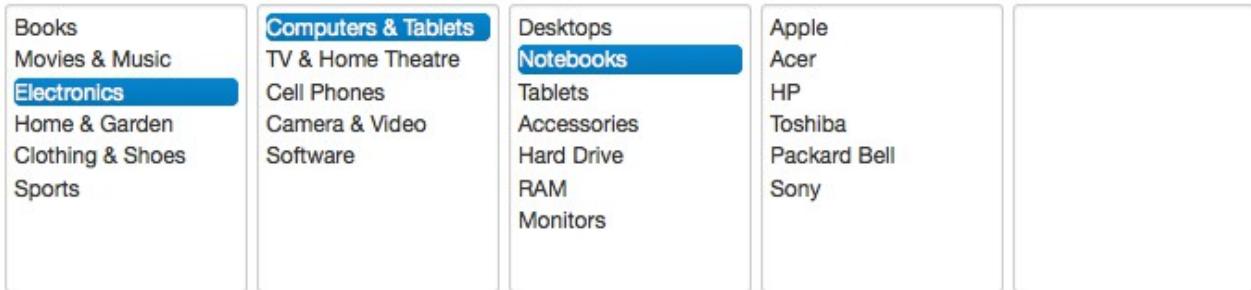
Property	Default	Type	Description
label	null	String	Label of the node.
data	null	Object	Additional data associated with the node.
icon	null	String	Icon code of the node.
electable	true	Boolean	#(a, to define if node is selectable.)
parent	null	String	Parent node instance.

Tips

IE 5 and IE 2 are not supported due to technical limitation. IE 9 is supported.

3.72 MultiSelectListbox

8 u(tiSe(ect6i tbo0 i u ed to e ect an ite " +ro " a co((ection o+ (i tbo0e that are in &arentBchi(d re(ation hi&.



Info

9a,	multiSelectListbox
*o "&onent *(a	org.primefaces.component.multiselectlistbox.MultiSelectListbox
*o "&onent 9y&e	org.primefaces.component.MultiSelectListbox
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.MultiSelectListboxRenderer
Renderer *(a	org.primefaces.component.multiselectlistbox.MultiSelectListboxRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean.
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
ty(e	nu((Strin,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the co " &onent.
di ab(ed	+a(e	1oo(ean	I+ trueE di ab(e the co " &onent.
e+ect	nu((Strin,	E+ect to u e ' hen ho ' in, a ,rou& o+ ite " .

Getting started with MultiSelectListbox

8 u(tiSe ect6i tbo0 need a co((ection o+ Se(ectlte " Grou& .

```

public class MultiSelectListboxBean {

    private List<SelectItem> categories;
    private String selection;

    @PostConstruct
    public void init() {
        categories = new ArrayList<SelectItem>();
        SelectItemGroup group1 = new SelectItemGroup("Group 1");
        SelectItemGroup group2 = new SelectItemGroup("Group 2");
        SelectItemGroup group3 = new SelectItemGroup("Group 3");
        SelectItemGroup group4 = new SelectItemGroup("Group 4");

        SelectItemGroup group11 = new SelectItemGroup("Group 1.1");
        SelectItemGroup group12 = new SelectItemGroup("Group 1.2");

        SelectItemGroup group21 = new SelectItemGroup("Group 2.1");

        SelectItem option31 = new SelectItem("Option 3.1", "Option 3.1");
        SelectItem option32 = new SelectItem("Option 3.2", "Option 3.2");
        SelectItem option33 = new SelectItem("Option 3.3", "Option 3.3");
        SelectItem option34 = new SelectItem("Option 3.4", "Option 3.4");

        SelectItem option41 = new SelectItem("Option 4.1", "Option 4.1");

        SelectItem option111 = new SelectItem("Option 1.1.1");
        SelectItem option112 = new SelectItem("Option 1.1.2");
        group11.setSelectItems(new SelectItem[]{option111, option112});

        SelectItem option121 = new SelectItem("Option 1.2.1", "Option 1.2.1");
        SelectItem option122 = new SelectItem("Option 1.2.2", "Option 1.2.2");
        SelectItem option123 = new SelectItem("Option 1.2.3", "Option 1.2.3");
        group12.setSelectItems(new SelectItem[]{option121, option122, option123});

        SelectItem option211 = new SelectItem("Option 2.1.1", "Option 2.1.1");
        group21.setSelectItems(new SelectItem[]{option211});

        group1.setSelectItems(new SelectItem[]{group11, group12});
        group2.setSelectItems(new SelectItem[]{group21});
        group3.setSelectItems(new SelectItem[]{option31, option32, option33,
                                             option34});
        group4.setSelectItems(new SelectItem[]{option41});

        categories.add(group1);
        categories.add(group2);
        categories.add(group3);
        categories.add(group4);
    }

    //getters-setters of categories and selection
}

```

```
<p:multiSelectListbox value="#{multiSelectListboxBean.selection}">
    <f:selectItems value="#{multiSelectListboxBean.categories}" />
</p:multiSelectListbox>
```

Note that SelectItems " Groups are not selectable on the value of SelectItems " can be added to the bean.

Effects

An additional action is executed during, for example, a mouse over or click event. These actions are available (or *effect*) attributes of individual items and are triggered by the mouse over or click event. The effect can be defined using the `onMouseOver` and `onClick` attributes.

Client Side API

Widget `PrimeFaces.widget.MultiSelectListbox`

Method	Params	Return Type	Description
enable()	Boolean	void	Enables the component.
disable()	Boolean	void	Disables the component.
highlightItem("Group")	(item) element or Query object	void	Show the selected item.

Skinning

Style classes are defined in a skin container which *style* and *styleClass* attributes apply. A skinning, type class are placed on the main container, selection or more specifically. # of items, in the structure of type class

Style Class	Applies
.ui-multiselect-item	Main container element.
.ui-multiselect-item-group	Item container.
.ui-multiselect-item-label	Each item in a list.

3.73 NotificationBar

The notification bar displays a message to the user. It can be used to inform the user about an action taken or a notification.

Info

9a,	notificationBar
*o "&onent *(a	org.primefaces.component.notificationbar.NotificationBar
*o "&onent 9y&e	org.primefaces.component.NotificationBar
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.NotificationBarRenderer
Renderer *(a	org.primefaces.component.notificationbar.NotificationBarRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	Unique identifier of the component
rendered	true	booleean	Boolean value to specify if the component is rendered. If false, it will not be rendered.
binding	nu((Object	An expression that refers to a server-side UI component in a backing bean.
type	nu((String	Style of the container element
type*(a	nu((String	Style of the container element
position	top	String	Position of the bar. Top or Bottom.
effect	fade	String	Effect of the bar. Fade In or Out.
effectSpeed	normal	String	Speed of the effect. Normal or Fast.
autoDismiss	false	booleean	When true, the bar is dismissed on click.
' id, et>ar	nu((String	Client-side id of the bar.

Getting started with NotificationBar

A notification bar is a component where any content can be placed in its body.

```
<p:notificationBar>
    //Content
</p:notificationBar>
```

Showing and Hiding

To show and hide the content of notification bar & provide an easy to use client-side API that can be accessed through the 'id', `<ar.show()` displays the bar and `hide()` hides it. `isVisible()` and `toggle()` are additional client-side API methods.

```
<p:notificationBar widgetVar="nv">
    //Content
</p:notificationBar>

<h:outputLink value="#" onclick="PF('nv').show()">Show</h:outputLink>
<h:outputLink value="#" onclick="PF('nv').hide()">Hide</h:outputLink>
```

Note that notification bar has a default built-in close icon to hide the content.

Effects

Default effect to be used when displaying and hiding the bar is `fade`. Another optional effect is `slide`.

```
<p:notificationBar effect="slide">
    //Content
</p:notificationBar>
```

If you'd like to turn off animation effect name to `Invisible`. In addition duration of the animation is controlled via `effectSpeed` attribute that can take integer values (0 or 1+ a float value).

Position

Default position of bar is `top` other visibility is `hidden`, the bar at the bottom or the `bottom`. Note that bar position is fixed even if it's cropped bar is not cropped.

```
<p:notificationBar position="bottom">
    //Content
</p:notificationBar>
```

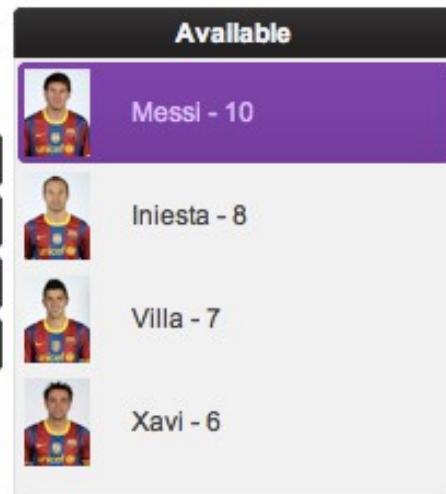
Skinning

Style and style* attributes apply to the main container element. Additionally there are two predefined classes used to control icons and body.

Selector	Applies
.ui-notificationbar	Main container element
.ui-notificationbar-body	Body icon element

3.74 OrderList

The `OrderList` component is used to sort a collection of items, allowing users to reorder them via drag-and-drop or keyboard navigation.

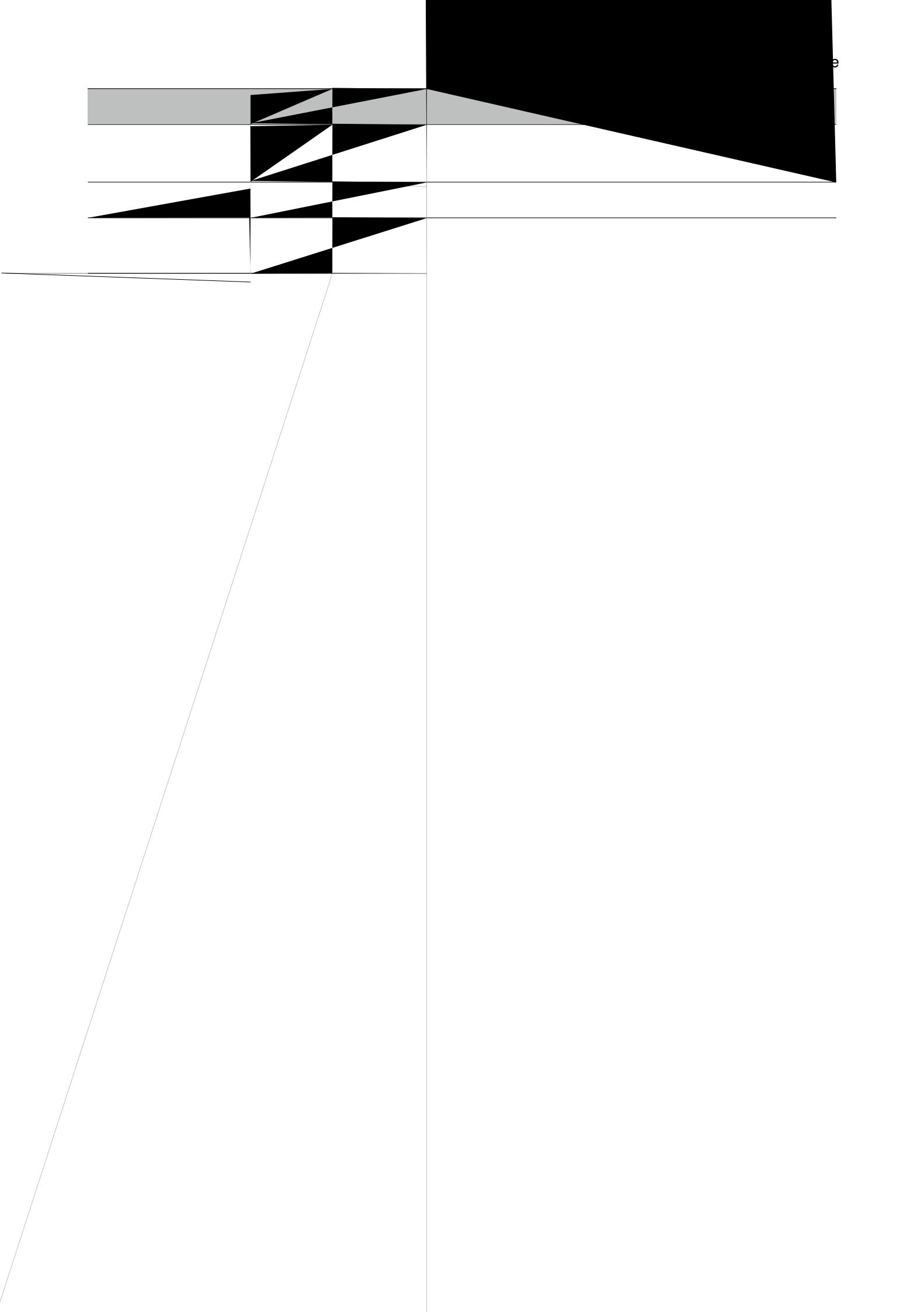


Info

9a,	orderList
*o " &onent *(a	org.primefaces.component.orderlist.OrderList
*o " &onent 9y&e	org.primefaces.component.OrderList
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.OrderListRenderer
Renderer *(a	org.primefaces.component.orderlist.OrderListRenderer

Attributes

Name	Default	Type	Description
id	nu((String	Unique identifier for the component.
rendered	true	Boolean	True (value to display the component, false to hide it) or a converter value to determine if it should be rendered.
binding	nu((Object	An expression that refers to a server-side UI component instance in a backing bean.
value	nu((Object	The value of the component rendered to a string.
converter	nu((Converter	An expression or an iterator that defines a converter for the component. When it's an Expression, it's resolved to a converter instance. In case it's a static text, it's used to refer to a converter id.



A (i t i required to use OrderList component.

```
public class OrderListBean {
    private List<String> cities;

    public OrderListBean() {
        cities = new ArrayList<String>();

        cities.add("Istanbul");
        cities.add("Ankara");
        cities.add("Izmir");
        cities.add("Antalya");
        cities.add("Bursa");
    }

    //getter&setter for cities
}
```

```
<p:orderList value="#{orderListBean.cities}" var="city"
              itemLabel="#{city}" itemValue="#{city}""/>>
```

Advanced OrderList

OrderList supports dynamic content instead of static items. In addition, it also supports a converter defined.

```
public class OrderListBean {
    private List<Player> players;

    public OrderListBean() {
        players = new ArrayList<Player>();

        players.add(new Player("Messi", 10, "messi.jpg"));
        players.add(new Player("Iniesta", 8, "iniesta.jpg"));
        players.add(new Player("Villa", 7, "villa.jpg"));
        players.add(new Player("Xavi", 6, "xavi.jpg"));
    }

    //getter&setter for players
}
```

```
<p:orderList value="#{orderListBean.players}" var="player" itemValue="#{player}"
              converter="player">
    <p:column style="width:25%">
        <p:graphicImage value="/images/barca/#{player.photo}" />
    </p:column>
    <p:column style="width:75%;">
        #{player.name} - #{player.number}
    </p:column>
</p:orderList>
```

Header

A facet called "caution" is provided to display a header content or the order item.

Effects

An animation is executed during reordering, default effect is slide and options are available (or effect attribute) including bounce, drop, slide, flip, turn and rotate.

Skinning

Order items are styled in a main container which has style and styleClass attribute applied. A skinning type class is applied to each item, either the item or its structure (type class).

Style Class	Applies
.uiBorder(item)	Main container element.
.uiBorder(itemTable)	Container of item.
.uiBorder(itemTableItem)	Each item in the item.
.uiBorder(itemTableCaption)	Caption of the item.

3.75 OutputLabel

This component is an extension to the standard output component "client".

The screenshot shows a dialog titled "New Person". Inside, there is a red validation message box containing the text: "J_idt15:name: Validation Error: Value is required." and "Extended Label: Validation Error: Value is required.". Below the message box are three input fields: "Standard Label" (empty), "Extended Label *" (empty), and "Number" (containing the value "1").

Info

9a,	outputLabel
*o " &onent *(a	org.primefaces.component.outputlabel.OutputLabel
*o " &onent 9y&e	org.primefaces.component.OutputLabel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.OutputLabelRenderer
Renderer *(a	org.primefaces.component.outputlabel.OutputLabelRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	True (value to identify the rendering, of the component) when set to false the component will not be rendered.
binding	null	Object	An expression indicating that it is to a server-side UI component instance in a backing bean
value	null	String	Value to display.
accesskey	null	String	The access key attribute is a standard .986 attribute that sets the access key that translates to the element's first letter.
dir	null	String	Direction indication (or text) that does not inherit directionality. Values are LTR and RTR.
encoding	true	Boolean	Defines if value should be encoded or not.

Name	Default	Type	Description
+or	nu()	String,	*o " &onent to attach the (abe(to.
tabindex0	nu()	String,	!o ition in tabbin, order.
tit(e	nu()	String,	Advi ory too(ti& in+or " ation.
ty(e	nu()	String,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu()	String,	Sty(e c(a o+ the co " &onent.
onb(ur	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen co " &onent (o e +ocu .
onc(ic3	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen co " &onent i c(ic3ed.
ondb(c(ic3	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen co " &onent i doub(e c(ic3ed.
on+ocu	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen co " &onent receive +ocu .
on3eydo ' n	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over co " &onent.
on3ey&re	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over co " &onent.
on3eyu&	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over co " &onent.
on " ou edo ' n	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over co " &onent.
on " ou eout	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " co " &onent.
on " ou eover	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto co " &onent.
on " ou eu&	nu()	String,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over co " &onent.

Getting Started with OutputLabel

U a,e i a"e a standard out&ut6abe(E an in&ut co " &onent i a ocated ' ith +or attribute.

```
<p:outputLabel for="input" value="Label" />
<p:inputText id="input" value="#{bean.text}" />
```

Auto Label

If the `label` attribute is not present, the component will automatically generate a label for the input field. This is useful when the label is not defined in the bean.

```
<h:outputLabel for="input" value="Field" />
<p:inputText id="input" value="#{bean.text}" label="Field"/>
```

can be written as

```
<p:outputLabel for="input" value="Field" />
<p:inputText id="input" value="#{bean.text}" />
```

Support for Advanced Components

So far, PrimeFaces does not render standard output components (like inner components) correctly. These components cannot be focused by the keyboard, and they do not have an associated label.

```
<h:outputLabel for="input" value="Can't apply focus" />
<p:outputLabel for="input" value="Can apply focus" />

<p:spinner id="input" value="#{bean.text}" />
```

Validations

When the required attribute is set to true, validation errors will be displayed below the input field. If there are any validation errors, the input field will be disabled.

Skinning

PrimeFaces provides several style classes for skinning input components:

Style Class	Applies
.ui-input-error	to the input element if validation errors occur.
.ui-state-error	to the input element when it is invalid.
.ui-input-error-indicator	to the error indicator element.

3.76 OutputPanel

Output panel is a component with the ability to auto update.

Info

9a,	outputPanel
*o " &onent *(a	org.primefaces.component.outputpanel.OutputPanel
*o " &onent 9y&e	org.primefaces.component.OutputPanel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.OutputPanelRenderer
Renderer *(a	org.primefaces.component.output.OutputPanelRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	Unique identifier of the component.
rendered	true	1oo(ean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin,	nu((7b/ect	An element reference that adds to a server-side UI component in a bean.
ty(e	nu((Strin,	Style of the host container element.
ty(e *(a	nu((Strin,	Style of the host container element.
(ayout	b(oc3	Strin,	Shortcut for the class attribute value where b(oc3 ?de+au(t@ and in(ine.
autoU&date	+a(e	1oo(ean	Enable auto update mode if set true.
de+ered	+a(e	1oo(ean	Deferred mode (load the content after a separate load to avoid a, e load).
de+ered 8 ode	(oad	Strin,	Define deferred loading, "ode" value are I(oadI ?a+ter &a, e (oad@ and Ivi ib(eI ?once the &ane(i vi ib(e on cro(@.
,(oba(+a(e	1oo(ean	Global ajax reCue target is defined by a/a0Statu co " &onentE ettin, ,(oba(to +a(e ' i((not tri, ,er a/a0Statu on de+ered (oadin,..

PlaceHolder

When a component is not rendered no "ar3u& i rendered o+or co " &onent ' ith conditiona(renderin, re, u(ar u&date "echani " "ay not ' or3 ince the "ar3u& to u&date on &a,e doe not e0i t. 7ut&ut!ane(i u e+u(in thi ca e to be u ed a a &(aceho(der.

Su&&o e the rendered condition on bean i +a(e ' hen &a,e i+ (oaded initia(y and earch "ethod on bean et the condition to be true "eanin, databab(e ' i((be rendered a+ter a &a,e ub "it. 9he &rob(e" i a(thou,h &artia(out&ut i ,eneratedE the "ar3u& on &a,e cannot be u&dated ince it doe n't e0i t.

```
<p: dataTable id="tbl" rendered="#{bean.condition}" ...>
    //columns
</p: dataTable>

<p: commandButton update="tbl" actionListener="#{bean.search}" />
```

So(ution i to u e the out&ut!ane(a a &(ace . o(der.

```
<p: outputPanel id="out">
    <p: dataTable id="tbl" rendered="#{bean.condition}" ...>
        //columns
    </p: dataTable>
</p: outputPanel>

<p: commandButton update="out" actionListener="#{bean.list}" />
```

Note that you ' on't need an out&ut!ane(i+ co " " and 1utton ha no u&date attribute &eci+iedE in thi ca e &arent +or " ' i((be u&dated &artia((y i " &(icit(y " a3in, an out&ut!ane(u e ob e(ete.

Deferred Loading

When thi +eature o&tion i enab(edE content o+ &ane(i not (oaded a(on, ' ith the &a,e but (oaded a+ter the &a,e on de " and. Initia(y &ane(di &(ay a (oadin, ani " ation a+ter &a,e (oad to indicate " ore content i co " in, u& and di &(ay content ' ith a/a0 u&date. U in, *deferredMode* o&tionE it i &o ib(e to (oad content not /u t a+ter &a,e (oad ?de+au(t " ode@ but ' hen it beco "e vi ib(e on &a,e cro((a ' e((. 9hi +eature i very u e+u(to increa e &a,e (oad &er+or " anceE a u " e you have one &art o+ the &a,e that ha co " &onent dea(in, ' ith bac3end and ta3in, ti " e@ ' ith de+erred " ode onE re t o+ the &a,e i (oaded in tant(y and ti " e ta3in, &roce i (oaded a+ter ' ard ' ith a/a0.

Layout

7ut&ut!ane(ha t ' o (ayout " ode @
G b(oc3 ?de+au(t@N Render a div
G in(inelN Render a &an

AutoUpdate

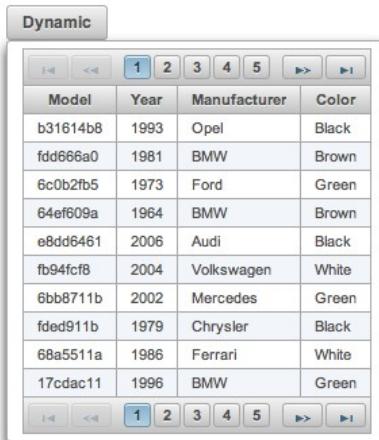
When auto u&date i enab(edE out&ut!ane(co " &onent i u&dated ' ith each a/a0 reCue t auto " atica((y.

Skinning

style and *styleClass* attribute are u ed to ty(e the out&ut!ane(by de+au(t .ui-outputpanel c c(a i added to e(e " ent and .ui-outputpanel-loading ' hen content i (oadin, in de+erred (oadin, ca e.

3.77 OverlayPanel

A dynamic component that can be displayed on top of other content.



Info

9a,	overlayPanel
*o " &onent *(a	org.primefaces.component.overlaypanel.OverlayPanelRenderer
*o " &onent 9y&e	org.primefaces.component.OverlayPanel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.OverlayPanelRenderer
Renderer *(a	org.primefaces.component.overlaypanel.OverlayPanelRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering of the component. If false, the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
' id, et>ar	null	String	<a href="#" of the client id, etc.
type	null	String	Inline type of the element.
type*(a	null	String	Style class of the element.
for	null	String	For reference to the element to display next to.
for ' Event	click	String	Event on target to trigger the element.

Name	Default	Type	Description
hideEvent	c(ic3	String,	Event on target to hide the panel.
hoverEffect	null	String,	Action to display when hovering, the panel.
hideEffect	null	String,	Action to display when hiding the panel.
autoEndBody	0	Boolean	When true, and ended to document body.
onShow	null	String,	*(initial) code called to execute when the panel is shown.
onHide	null	String,	*(initial) code called to execute when the panel is hidden.
"y"	(event to)	String,	!option on the panel relative to the target.
at	(event button)	String,	!option on the target relative to the panel.
dynamicContent	+a(e	Boolean	Define content (adding, "ode.
displayable	true	Boolean	When set true, click in, out hide option on the panel hide the overlay.
hiddenIcon	+a(e	Boolean	Display a close icon to hide the overlay by default if +a(e.

Getting started with OverlayPanel

Overlays need a container element in addition to the content to display. E.g. "default" on targets an overlay attached to a button to show a chart in a modal.

```
<p:commandButton id="chartBtn" value="Basic" type="button" />

<p:overlayPanel for="chartBtn">
    <p:pieChart value="#{chartBean.pieModel}" legendPosition="w"
        title="Sample Pie Chart" style="width:400px;height:300px" />
</p:overlayPanel>
```

Events

Default event on target to show and hide the panel initially. The events are used to "load" in, *showEvent* and *hideEvent* option.

```
<p:commandButton id="chartBtn" value="Basic" type="button" />

<p:overlayPanel showEvent="mouseover" hideEvent="mouseout">
    //content
</p:overlayPanel>
```

Effects

Bind bounce drop effect on mouseover/mouseout events. These effects are available via value or *showEffect* and *hideEffect* option if you'd like display animation.

Positioning

1y de+au(tE (e+t to& corner o+ &ane(i a(i, ned to (e+t botto " corner o+ the tar, et i+ there i enou, h &ace in ' indo ' vie ' &ortE i+ not the &o ition i +(i&&ed on the +(y to +ind the be t (ocation to di &(ay. In order to cu to " i=e the &o ition u e my and at o&tion that ta3e co " bination o+ (e+tE ri, htE botto " and to& e.,.. "ri, ht botto " ".

Dynamic Mode

Dyna " ic " ode enab(e (a=y (oadin, o+ the content in thi " ode content o+ the &ane(i not rendered on &a,e (oad and (oaded /u t be+ore &ane(i ho ' n. A(o content i cached o con ecutive di &(ay do not (oad the content a,ain. 9hi +eature i u e+u(to reduce the &a,e i=e and reduce &a,e (oad ti " e.

Skinning

!ane(re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiBover(ay&ane(8 ain container e(e " ent o+ &ane(

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

Tips

- G Enab(e a&&end9o 1ody ' hen over(ay !ane(i in other &ane(co " &onent (i3e (ayoutE dia(o, ...
- I+ there i a co " &onent ' ith a &o&u& (i3e ca(endarE autoco " &(ete &(aced in ide the over(ay &ane(E &o&u& &art " i, ht e0ceed the boundarie o+ &ane(and c(ic3in, the out ide hide the &ane(. 9hi i unde irab(e o in ca e (i3e thi u e over(ay !ane(' ith *dismissable* +a(e and o&tiona(*showCloseIcon* ettin, .

3.78 Panel

!ane(i a ,rou&in, co " &onent ' ith content to , ,(eE c(o e and " enu inte , ration.



Info

9a,	panel
*o " &onent *(a	org.primefaces.component.panel.Panel
*o " &onent 9y&e	org.primefaces.component.Panel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PanelRenderer
Renderer *(a	org.primefaces.component.panel.PanelRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+iier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7 b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
header	nu((Strin ,	. eader te0t
+ooter	nu((Strin ,	#ootter te0t
to , ,(eab(e	+a(e	1oo(ean	8 a3e &ane(to , ,(eab(e.
to , ,(eS&eed	1000	Inte ,er	S&eed o+ to , ,(in , in " i((i econd
co((a& ed	+a(e	1oo(ean	Render a to , ,(eab(e &ane(a co((a& ed.
ty(e	nu((Strin ,	Sty(e o+ the &ane(
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the &ane(

Name	Default	Type	Description
c(o ab(e	+a(e	1oo(ean	8 a3e &ane(c(o ab(e.
c(o eS&eed	1000	Inte,er	S&eed o+ c(o in , e+ect in " i((i econd
vi ib(e	true	1oo(ean	Render &ane(a vi ib(e.
c(o e9it(e	nu((Strin,	9oo(ti& +or the c(o e button.
to,,(e9it(e	nu((Strin,	9oo(ti& +or the to,,(e button.
" enu9it(e	nu((Strin,	9oo(ti& +or the " enu button.
to,,(e7rientation	vertica(Strin,	De+ine the orientation o+ the to,,(in,E va(id va(ue are vertica(and hori=onta(.
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et

Getting started with Panel

!ane(enca& u(ate other co " &onent .

```
<p:panel>
    //Child components here...
</p:panel>
```

Header and Footer

. eader and #ooter te0t can be &rovided by *header* and *footer* attribute or the corre &ondin , +acet . When a " e attribute and +acet na " e are u edE +acet ' i((be u ed.

```
<p:panel header="Header Text">
    <f:facet name="footer">
        <h:outputText value="Footer Text" />
    </f:facet>

    //Child components here...
</p:panel>
```

Ajax Behavior Events

!ane(&rovide cu to " a/a0 behavior event +or to,,(in, and c(o in , +eature .

Event	Listener Parameter	Fired
to,,(e	or,.&ri " e+ace .event.9o,,(eEvent	When &ane(i e0&anded or co((a& ed.
c(o e	or,.&ri " e+ace .event.*(o eEvent	When &ane(i c(o ed.

Popup Menu

The `p:panel` has built-in support to display a menu. An icon to display the menu is placed at the top-left corner. This feature is enabled by default, and it can be disabled by setting the `closable` attribute to `false`.

```
<p:panel closable="true">
    //Child components here...

    <f:facet name="options">
        <p:menu>
            //Menuitems
        </p:menu>
    </f:facet>
</p:panel>
```

Custom Action

If you'd like to add a custom action to the toolbar, use the `actions` facet.

```
<p:panel>
    <f:facet name="actions">
        <h:commandLink styleClass="ui-panel-titlebar-icon
            ui-corner-all ui-state-default">
            <h:outputText styleClass="ui-icon ui-icon-help" />
        </h:commandLink>
    </f:facet>
    //content
</p:panel>
```

Skinning Panel

There are styles in a panel container which `style` and `styleClass` attributes apply to the panel structure (type class).

Style Class	Applies
.ui-panel	A panel container element.
.ui-panel-header	Header container.
.ui-panel-title	Header title.
.ui-panel-titleicon	Action icon in header.
.ui-panel-content	Panel content.
.ui-panel-footer	Panel footer.

A skinning type class are available for the panel structure (type class).

3.79 PanelGrid

The PanelGrid is an extension to the standard Grid component with additional features such as the "in" and "colspan" attributes.

1995-96 NBA Playoffs							
Conf. Semifinals		Conf. Finals		NBA Finals		Champion	
Seattle	4	Seattle	4	Seattle	2	Chicago	
Houston	0						
Utah	4	Utah	3				
San Antonio	2						
Chicago	4	Chicago	4				
New York	1			Chicago	4		
Atlanta	1	Orlando	0				
Orlando	4						
Finals MVP						Michael Jordan (Chicago)	
Season MVP							
Top Scorer							

Info

9a,	panelGrid
*o " &onent *(a	org.primefaces.component.panelgrid.PanelGridRenderer
*o " &onent 9y&e	org.primefaces.component.PanelGrid
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PanelGridRenderer
Renderer *(a	org.primefaces.component.panelgrid.PanelGridRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	True (value to display the rendering) or the component is rendered when set to false (the component is not rendered).
binding	null	Object	An expression indicating that it is a server-side UI component instance in a backing bean
component	null	Integer	<u>The component identifier in the page

Name	Default	Type	Description
ty(e	null	String,	Inlined style on the element.
ty(e*(a	null	String,	Style class on the element.
co(u "n*(a e	null	String,	*only* a separated (into co(u "n style class.

Getting started with PanelGrid

A basic Grid is a standard one.

```
<p:panelGrid columns="2">
    <h:outputLabel for="firstname" value="Firstname:" />
    <p:inputText id="firstname" value="#{bean.firstname}" label="Firstname" />

    <h:outputLabel for="surname" value="Surname:" />
    <p:inputText id="surname" value="#{bean.surname}" label="Surname"/>
</p:panelGrid>
```

Header and Footer

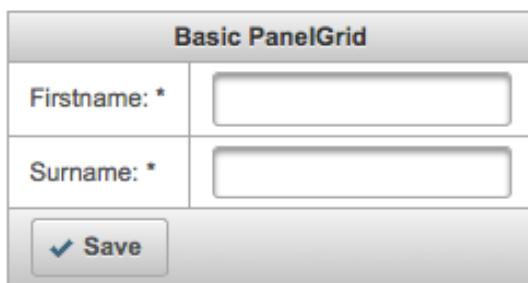
A Grid provides facets for header and footer content.

```
<p:panelGrid columns="2">
    <f:facet name="header">
        Basic PanelGrid
    </f:facet>

    <h:outputLabel for="firstname" value="Firstname: *"/>
    <p:inputText id="firstname" value="#{bean.firstname}" label="Firstname" />

    <h:outputLabel for="surname" value="Surname: *"/>
    <p:inputText id="surname" value="#{bean.surname}" label="Surname"/>

    <f:facet name="footer">
        <p:commandButton type="button" value="Save" icon="ui-icon-check" />
    </f:facet>
</p:panelGrid>
```



Rowspan and Colspan

A Grid supports rowspan and colspan options across the entire grid and columns can be defined uniquely.

```

<p:panelGrid>
    <p:row>
        <p:column rowspan="3">AAA</p:column>
        <p:column colspan="4">BBB</p:column>
    </p:row>

    <p:row>
        <p:column colspan="2">CCC</p:column>
        <p:column colspan="2">DDD</p:column>
    </p:row>

    <p:row>
        <p:column>EEE</p:column>
        <p:column>FFF</p:column>
        <p:column>GGG</p:column>
        <p:column>HHH</p:column>
    </p:row>
</p:panelGrid>

```

Skinning

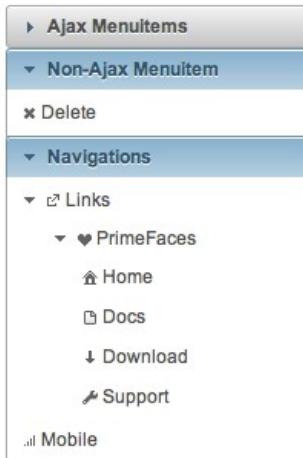
The Grid re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the
 (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB&ane(,rid	8 ain container e(e " ent o+ &ane(Grid.
.uiB&ane(,ridBheader	. eader.
.uiB&ane(,ridBouter	#ooter.
.uiB&ane(,ridBeven	Even nu " bered ro' .
.uiB&ane(,ridBodd	7 dd nu " bered ro' .

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.80 PanelMenu

This component is a hybrid component combining accordion and tree components.



Info

9a,	panelMenu
*o "&onent *(a	org.primefaces.component.panelmenu.PanelMenu
*o "&onent 9y&e	org.primefaces.component.PanelMenu
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PanelMenuRenderer
Renderer *(a	org.primefaces.component.panelmenu.PanelMenuRenderer

Attributes

Name	Default	Type	Description
id	nu((String	Unique identifier for the component.
rendered	true	Boolean	Value indicating whether the component is rendered. If false, it will not be rendered.
binding	nu((Object	An expression or reference that binds to a server-side UI component in a managed bean.
" value	nu((String	Value displayed in the component.
type	nu((String	Inline type for the component.
style*(a	nu((String	Style class for the component.
' id, et>ar	nu((String	Client-side id attribute.

Getting started with PanelMenu

The following example shows how to use the `PanelMenu` component. It consists of three submenus: "Ajax Menuitems", "Non-Ajax MenuItem", and "Navigations". The "Ajax Menuitems" submenu contains two items: "Save" and "Update". The "Non-Ajax MenuItem" submenu contains one item: "Delete". The "Navigations" submenu contains three items: "Home", "Docs", and "Support". The "Mobile" item has an outcome of "/mobile/index".

```
<p:panelMenu style="width:200px">
    <p:submenu label="Ajax Menuitems">
        <p:menuItem value="Save" actionListener="#{buttonBean.save}" />
        <p:menuItem value="Update" actionListener="#{buttonBean.update}" />
    </p:submenu>

    <p:submenu label="Non-Ajax MenuItem">
        <p:menuItem value="Delete" actionListener="#{buttonBean.delete}" ajax="false"/>
    </p:submenu>

    <p:submenu label="Navigations" >
        <p:submenu label="Links" icon="ui-icon-extlink">
            <p:submenu label="PrimeFaces" icon="ui-icon-heart">
                <p:menuItem value="Home" url="http://www.primefaces.org" />
                <p:menuItem value="Docs" url="http://www.primefaces.org/..." />
                <p:menuItem value="Support" url="http://www.primefaces.org/..." />
            </p:submenu>
        </p:submenu>
        <p:menuItem value="Mobile" outcome="/mobile/index" />
    </p:submenu>
</p:panelMenu>
```

Skinning

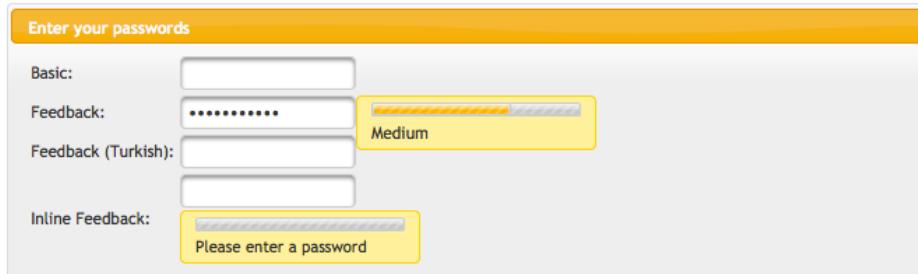
The following table lists the available style classes for the `PanelMenu` component.

Style Class	Applies
.ui-panelmenu-enu	8 ain container e(e " ent.
.ui-panelmenu-enuheader	. eader o+ a &ane(.
.ui-panelmenu-enucontent	#ooter o+ a &ane(.
.ui-panelmenu-enu .ui-panelmenu-item	9ree container.
.ui-panelmenu-enu .ui-panelmenu-item-enuite	A " enuite " in tree.

A 3innin, ty(e c(a e are ,(oba(ee the " ain the " in, ection +or " ore in+or " ation.

3.81 Password

Password component is an extended version of standard inputSecret component with theme integration and strength indicator.



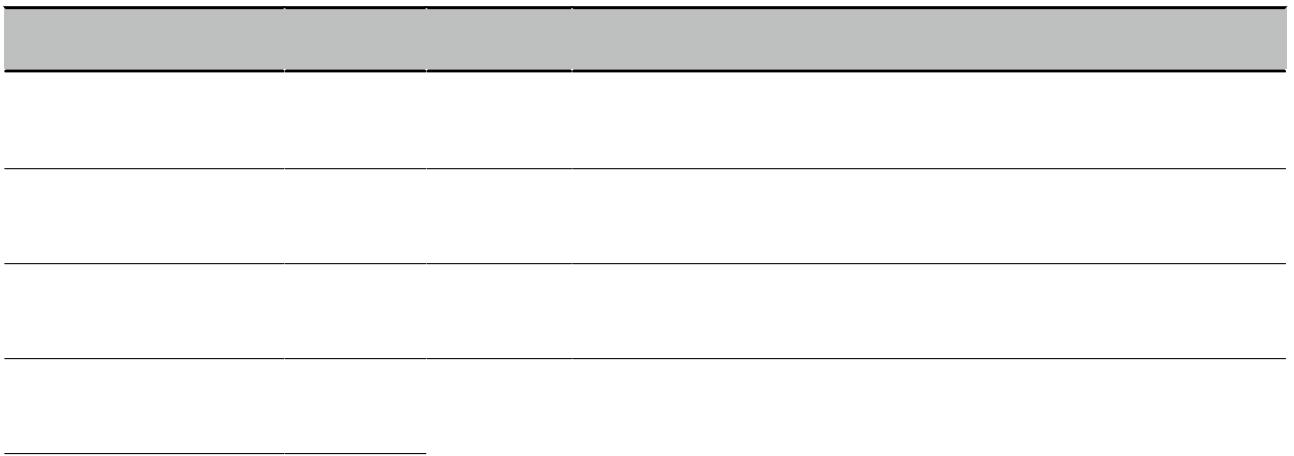
Info

9a,	password
*o " &onent *(a	org.primefaces.component.password.Password
*o " &onent 9y&e	org.primefaces.component.Password
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PasswordRenderer
Renderer *(a	org.primefaces.component.password.PasswordRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Value indicating whether the component is rendered. If false, it will not be rendered.
binding	null	Object	An expression that binds the component to a server-side UI component in a backing bean.
value	null	Object	A value of the component that can be either an E6 expression or a (iterable) object.
converter	null	Converter	An expression or a converter that defines a converter for the component. When it's an E6 expression, it's resolved to a converter in context. In case it's a static text, it's used to refer to a converter id.
immediate	false	Boolean	When set to true, validation (optional) is executed at the same time as the component's value is set.
required	false	Boolean	Specifies if the component is required.

Name	Default	Type	Description
va(imator	nu((8 ethod E0&r	A " ethod e0&re ion that refer to a " ethod va(ivation , the in&ut.
va(ue*han,e6i tener	nu((8 ethod E0&r	A " ethod bindin , e0&re ion that refer to a " ethod +or hand(in , a va(uechan ,eevent
reCured 8 e a,e	nu((Strin ,	8 e a,e to be di &(ayed ' hen reCured +ie(d va(ivation +ai(.
converter 8 e a,e	nu((Strin ,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(imator 8 e a,e	nu((Strin ,	8 e a,e to be di &(ayed ' hen va(ivation +ie(d .
+eedbac3	+a(e	1oo(ean	Enab(e tren ,th indicator.
in(ine	+a(e	boo(ean	Di &(ay +eedbac3 in(ine rather than u in , a &o&u&.
&ro "&t6abe(!(ea e enter a &a ' ord	Strin ,	6abe(o+ &ro " &t.
(eve(1	Inte ,er	6eve(o+ ecurity.
' ea36abe(Wea3	Strin ,	6abe(o+ ' ea3 &a ' ord.
,ood6abe(Good	Strin ,	6abe(o+ ,ood &a ' ord.
tron ,6abe(Stron ,	Strin ,	6abe(o+ tron , &a ' ord.
redi &(ay	+a(e	1oo(ean	Whether or not to di &(ay &reviou va(ue.
" atch	nu((Strin ,	Id o+ another &a ' ord co " &onent to " atch va(ue a,ain t.
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
acce 3ey	nu((Strin ,	Acce 3ey that ' hen &re ed tran +er +ocu to the in&ut e(e " ent.
a(t	nu((Strin ,	A(ternate te0tua(de cri&tion o+ the in&ut +ie(d.
autoco " &(ete	nu((Strin ,	*ontro(bro ' er autoco " &(ete behavior.
dir	nu((Strin ,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
di ab(ed	+a(e	1oo(ean	Di ab(e in&ut +ie(d
(abe(nu((Strin ,	A(oca(i=ed u er &re entab(e na " e.
(an,	nu((Strin ,	*ode de cribin , the (an,ua,e u ed in the ,enerated " ar3u& +or thi co " &onent.
" a0(en,th	nu((Inte ,er	8 a0i " u " nu " ber o+ character that " ay be entered in thi +ie(d.
onb(ur	nu((Strin ,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu .



Getting Started with Password

A password field can be used to store sensitive information and it is standard input type. When `feedback` option is enabled a strength indicator is displayed.

```
<p:password value="#{bean.password}" feedback="true|false" />
```

```
public class Bean {

    private String password;

    public String getPassword() { return password; }
    public void setPassword(String password) { this.password = password; }
}
```

I18N

Altough the labels are in English by default you can provide custom to them. There are three levels of strength indicators in, a password, and a strength indicator.

```
<p:password value="#{bean.password}" promptLabel="Lütfen şifre giriniz"
weakLabel="Zayıf" goodLabel="Orta seviye" strongLabel="Güçlü" feedback= "true"/>
```

Inline Strength Indicator

If you want to have an inline strength indicator instead of an inline indicator you enable it in the "ode" attribute.

```
<p:password value="#{mybean.password}" inline="true" feedback= "true"/>
```

Confirmation

A confirmation action is a code that can be used to provide an easy way to identify the entered password. The other password component's id should be used to define the `match` option.

```
<p:password id="pwd1" value="#{passwordBean.password6}" feedback="false"
            match="pwd2" label="Password 1" required="true"/>

<p:password id="pwd2" value="#{passwordBean.password6}" feedback="false"
            label="Password 2" required="true"/>
```

Skinning

Structure selector or a password are:

Name	Applies
.ui-password	Input element only.
.ui-password input	Overlays the strength indicator.
.ui-password-outer	Strength indicator.

Name	Applies
.uiB&a ' ordBin+o	Stren, th (abe).

A 3innin, ty(e c(a e are ,(oba(ee the "ain the" in, ection +or "ore in+or" ation.

3.82 PhotoCam

This component is used to take a photo with the camera and send the image to the server backend.

Info

9a,	photoCam
*o "&onent *(a	org.primefaces.component.photocam.PhotoCam
*o "&onent 9y&e	org.primefaces.component.PhotoCam
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PhotoCamRenderer
Renderer *(a	org.primefaces.component.photocam.PhotoCamRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Value indicating whether the component is rendered, or the component is hidden and not to be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean
value	null	Object	>A value or the component that can be either an Expression or a converter
converter	null	Converter or String	An expression or a converter that defines a converter for the component. When it's an Expression or a converter, it's referred to a converter instance. In case it's a static text, it's used to refer to a converter id
immediate	false	Boolean	When set to true, the validation is executed at the same time as the component's value is set.
required	false	Boolean	The component is required
validator	null	Method	A method binding, expression that refers to a method validation, the input
valueChangeListener	null	Method	A method binding, expression that refers to a method for handling a valuechange event
requiredValidator	null	String	The attribute is used to define the required validation method.
converterValidator	null	String	The attribute is used to define the converter validation method.
validatorValidator	null	String	The attribute is used to define the validation validation method.

Name	Default	Type	Description
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
ty(e	nu((Strin,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the co " &onent.
&roce	nu((Strin,	Identifi+ier o+ co " &onent to &roce durin, ca&ture.
u&date	nu((Strin,	Identifi+ier o+ co " &onent to u&date durin, ca&ture.
(i tener	nu((8 ethod E0&r	8 ethod e0&re ion to (i ten to ca&ture event .

Getting started with PhotoCam

*a&tur e i tri , , ered via c(ient ide a&i' *capture* " ethod. A(o a " ethod e0&re ion i nece ary to inv03e ' hen an i " a,e i ca&tured. Sa " &(e be(o ' ca&ture an i " a,e and ave it to a directory.

```
<h:form>
    <p:photoCam widgetVar="pc" listener="#{photoCamBean.oncapture}" update="photos"/>
        <p:commandButton type="button" value="Capture" onclick="PF('pc').capture()"/>
</h:form>
```

```
public class PhotoCamBean {
    public void oncapture(CaptureEvent captureEvent) {
        byte[] data = captureEvent.getData();

        ServletContext servletContext = (ServletContext)
FacesContext.getCurrentInstance().getExternalContext().getContext();
        String newFileName = servletContext.getRealPath("") + File.separator +
"photocam" + File.separator + "captured.png";

        FileImageOutputStream imageOutput;
        try {
            imageOutput = new FileImageOutputStream(new File(newFileName));
            imageOutput.write(data, 0, data.length);
            imageOutput.close();
        }
        catch(Exception e) {
            throw new FacesException("Error in writing captured image.");
        }
    }
}
```

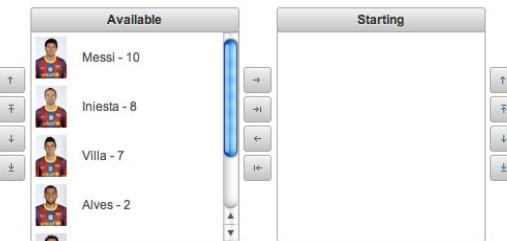
Notes

!photo*a" i a +(a hE canva and /ava cri&t o(ution.

It i not u&&orted in IE at the " o " ent and thi ' i((be ' or3ed on in +uture ver ion .

3.83 PickList

!ic36i t i u ed +or tran +errin, data bet ' een t ' o di+erent co((ection .



Info

9a,	pickList
*o "&onent *(a	org.primefaces.component.picklist.Panel
*o "&onent 9y&e	org.primefaces.component.PickList
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PickListRenderer
Renderer *(a	org.primefaces.component.picklist.PickListRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering, or the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component in a backing bean
value	null	Object	>a(ue or the component than can be either an E6 expression or a iterator
converter	null	Converter or String	An expression or a converter that defines a converter for the component. When it's an E6 expression it's resolved to a converter in context. In case it's a static text it's used directly to a converter id
immediate	false	Boolean	When set true it's validation logic is executed at any time before or after this component.
required	false	Boolean	Specifies if a component is required
validator	null	Method	A method binding expression that refers to a method validation, the input

Name	Default	Type	Description
valueChangeListener	null	Method	A " method binding, exposed to a " method or handler, a valueChange event.
requiredValue	null	String	The value to be displayed when required validation failed.
converter	null	String	The value to be displayed when conversion failed.
validator	null	String	The value to be displayed when validation failed.
var	null	String	<a value of the iterator.
itemLabel	null	String	Label of an item.
itemLabelExpr	null	Object	Label of an item.
itemContainer	null	String	Style of the main container.
itemContainerLabel	null	String	Style of a container label.
clientId	null	String	<a value of the client identifier.
disabled	false	Boolean	Disabled the component.
effect	null	String	<a value of the animation to display.
effectSpeed	null	String	Speed of the animation.
addAction	Add	String	Method of add button.
addActionA	AddAction	String	Method of add all button.
remove	Remove	String	Method of remove button.
removeAll	RemoveAll	String	Method of remove all button.
moveUp	MoveUp	String	Method of move up button.
moveDown	MoveDown	String	Method of move down button.
moveFirst	MoveFirst	String	Method of move first button.
moveLast	MoveLast	String	Method of move last button.
source	Source	String	Selected source (initial).
target	Target	String	Selected target (initial).
onTransfer	null	String	Transfer code (batch execute when an item is transferred to another).
(action)	null	String	Action (called after & before enter and tab).
itemDisabled	false	Boolean	Specifies if an item can be selected or not.
disabledSource	false	Boolean	Disabled the source (initial).

Name	Default	Type	Description
ho ' 9ar, et#i(ter	+a(e	1oo(ean	Di &(ay and in&ut +i(ter +or tar, et (i t.
+i(ter 8 atch 8 ode	tart With	Strin,	8 atch " ode +or +i(terin, E va(id va(ue are tart WithE contain E end With and cu to " .
+i(ter#unction	nu((Strin,	<a " e o+ the /ava cri&t +unction +or cu to " +i(terin,..
ho ' *hec3bo0	+a(e	1oo(ean	When trueE a chec3bo0 i di &(ayed ne0t to each ite " .
(abe(Di &(ay	too(ti&	Strin,	De+ine ho ' the button (abe(di &(ayedE va(id va(ue are Itoo(ti&I ?de+au(t@ and Iin(ineI.

Getting started with PickList

You need to create cu to " "ode(ca((ed *org.primefaces.model.DualListModel* to u e !ic36i t. A the na"e u,,e t it con i t o+t'o(i t E one i the ource(i t and the other i the tar,et. A the +ir t e0a "&(e ' e'((create a Dua(6i t 8 ode(that contain ba ic Strin, .

```
public class PickListBean {

    private DualListModel<String> cities;

    public PickListBean() {
        List<String> source = new ArrayList<String>();
        List<String> target = new ArrayList<String>();

        citiesSource.add("Istanbul");
        citiesSource.add("Ankara");
        citiesSource.add("Izmir");
        citiesSource.add("Antalya");
        citiesSource.add("Bursa");

        //more cities

        cities = new DualListModel<String>(citiesSource, citiesTarget);
    }

    public DualListModel<String> getCities() {
        return cities;
    }

    public void setCities(DualListModel<String> cities) {
        this.cities = cities;
    }
}
```

And bind the citie dua((i t to the &ic3(i t

```
<p:pickList value="#{pickListBean.cities}" var="city"
            itemLabel="#{city}" itemValue="#{city}">
```

When the encoder is used " instead of the dual reference it is used ' with the new value and you can access the value ' with Dua(6 instead of Dua(6).setSource() and Dua(6).setTarget(), setTarget() and.

POJOs

So to the tip " you ' ou(d deal ' ith co " &(e0 &o/o rather than i " &(e type (i3e String, .9hi u e ca e i no different encode&t the addition of a converter. #o((o ' in, &ic36i t di &(ay a (i t o+&(ayer ?na " EE a,e ...@.

```
<p:pickList value="#{pickListBean.players}" var="player"
    itemLabel="#{player.name}" itemValue="#{player}" converter="player">
```

! (ayer*onverter in thi ca e hou(d i " &(e " ent javax.faces.convert.Converter contract and i " &(e " ent ,etA String, ,etA 7b/ect " ethod . <ote that a converter is a(ay nece ary +or &ri " itive ty&e (i3e (on, E inte,erE boo(ean a ' e(.

In addition to " content in stead of i " &(e trin, can be displayed by using, co(u " n .

```
<p:pickList value="#{pickListBean.players}"
    var="player" iconOnly="true" effect="bounce"
    itemValue="#{player}" converter="player"
    showSourceControls="true" showTargetControls="true">
    <p:column style="width:25%">
        <p:graphicImage value="/images/barca/#{player.photo}"/>
    </p:column>
    <p:column style="width:75%">
        #{player.name} - #{player.number}
    </p:column>
</p:pickList>
```

Reordering

! ic36i t u&&ort reordering, o+ ource and tar, et (i t E the e are enab(ed by *showSourceControls* and *showTargetControls* option .

Effects

An animation is displayed when transitioning, ' hen ite " to another or reordering, a (i t E de+au(t effect is made and #o((o ' in, option are available to be a&&(ied u in, *effect* attribute b(indE bounceE c(i& dro&E e0&&odeE +o(dE hi ,h(i ,htE &u++E &u(ateE ca(eE ha3eE i=e and (ide. *effectSpeed* attribute is used to control the animation &eedE va(id value are slowE normal and fast.

Captions

*action types or (i t are defined ' ith facet names " ed *sourceCaption* and *targetCaption*

```
<p:pickList value="#{pickListBean.cities}" var="city"
    itemLabel="#{city}" itemValue="#{city}" onTransfer="handleTransfer(e)">
    <f:facet name="sourceCaption">Available</f:facet>
    <f:facet name="targetCaption">Selected</f:facet>
</p:pickList>
```

Filtering

!ic36i t &rovide bui(tBin c(ient ide +i(terin,, #i(terin, i enab(ed by ettin, the corre &ondin, +i(terin, attribute o+ a (i t. #or ource (i t thi i *showSourceFilter* and +or tar,et (i t it i *showTargetFilter*. De+au(t "atch "ode i tart With and contain E end With are a(o avai(ab(e o&tion .

When you need to a cu to " " atch " ode et *filterMatchMode* to cu to " and ' rite a /ava cri&t +unction that ta3e ite " 6abe(and +i(ter>a(ue a &ara " eter . Return +a(e to hide an ite " and true to di &(ay.

```
<p:pickList value="#{pickListBean.cities}" var="city"
    itemLabel="#{city}" itemValue="#{city}"
    showSourceFilter="true" showTargetFilter="true"
    filterMatchMode="custom" filterFunction="myfilter">
</p:pickList>
```

```
function myfilter(itemLabel, filterValue) {
    //return true or false
}
```

onTransfer

If you'd (i3e to e0ecute cu to " /ava cri&t ' hen an ite " i tran +erred bind your /ava cri&t +unction to *onTransfer* attribute.

```
<p:pickList value="#{pickListBean.cities}" var="city"
    itemLabel="#{city}" itemValue="#{city}" onTransfer="handleTransfer(e)">
```

```
<script type="text/javascript">
    function handleTransfer(e) {
        //item = e.item
        //fromList = e.from
        //toList = e.toList
        //type = e.type (type of transfer; command, dblclick or dragdrop)
    }
</script>
```

Ajax Behavior Events

!ic36i t &rovide *transfer* a the de+au(t and on(y a/a0 behavior event that i +ired ' hen an ite " i "oved +ro " one (i t to the other. Eo a " &(e be(o ' de " on trate ho ' to u e thi event.

```
<p:pickList value="#{pickListBean.cities}" var="city"
    itemLabel="#{city}" itemValue="#{city}">
    <p:ajax event="transfer" listener="#{pickListBean.handleTransfer}" />
</p:pickList>
```

```

public class PickListBean {

    //DualListModel code

    public void handleTransfer(TransferEvent event) {
        //event.getItems() : List of items transferred
        //event.isAdd() : Is transfer from source to target
        //event.isRemove() : Is transfer from target to source
    }
}

```

Skinning

The skinning of the `ui:pickList` component is done in a "skin container" which *style* and *styleClass* attributes are applied to the internal structure of the component.

Style Class	Applies
.ui:pickList	8 skin container element
.ui:pickListItem	6 item or a &ic3(item)
.ui:pickListSource	Source item
.ui:pickListTarget, .ui:pickListControl	9 target item or reorder control
.ui:pickListTargetControl	* container element or target item or reorder control
.ui:pickListButton	1 button or a &ic3(button)
.ui:pickListButtonAbove	8 above button
.ui:pickListButtonBelow	8 below button
.ui:pickListButtonDown	8 down button
.ui:pickListButtonBottom	8 bottom button
.ui:pickListAdd	Add button
.ui:pickListAddBefore	Add after button
.ui:pickListAddAfter	Re above after button
.ui:pickListAdd	Add button

A skinning, ty(e c(a e are , (oba(E ee the " skin the " in, ection +or " ore in+or " ation.

3.84 Poll

!o((i an a/a0 co " &onent that ha the abi(ity to end &eriodica(a/a0 reCue t .

Info

9a,	poll
*o " &onent *(a	org.primefaces.component.poll.Poll
*o " &onent 9y&e	org.primefaces.component.Poll
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.PollRenderer
Renderer *(a	org.primefaces.component.poll.PollRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to identify the renderer, of the component. If set to false the component will not be rendered.
binding	null	Object	An object reference that maps to a server-side UI component instance in a backing bean.
' id, et>ar	null	String	 id, et.
interval	%	Integer	Interval in seconds to do periodic ajax requests.
update	null	String	*o " &onent? @ to be updated with ajax.
(i tener	null	Method	A method reference to invoke by onclick,.
i " " ediate	false	Boolean	Boolean value that determines if the action is performed at the same time as the click or after the click.
async	false	Boolean	When set to true ajax requests are not queued.
&roce	null	String	*o " &onent id? @ to &roce &artia((y instead of ' href'.
onstart	null	String	Java script handler to execute before ajax request begins.
oncomplete	null	String	Java script handler to execute when ajax request completes.
onsuccess	null	String	Java script handler to execute when ajax request succeeds.
onerror	null	String	Java script handler to execute when ajax request fails.

Name	Default	Type	Description
,(oba(true	1oo(ean	G(oba(a/a0 reCue t are (i tened by a/a0Statu co " &onentE ettin, ,(oba(to +a(e ' i((not tri, ,er a/a0Statu .
de(ay	nu((Strin ,	I+(e than delay " i((i econd e(a& e bet ' een ca((to request() on(y the " o t recent one i ent and a((other reCue t are di carded. I+ thi o&tion i not &eci+iedE or i+ the va(ue o+ delay i the (itera(trin, lnone) ' ithout the Cuote E no de(ay i u ed.
&artia(Sub " it	+a(e	1oo(ean	Enab(e eria(i=ation o+ va(ue be(on,in, to the &artia((y &roce ed co " &onent on(y.
autoStart	true	1oo(ean	In autoStart " odefE &o((in, tart auto " atica((y on &a,e (oadE to tart &o((in, on de " and et to +a(e.
to&	+a(e	1oo(ean	Sto& &o((in, ' hen true.
re et>a(ue	+a(e	1oo(ean	I+ trueE (oca(va(ue o+ in&ut co " &onent to be u&dated ' ithin the a/a0 reCue t ' ou(d be re et.
i,noreAutoU&date	+a(e	1oo(ean	I+ trueE co " &onent ' hich autoU&datePItrueI ' i((not be u&dated +or thi reCue t. I+ not &eci+iedE or the va(ue i +a(eE no uch indication i " ade.

Getting started with Poll

!o((be(o ' invo3e incre " ent " ethod on *ounter1ean every % econd and *txt_count* i u&dated ' ith the ne ' va(ue o+ the count variab(e. <ote that &o((" u t be ne ted in ide a +or " .

```
<h:outputText id="txt_count" value="#{counterBean.count}" />
<p:poll listener="#{counterBean.increment}" update="txt_count" />
```

```
public class CounterBean {
    private int count;

    public void increment() {
        count++;
    }

    public int getCount() {
        return this.count;
    }

    public void setCount(int count) {
        this.count = count;
    }
}
```

Tuning timing

By default the periodic interval is 1 second. This can be changed with the interval attribute. #000 in, or every 5 seconds.

```
<h:outputText id="txt_count" value="#{counterBean.count}" />
<p:poll listener="#{counterBean.increment}" update="txt_count" interval="5" />
```

Start and Stop

The poll can be started and stopped using client-side JavaScript.

```
<h:form>
    <h:outputText id="txt_count" value="#{counterBean.count}" />
    <p:poll interval="5" actionListener="#{counterBean.increment}"
        update="txt_count" widgetVar="myPoll" autoStart="false" />
    <a href="#" onclick="PF('myPoll').start();">Start</a>
    <a href="#" onclick="PF('myPoll').stop();">Stop</a>
</h:form>
```

Bind a boolean variable to the stop attribute and set it to true at any arbitrary time.

3.85 Printer

The `p:printer` component is used to print content from the page. It can be attached to any component (like a button or an outputText) or to a specific outputLink.

Info

9a,	printer
1behavior *(a	org.primefaces.component.behavior.Printer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Logical value to specify if the component should be rendered. If false, it will not be rendered.
binding	null	Object	An expression reference that maps to a server-side UI component instance in a backing bean.
target	null	String	Name of the component to print.

Getting started with the Printer

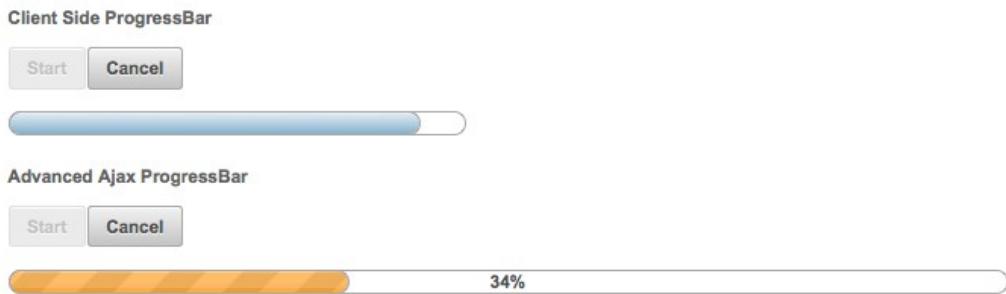
Attach the `p:printer` component to any component (like a button or an outputText) or to a specific outputLink.

```
<h:commandButton id="btn" value="Print">
    <p:printer target="output" />
</h:commandButton>
<h:outputText id="output" value="PrimeFaces Rocks!" />

<h:outputLink id="lnk" value="#">
    <p:printer target="image" />
    <h:outputText value="Print Image" />
</h:outputLink>
<p:graphicImage id="image" value="/images/nature1.jpg" />
```

3.86 ProgressBar

!ro,re 1ar i a &roce tatu indicator that can either 'or3 &ure(y on c(ient ide or interact 'ith erver ide u in, a/a0.



Info

9a,	progressBar
*o " &onent *(a	org.primefaces.component.progressbar.ProgressBar
*o " &onent 9y&e	org.primefaces.component.ProgressBar
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ProgressBarRenderer
Renderer *(a	org.primefaces.component.progressbar.ProgressBarRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et
va(ue	0	Inte ,er	>a(ue o+ the &ro,re bar
di ab(ed	+a(e	1oo(ean	Di ab(e or enab(e the &ro,re bar
a/a0	+a(e	1oo(ean	S&eci+ie the " ode o+ &ro,re 1arf in a/a0 " ode &ro,re va(ue i retrieved +ro " a bac3in, bean.
interva()000	Inte ,er	Interva(in econd to do &eriodic reCue t in a/a0 " ode.

Name	Default	Type	Description
ty(e	nu((Strin ,	In(ine ty(e o+ the " ain container e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the " ain container e(e " ent.
(abe(9e " &(ate	Vva(ueW	Strin ,	9e " &(ate o+ the &ro,re (abe(.
di &(ay7n(y	+a(e	1oo(ean	Enab(e tatic di &(ay " ode.
,(oba(true	1oo(ean	G(oba(a/a0 reCue t are (i tened by a/a0Statu co " &onentE ettin, ,(oba(to +a(e ' i((not tri,,er a/a0Statu .

Getting started with the ProgressBar

!ro,re 1ar ha t' o " ode E Ic(ientI?de+au(t@ or Ia/a0I. #o((o ' in, i a &ure c(ient ide &ro,re 1ar.

```
<p:progressBar widgetVar="pb" />

<p:commandButton value="Start" type="button" onclick="start()" />
<p:commandButton value="Cancel" type="button" onclick="cancel()" />

<script type="text/javascript">
    function start() {
        this.progressInterval = setInterval(function(){
            PF('pb').setValue(PF('pb').getValue() + 10);
        }, 2000);
    }

    function cancel() {
        clearInterval(this.progressInterval);
        PF('pb').setValue(0);
    }
</script>
```

Ajax Progress

A/a0 " ode i enab(ed by ettin, a/a0 attribute to true in thi ca e the va(ue de+ined on a " ana,ed bean i retrieved &eriodica((y and u ed to u&date the &ro,re .

```
<p:progressBar ajax="true" value="#{progressBean.progress}" />
```

```
public class ProgressBean {

    private int progress;

    //getter-setter
}
```

Interval

The progress bar is based on the interval attribute, which defines the time interval between each update. The value can be set to a different value using the interval attribute.

```
<p:progressBar interval="5000" />
```

Ajax Behavior Events

The progress bar provides a complete event that fired when the progress bar is completed. This event has no return value.

```
public class ProgressBean {

    private int progress;

    public void handleComplete() {
        //Add a faces message
    }

    public int getProgress() {
        return progress;
    }

    public void setProgress(int progress) {
        this.progress = progress;
    }
}
```

```
<p:progressBar value="#{progressBean.progress}" ajax="true">
    <p:ajax event="complete" listener="#{progressBean.handleComplete}"
        update="messages" />
</p:progressBar>

<p:growl id="messages" />
```

Display Only

If you have a static value or the value of the progress bar on each page, you can use a static progress bar.

```
<p:progressBar value="50" displayOnly="true" />
```

Client Side API

Widjet PrimeFaces.widget.ProgressBar

Method	Params	Return Type	Description
getProgress()		Number	Return current value

Method	Params	Return Type	Description
setVaue(Vaue)	Vaue>a(ue to di &(ay	void	Set current Vaue
start()	B	void	Start a/a0 &ro, re bar
cancel()	B	void	Stop a/a0 &ro, re bar

Skinning

!ro, re 1ar re ide in a "ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB&ro, re bar	8 ain container.
.uiB&ro, re barBva(ue	>a(ue o+ the &ro, re bar
.uiB&ro, re barBabe(!ro, re (abe(.

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or "ore in+or" ation.

3.87 RadioButton

Radio button is a helper component SelectOneRadio to implement "radio" layout.

Info

9a,	radioButton
*o "&onent *(a	org.primefaces.component.radioButton.RadioButton
*o "&onent 9y&e	org.primefaces.component.RadioButton
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.RadioButtonRenderer
Renderer *(a	org.primefaces.component.radioButton.RadioButtonRenderer

Attributes

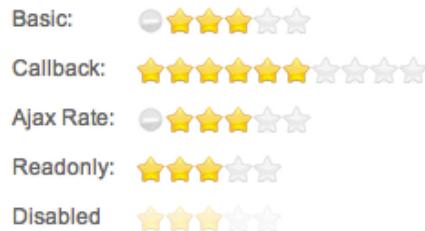
Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
disabled	false	Boolean	Disables the component.
itemIndex	null	Integer	The index of the item selected by the user.
onchange	null	String	JavaScript code to execute on state change.
for	null	String	Id of the Radio component to attach to.
type	null	String	Indicates the type of the component.
type*(a	null	String	Style class of the container.
tabindex	null	String	Specifies the tab order of the component in tab navigation.

Getting started with RadioButton

See `radio` part in `SelectOneRadio` section for more information.

3.88 Rating

Ratin, co " &onent +eature a tar ba ed ratin, y te".



Info

9a,	rating
*o " &onent *(a	org.primefaces.component.rating.Rating
*o " &onent 9y&e	org.primefaces.component.Rating
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.RatingRenderer
Renderer *(a	org.primefaces.component.rating.RatingRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7 b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
va(ue	nu((7 b/ect	>a(ue o+ the co " &onent than can be either an E6 e0&re ion o+ a (itera(te0t
converter	nu((*onverter(Strin ,	An e(e0&re ion or a (itera(te0t that de+ine a converter +or the co " &onent. When it' an E6 e0&re ionE it' re o(ved to a converter in tance. In ca e it' a static te0tE it " u t re+er to a converter id
i " " ediate	0	1oo(ean	1oo(ean va(ue that &eci+ie the (i+ecyc(e &ha e the va(ue*han,eEvent hou(d be &roce edE ' hen true the event ' i((be +ired at Ia&&(y reCue t va(ue IE i " " ediate i et to +a(eE va(ue*han,e Event are +ired in I&roce va(idation I &ha e

Name	Default	Type	Description
reCquired	+a(e	1oo(ean	8 ar3 co " &onent a reCquired
va(idator	nu((8 ethodE0&r	A " ethod bindin, e0&re ion that re+er to a " ethod va(idation, the in&ut
va(ue*han,e6i tener	nu((8 ethodE0&r	A " ethod bindin, e0&re ion that re+er to a " ethod +or hand(in, a va(uchan, eevent
reCquired 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen reCquired +ie(d va(idation +ai(.
converter 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator 8 e a,e	nu((Strin,	8 e a,e to be di &(ayed ' hen va(idation +ie(d .
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
tar	5	Inte,er	<u " ber o+ tar to di &(ay
di ab(ed	+a(e	1oo(ean	Di ab(e u er interaction
readon(y	+a(e	1oo(ean	Di ab(e u er interaction ' ithout di ab(ed vi ua(.
onRate	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen rate ha&&en .
ty(e	nu((Strin,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the co " &onent.
cance(true	1oo(ean	When enab(ed di &(ay a cancel icon to re et.

Getting Started with Rating

Ratin , i an in&ut co " &onent that ta3e a doub(e variab(e a it' va(ue.

```
public class RatingBean {
    private Integer rating;
    //getter-setter
}
```

```
<p:rating value="#{ratingBean.rating}" />
```

Number of Stars

De+au(t nu " ber o+ tar i 5E i+ you need (e or " ore tar u e the tar attribute. #o((o ' in, ratin, con i t o+ 10 tar .

```
<p:rating value="#{ratingBean.rating}" stars="10"/>
```

Display Value Only

In case here you only want to use the rating, content to display the rating, value and disabled user interaction set `readonly` to true. Using, `disabled` attribute does the same but adds disabled via user interface.

Ajax Behavior Events

Rating provides `rate` and `cancel` ajax behavior events. A defined (in tener or rate event) will be executed by an, an `org.primefaces.event.RateEvent` as a parameter and cancel event will be invoked with no parameter.

```
<p:rating value="#{ratingBean.rating}">
    <p:ajax event="rate" listener="#{ratingBean.handleRate}" update="msgs" />
    <p:ajax event="cancel" listener="#{ratingBean.handleCancel}" update="msgs" />
</p:rating>
<p:messages id="msgs" />
```

```
public class RatingBean {

    private Integer rating;

    public void handleRate(RateEvent rateEvent) {
        Integer rate = (Integer) rateEvent.getRating();
        //Add facesmessage
    }

    public void handleCancel() {
        //Add facesmessage
    }

    //getter-setter
}
```

Client Side Callbacks

`onRate` is called when a rating is selected with `value` as the only parameter.

```
<p:rating value="#{ratingBean.rating}" onRate="alert('You rated: ' + value)" />
```

Client Side API

Widget `PrimeFaces.widget.Rating`

Method	Params	Return Type	Description
<code>getRating()</code>		<code>String</code>	Return the current value
<code>setRating(String value)</code>	<code>String</code>	<code>void</code>	Update rating, value with provided one.
<code>disable()</code>		<code>void</code>	Disable component.
<code>enable()</code>		<code>void</code>	Enable component.

Method	Params	Return Type	Description
re et?@	B	void	*(ear the rating,..

Skinning

Rating, resides in a "rating" container which *style* and *styleClass* attribute are available in the structure of the rating.

Style Class	Applies
.ui-rating,	Rating container element.
.ui-rating, .ui-rating .cancel	Cancel icon
.ui-rating, .ui-rating .deactivate	Deactivate state
.ui-rating, .ui-rating .active	Active state

3.89 RemoteCommand

Re "ote*o" " and &rovide a 'ay to e0ecute bac3in, bean "ethod direct(y +ro" /ava cri&t.

Info

9a,	remoteCommand
*o " &onent *(a	org.primefaces.component.remotecommand.RemoteCommand
*o " &onent 9y&e	org.primefaces.component.RemoteCommand
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.RemoteCommandRenderer
Renderer *(a	org.primefaces.component.remotecommand.RemoteCommandRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that "a& to a erver ide UI*o" &onent in tance in a bac3in, bean
action	nu((8 ethod E0&r	A " ethod e0&re ion that'd be &roce ed in the &artia(reCue t cau ed by uia/a0.
action6i tener	nu((8 ethod E0&r	An action(i tener that'd be &roce ed in the &artia(reCue t cau ed by uia/a0.
i " " ediate	+a(e	1oo(ean	1oo(ean va(ue that deter " ine the &ha eldE ' hen true action are &roce ed at a&&(yLreCue tLva(ue E ' hen +a(e at invo3eLa&&(ication &ha e.
na " e	nu((Strin,	<a " e o+ the co " " and
a ync	+a(e	1oo(ean	When et to trueE a/a0 reCue t are not Cueued.
&roce	nu((Strin,	*o " &onent? @ to &roce &artia((y in tead o+ ' ho(e vie ' .
u&date	nu((Strin,	*o " &onent? @ to u&date ' ith a/a0.
on tart	nu((Strin,	Aava cri&t hand(er to e0ecute be+ore a/a0 reCue t i be,in .
onco " &(ete	nu((Strin,	Aava cri&t hand(er to e0ecute ' hen a/a0 reCue t i co " &(eted.
on ucce	nu((Strin,	Aava cri&t hand(er to e0ecute ' hen a/a0 reCue t ucceed .

Name	Default	Type	Description
onerror	null	String	A Java script handler to execute when an error occurs.
,(obal	true	Boolean	Global (all requests) or local (only by action). If true, the global error handler will be triggered.
delay	null	String	If (greater than delay) it (will be delayed until the next request or other request is received. It is not executed if the value of delay is the iterator (true, none) without the quote and no delay is used.)
&artia(Sub " it	false	Boolean	Enable aria-label attribute on the element, to the aria-labelledby attribute and its value is the same as the value of the input element.
autoRun	false	Boolean	When enabled, it is executed on load.
re et>a(ue	false	Boolean	If true (local value or input value) and to be updated within the current request. It is not updated if the value is false or the value is null and no such indication is made.
i, noreAutoUpdate	false	Boolean	If true, it will not be updated for the current request. It is not updated if the value is false or the value is null and no such indication is made.

Getting started with RemoteCommand

Re "ote*o" " and is used by invoking, the code " " and +ro " your /ava cri&t code.

```
<p:remoteCommand name="increment" actionListener="#{counter.increment}"
    out="count" />

<h:outputText id="count" value="#{counter.count}" />
```

```
<script type="text/javascript">
    function customfunction() {
        //your custom code

        increment();           //makes a remote call
    }
</script>
```

Note that whenever you execute your code to "/ava cri&t function", it calls the function defined above. Note that re "ote*o" " and "u t be ne ted in ide a +or".

Passing Parameters

Re "ote co " " and can send dynamic parameters in the form of objects in, 'ay

```
increment([{name:'x', value:10}, {name:'y', value:20}]);
```

3.90 ResetInput

Inlet co " &onent 3ee& their oca(va(ue at tate ' hen va(idation +ai(. Re etIn&ut i u ed to c(ea the cached va(ue +ro " tate o that co " &onent retrieve their va(ue +ro " the bac3in, bean " ode(in tead.

Info



co " &onent at once! &rovide a (i t o+ id or /u t &rovide an ance tor co " &onent (i3e the &ane(in a " &(e above.

Reset Programmatically

Re etIn&ut ta, i the dec(arative 'ay to re et in&ut co " &onent & another 'ay i re ettin, &ro,ra " " atica((y. 9hi i a(o handy i+ in&ut hou(d ,et re et ba ed on a condition. #o((o ' in, a " &(e de " on trate ho ' to u e ReCue t*onteOt to do the re et ' ithin an a/a0behavior (i tener. !ara " eter o+ the re et " ethod can be a in,(e c(ientId or a co((ection o+ c(ientId .

```
<p:inputText value="#{bean.value}">
    <p:ajax event="blur" listener="#{bean.listener}" />
</p:inputText>
```

```
public void listener() {
    RequestContext context = RequestContext.getCurrentInstance();
    context.reset("form:panel");
}
```

Tip

&Na/a0 ha bui(tBn *resetValues* attribute to re et va(ue o+ &roce ed co " &onent durin, a reCue t.

3.91 Resizable

Re i=ab(e co " &onent i u ed to " a3e another AS# co " &onent re i=ab(e.

Info

9a,	resizable
*o "&onent *(a	org.primefaces.component.resizable.Resizable
*o "&onent 9y&e	org.primefaces.component.Resizable
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ResizableRenderer
Renderer *(a	org.primefaces.component.resizable.ResizableRenderer

Attributes

Name	Default	Type	Description
contain "ent	+a(e	1oo(ean	Set re i=ab(e boundarie a the &arent i=e.
,rid	1	Inte,er	Sna& re i=in, to ,rid tructure.
onStart	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen re i=in, be,in .
onRe i=e	nu((Strin,	*(ient ide ca((bac3 to e0ecute durin, re i=in,.
onSto&	nu((Strin,	*(ient ide ca((bac3 to e0ecute a+ter re i=in, end.

Getting started with Resizable

Re i=ab(e i u ed by ettin, *for o&option a the identi+ier o+ the tar, et.*

```
<p:graphicImage id="img" value="campnou.jpg" />
<p:resizable for="img" />
```

Another e0a " &(e i the in&ut +ie(d E i+ u er need " ore &ace +or a te0tareaE " a3e it re i=ab(e by%

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area" />
```

Boundaries

9o &revent over(a&&in, ' ith other e(e" ent on &a,eE boundarie need to be &eci+ied. 9here're - attribute +or thi *minWidth*, *maxWidth*, *minHeight* and *maxHeight*. 9he va(id va(ue +or the e attribute are nu " ber in ter" o+ &i0e(.

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area" minWidth="20" minHeight="40" maxWidth="50" maxHeight="100"/>
```

Handles

Re i=e hand(e to di &(ay are cu to " i=e u in, *handles* attribute ' ith a co " bination o+ neE seE swE seE nwE and nw a the va(ue. De+au(t va(ue i IeE seE seI.

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area" handles="e,w,n,se,sw,ne,nw"/>
```

Visual Feedback

Re i=e he(&er i the e(e" ent u ed to &rovide vi ua(+eedbac3 durin, re i=in,. 1y de+au(t actua(e(e" ent it e(+ i the he(&er and t ' o o&option are avai(ab(e to cu to " i=e the ' ay +eedbac3 i &rovided. Enab(in, *ghost* o&option di &(ay the e(e" ent it e(+ ' ith a (o ' er o&acityE in addition enab(in, *proxy* o&option add a c c(a ca((ed .ui-resizable-proxy ' hich you can override to cu to " i=e.

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area" proxy="true" />
```

```
.ui-resizable-proxy {
    border: 2px dotted #00F;
}
```

Effects

Resizable can be animated using, *animate* option and setting, an *effect* name. Animation needs to be defined in the *for* attribute, *effectDuration* option "slow", "normal" and "fast" are valid values.

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area" animate="true" effect="swing" effectDuration="normal" />
```

#0(0'in, i the (i t o+ avai(ab(e effect name

G ' in,	G ea elnDuart	G ea e7utSine	G ea elnE(a tic	G ea eln1ounce
G ea elnDuard	G ea e7utDuart	G ea elnE0&o	G ea e7utE(a tic	G ea e7ut1ounce
G ea e7utDuard	G ea eln7utDuart	G ea e7utE0&o	G ea eln7utE(a tic	G ea eln7ut1ounce
G ea eln7utDuard	G ea elnDuint	G ea eln7utE0&o	G ea eln1ac3	
G ea eln*ubic	G ea e7utDuint	G ea eln*irc	G ea e7ut1ac3	
G ea e7ut*ubic	G ea eln7utDuint	G ea e7ut*irc	G ea eln7ut1ac3	
G ea eln7ut*ubic	G ea elnSine	G ea eln7ut*irc		

Ajax Behavior Events

Resizable provides default and on(y *resize* event that is called on resize end. In case you have a (i tener defined it ' i((be called by &a in, an *org.primefaces.event.ResizeEvent* interface a a &ara " after.

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area">
    <p:ajax listener="#{resizeBean.handleResize}">
</p:resizable>
```

```
public class ResizeBean {
    public void handleResize(ResizeEvent event) {
        int width = event.getWidth();
        int height = event.getHeight();
    }
}
```

Client Side Callbacks

Re i=ab(e ha three c(ient ide ca((bac3 you can u e to hoo3Bin your /ava cri&t" onStart& onResize and onStop. A((o+ the e ca((bac3 receive t' o &ara" eter that &rovide variou in+or" ation about re i=e event.

```
<h:inputTextarea id="area" value="Resize me if you need more space" />
<p:resizable for="area" onStop="handleStop(event, ui)" />
```

```
function handleStop(event, ui) {
    //ui.helper = helper element as a jQuery object
    //ui.originalPosition = top, left position before resizing
    //ui.originalSize = width, height before resizing
    //ui.position = top, left after resizing
    //ui.size = width height of current size
}
```

Skinning

Style Class	Applies
.ui-resizable	Elements that are resizable
.ui-resizable-handle	Handles
.ui-resizable-handle-edge	Particular handles identified by edge (top, left, bottom, right)
.ui-resizable-corner	Corner handles

3.92 Ring

Rin, i a data di &(ay co " &onent ' ith a circu(ar ani " ation.



Info

9a,	ring
*o " &onent *(a	org.primefaces.component.ring.Ring
*o " &onent 9y&e	org.primefaces.component.Ring
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.RingRenderer
Renderer *(a	org.primefaces.component.ring.RingRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
va(ue	nu((7b/ect	*o((ection to di &(ay.
var	nu((Strin,	<a " e o+ the data iterator.
ty(e	nu((Strin,	In(ine ty(e o+ the container e(e " ent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the container e(e " ent.
ea in,	' in,	Strin,	9y&e o+ ea in, to u e in ani " ation.

Getting started with Ring

A component is required to use the Ring, component.

```
public class RingBean {
    private List<Player> players;

    public RingBean() {
        players = new ArrayList<Player>();

        players.add(new Player("Messi", 10, "messi.jpg", "CF"));
        players.add(new Player("Iniesta", 8, "iniesta.jpg", "CM"));
        players.add(new Player("Villa", 7, "villa.jpg", "CF"));
        players.add(new Player("Xavi", 6, "xavi.jpg", "CM"));
        players.add(new Player("Puyol", 5, "puyol.jpg", "CB"));
    }

    //getter&setters for players
}
```

```
<p:ring value="#{ringBean.players}" var="player">
    <p:graphicImage value="/images/barca/#{player.photo}"/>
</p:ring>
```

Item Selection

A component is required to choose an item from the ring, component.

```
<p:ring value="#{ringBean.players}" var="player">
    <p:column>
        //UI to select an item e.g. commandLink
    </p:column>
</p:ring>
```

Easing

#option, indicating the intention to avail(ab)option or easing, animation.

Skinning

Re ide in a "ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in , i the (i t o+ tructura(ty(e c(a e .

Style Class	Applies
.uiBrin,	8 ain container e(e " ent.
.uiBrin,Bite "	Each ite " in the (i t.

3.93 Row

Ro ' i a he(&er co " &onent +or datab(e.

Info

9a,	row
*o " &onent *(a	org.primefaces.component.row.Row
*o " &onent 9y&e	org.primefaces.component.Row
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	True (value to identify the rendering, or the component) when set to false the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.

Getting Started with Row

See [datab\(e ,rou&in , ection +or " ore in+or " ation about ho ' ro ' i u ed.](#)

3.94 RowEditor

Ro ' Editor i a he(&er co " &onent +or datab(e.

Info

9a,	rowEditor
*o " &onent *(a	org.primefaces.component.roweditor.RowEditor
*o " &onent 9y&e	org.primefaces.component.RowEditor
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.RowEditorRenderer
Renderer *(a	org.primefaces.component.roweditor.RowEditorRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	True (value to identify the rendering, or the component) when set to false the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.

Getting Started with RowEditor

See introduction, section in database documentation or more information about usage.

3.95 RowExpansion

Ro ' E0&an ion i a he(&er co " &onent o+ databab(e u ed to i " &(e " ent e0&andab(e ro ' .

Info

9a,	rowExpansion
*o " &onent *(a	org.primefaces.component.rowexpansion.RowExpansion
*o " &onent 9y&e	org.primefaces.component.RowExpansion
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	True (value to identify the rendering, or the component) when set to false the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component in a backing bean.
style*(a	null	String	Style class for the component.

Getting Started with RowExpansion

See databab(e e0&andab(e ro ' section for more information about how to use this component.

3.96 RowToggler

Ro ' 9o , ,(er i a he(&er co " &onent +or datab(e.

Info

9a,	rowToggler
*o " &onent *(a	org.primefaces.component.rowtoggler.RowToggler
*o " &onent 9y&e	org.primefaces.component.RowToggler
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.RowTogglerRenderer
Renderer *(a	org.primefaces.component.rowtoggler.RowTogglerRenderer

Attributes

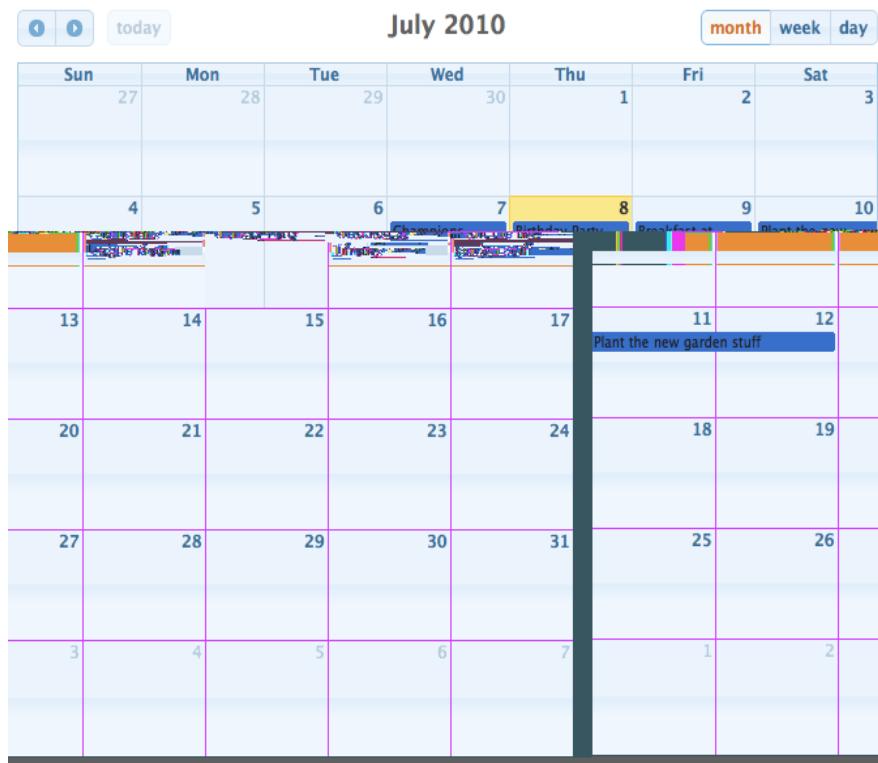
Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	True (value to identify the rendering, or the component) when set to false (the component will not be rendered).
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.

Getting Started with Row

See the section in the documentation for more information about usage.

3.97 Schedule

Schedule provides an interface forendar events to ana, e event .



Info

9a,	schedule
*o " &onent *(a	org.primefaces.component.schedule.Schedule
*o " &onent 9y&e	org.primefaces.component.Schedule
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ScheduleRenderer
Renderer *(a	org.primefaces.component.schedule.ScheduleRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	True (value to identify the rendering of the component) when set to true the component will not be rendered.
binding	null	Object	An expression indicating that it goes to a server-side UI component instance in a backing bean

Name	Default	Type	Description
' id, et>ar	nu()	String,	<a " e o+ the c(ient ide ' id,et.
va(ue	nu()	Boolean	An or ,.&ri " e+ace . " ode(.Schedu(e 8 ode(in tance re&re entin, the bac3ed " ode(
(oca(e	nu()	Boolean	6oca(e +or (oca(i=ationE can be String, or a /ava.uti(.6oca(e in tance
a &ectRatio	nu()	#(oat	Ratio o+ ca(endar ' idth to hei,htE hi,her the va(ue horter the hei,ht i
vie '	" onth	String,	9he vie ' ty&e to u eE &o ib(e va(ue are () " onthE a,endaDayE a,endaWee3E ba icWee3E ba icDay
initia(Date	nu()	Boolean	9he initia(date that i u ed ' hen chedu(e (oad . l+ o " " ittedE the chedu(e tart on the current date
ho ' Wee3end	true	Boolean	S&eci+ie inc(u ion Saturday/Sunday co(u " n in any o+ the vie '
ty(e	nu()	String,	Sty(e o+ the " ain container e(e " ent o+ chedu(e
ty(e*(a	nu()	String,	Sty(e c(a o+ the " ain container e(e " ent o+ chedu(e
dra, ,ab(e	true	Boolean	When trueE event are dra, ,ab(e.
re i=ab(e	true	Boolean	When trueE event are re i=ab(e.
ho ' . eader	true	Boolean	S&eci+ie vi ibi(ity o+ header content.
(e+t . eader9e "&(ate	&revE ne0t today	String,	*ontent o+ (e+t ide o+ header.
center . eader9e "&(ate	title	String,	*ontent o+ center o+ header.
ri, ht . eader9e "&(ate	" onthE a,endaWee3E a,endaDay	String,	*ontent o+ ri, ht ide o+ header.
a((DayS(ot	true	Boolean	Deter " ine i+ a((Bday (ot ' i((be di &(ayed in a,endaWee3 or a,endaDay vie '
(ot 8 inute)0	Integer	Interva(in " inute in an hour to create a (ot.
+ir t . our	4	Integer	#ir t hour to di &(ay in day vie '.
" in9i " e	nu()	String,	8 ini " u " ti " e to di &(ay in a day vie '.
" a09i " e	nu()	String,	8 a0i " u " ti " e to di &(ay in a day vie '.
a0i #or " at	nu()	String,	Deter " ine the ti " eBte0t that ' i((be di &(ayed on the vertica(a0i o+ the a,enda vie ' .

Name	Default	Type	Description
ti " e#or " at	nu((Strin,	Deter " ine the ti " eBteOt that ' i((be di &(ayed on each event.
co(u " n#or " at	nu((Strin,	#or " at +or co(u " n header .
ti " eUone	nu((7b/ect	Strin, or a /ava.uti(.9i " eUone in tance to &eci+y the ti " e=one u ed +or date conver ion.
i ,nore9i " e=one	true	1oo(ean	When &ar in, date E ' hether U9* o++ et hou(d be i,nored ' hi(e &roce in, event data.
too(ti&	+a(e	1oo(ean	Di &(ay de cri&tion o+ event on a too(ti&.

Getting started with Schedule

Schedu(e need to be bac3ed by an *org.primefaces.model.ScheduleModel* in tanceE a chedu(e " ode(con i t o+ *org.primefaces.model.ScheduleEvent* in tance .

```
<p:schedule value="#{scheduleBean.model}" />
```

```
public class ScheduleBean {

    private ScheduleModel model;

    public ScheduleBean() {
        eventModel = new ScheduleModel<ScheduleEvent>();
        eventModel.addEvent(new DefaultScheduleEvent("title", new Date(),
            new Date()));
    }

    public ScheduleModel getModel() {
        return model;
    }
}
```

De+au(tSchedu(eEvent i the de+au(t i " &(e " entation o+ Schedu(eEvent inter+ace. 8 andatory &ro&ertie reCquired to create a ne ' event are the tit(eE start date and end date. 7ther &ro&ertie uch a a((Day ,et en ib(e de+au(t va(ue .

9ab(e be(o ' de ccribe each &ro&erty in detai(.

Property	Description
id	U ed interna((y by !ri " e#ace E auto ,enerated.
tit(e	9it(e o+ the event.
tartDate	Start date o+ ty&e /ava.uti(.Date.

Property	Description
endDate	End date or type /java.util.Date.
addAction	#(a, indicator, event in action day).
type*(action)	> indicates the action to enable "update source" & (ay).
data	Additional data you can set to be rendered by Event.
editable	Whether the event is editable or not.
description	Description to display over or an event.

Ajax Behavior Events

Schedule provides various ajax behavior events to respond after action.

Event	Listener Parameter	Fired
dateSelect	or, #&{event.SlectEvent}	When a date is selected.
eventSelect	or, #&{event.SelectEvent}	When an event is selected.
eventSolve	or, #&{event.SolveEvent}	When an event is solved.
eventResolve	or, #&{event.ResolveEvent}	When an event is resolved.
viewChange	or, #&{event.SelectEvent}	When a view is changed.

Ajax Updates

Schedule has a feature called UI which is generated on the fly by the client side! It is a standard event scheduler that adds and increments a new or previous value. And when you try to update schedule (like when a user enters !) it will notice a UI again, and the UI will be re-generated and re-created. In fact, Schedule provides a client-side method named *update*.

Whenever you call update on the schedule, it will check if there is a scheduled event in the same time range. If there is, it will cancel the previous event and create a new one. This is useful when you want to update a specific event without affecting others.

TimeZone

Time zone is one of the most important features of the schedule component. Set the time zone to UTC so that the schedule takes care of the difference between the client browser time and the event date so that events are displayed at the correct time.

Editable Schedule

Get it to either to do "enable" or "disabled" and call "schedule".

```

<h:form>
    <p:schedule value="#{bean.eventModel}" editable="true" widgetVar="myschedule">
        <p:ajax event="dateSelect" listener="#{bean.onDateSelect}"
            update="eventDetails" oncomplete="eventDialog.show()" />
        <p:ajax event="eventSelect" listener="#{bean.onEventSelect}">
    </p:schedule>

    <p:dialog widgetVar="eventDialog" header="Event Details">
        <h:panelGrid id="eventDetails" columns="2">
            <h:outputLabel for="title" value="Title:" />
            <h:inputText id="title" value="#{bean.event.title}" required="true"/>

            <h:outputLabel for="from" value="From:" />
            <p:inputMask id="from" value="#{bean.event.startDate}" mask="99/99/9999">
                <f:convertDateTime pattern="dd/MM/yyyy" />
            </p:inputMask>

            <h:outputLabel for="to" value="To:" />
            <p:inputMask id="to" value="#{bean.event.endDate}" mask="99/99/9999">
                <f:convertDateTime pattern="dd/MM/yyyy" />
            </p:inputMask>

            <h:outputLabel for="allDay" value="All Day:" />
            <h:selectBooleanCheckbox id="allDay" value="#{bean.event.allDay}" />

            <p:commandButton type="reset" value="Reset" />
            <p:commandButton value="Save" actionListener="#{bean.addEvent}"
                oncomplete="PF('myschedule').update();PF('eventDialog').hide();"/>
        </h:panelGrid>
    </p:dialog>
</h:form>

```

```

public class ScheduleBean {

    private ScheduleModel<ScheduleEvent> model;
    private ScheduleEventImpl event = new DefaultScheduleEvent();

    public ScheduleBean() {
        eventModel = new ScheduleModel<ScheduleEvent>();
    }

    public ScheduleModel<ScheduleEvent> getModel() { return model; }

    public ScheduleEventImpl getEvent() { return event; }
    public void setEvent(ScheduleEventImpl event) { this.event = event; }

    public void addEvent() {
        if(event.getId() == null)
            eventModel.addEvent(event);
        else
            eventModel.updateEvent(event);

        event = new DefaultScheduleEvent(); //reset dialog form
    }

    public void onEventSelect(SelectEvent e) {
        event = (ScheduleEvent) e.getObject();
    }

    public void onDateSelect(SelectEvent e) {
        Date date = (Date) e.getObject();
        event = new DefaultScheduleEvent("", date, date);
    }
}

```

Lazy Loading

Schedule a user-defined event are easily provided in Schedule mode if you have a huge data set or event data (e.g. feature 'ou(d he to i" &rove &eror "ance). In (e.g. oadin, "ode on(y the event that be(on, to the di &(ayed ti " e +ra" e are +etched herea in de+au(t ea, er " ore a((event need to be loaded.

```
<p:schedule value="#{scheduleBean.lazyModel}" />
```

To enable lazy loading, you don't need to provide an instance of `org.primefaces.model.LazyScheduleModel` and implement the `loadEvents` method. `loadEvents` method is called with the boundaries every time it is triggered.

```

public class ScheduleBean {

    private ScheduleModel lazyModel;

    public ScheduleBean() {

        lazyModel = new LazyScheduleModel() {

            @Override
            public void loadEvents(Date start, Date end) {
                //addEvent(...);
                //addEvent(...);
            }
        };
    }

    public ScheduleModel getLazyModel() {
        return lazyModel;
    }
}

```

Customizing Header

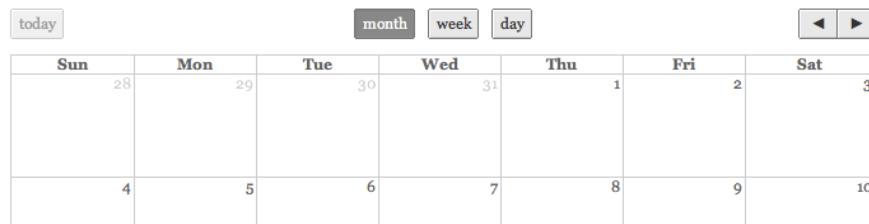
The header controls can be customized based on the date value and options available.

- `today`: current month day action
- `prev`: previous month one month day.
- `next`: next month one month day.
- `year`: previous year one year
- `nextYear`: next year one year
- `today`: current month day.
- `viewName`: change the view based on the view type.

The header controls can be placed at three locations on the header which are defined with `leftHeaderTemplate`, `rightHeaderTemplate` and `centerTemplate` attributes.

```

<p:schedule value="#{scheduleBean.model}"
    leftHeaderTemplate="today"
    rightHeaderTemplate="prev,next"
    centerTemplate="month, agendaWeek, agendaDay"
/>
```

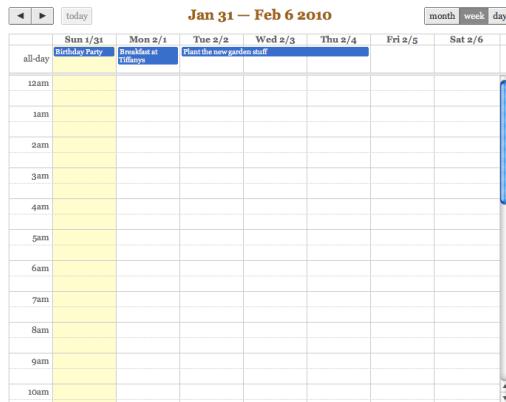


Views

5 different views are supported: the are "ontheweek", "endaweek", "endaday", "basicweek" and "icday".

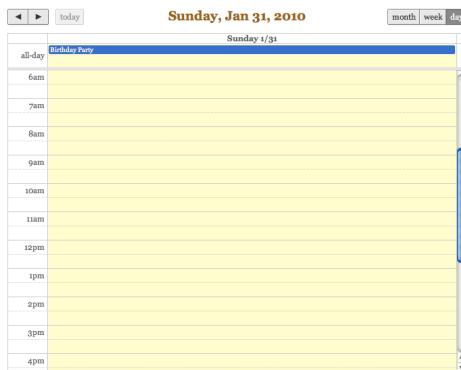
a. endaweek

```
<p:schedule value="#{scheduleBean.model}" view="agendaWeek"/>
```



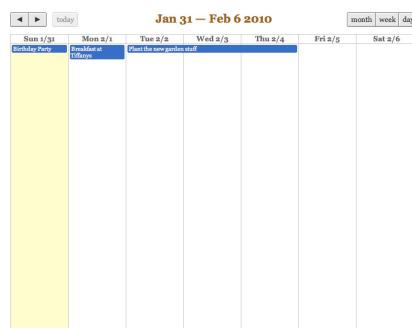
a. endaday

```
<p:schedule value="#{scheduleBean.model}" view="agendaDay"/>
```



b. basicweek

```
<p:schedule value="#{scheduleBean.model}" view="basicWeek"/>
```



ba icDay

```
<p:schedule value="#{scheduleBean.model}" view="basicDay"/>
```



Locale Support

1 y de+au(t (oca(e in+or " ation i retrieved +ro " the vie ' ' (oca(e and can be overridden by the (oca(e attribute. 6oca(e attribute can ta3e a (oca(e 3ey a a Strin, or a /ava.uti(.6oca(e in tance. De+au(t (an, ua, e o+ (abe(are En,(i h and you need to add the nece ary tran (ation to your &a,e " anua(yy a !ri " e#ace doe not inc(ude (an, ua, e tran (ation . !ri " e#ace Wi3i !a, e +or !ri " e#ace 6oca(e i a co" " unity driven &a,e ' here you "ay +ind the tran (ation you need. !(ea e contribute to thi ' i3i ' ith your o ' n tran (ation .

<http://wiki.primefaces.org/display/Components/PrimeFaces+Locales>

9ran (ation i a i" &(e /ava cri&t ob/ectE ' e u., ,e t addin, the code to a /ava cri&t +i(e and inc(ude in your a&&ication. #o((o ' in, i a 9ur3i h ca(endar.

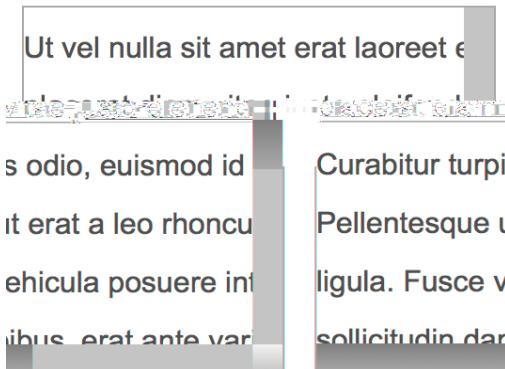
```
<p:schedule value="#{scheduleBean.model}" locale="tr"/>
```

Skinning

Schedule ride in a "ain container 'hich *style* and *styleClass* attribute a&&(y. A 3innin, tye c(a e are .(obaE ee the "ain the "in_ ection +or " ore in+or " ation.

3.98 ScrollPanel

Scro((!ane(i u ed to di &(ay cro((ab(e content ' ith the " e a ' are cro((bar in tead o+ native bro ' er cro((bar .



Info

9a,	scrollPanel
*o " &onent *(a	org.primefaces.component.scrollpanel.ScrollPanel
*o " &onent 9y&e	org.primefaces.component.ScrollPanel
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ScrollPanelRenderer
Renderer *(a	org.primefaces.component.scrollpanel.ScrollPanelRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+iier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a server ide UI *o " &onent in tance in a bac3in, bean
ty(e	nu((Strin ,	In(ine ty(e o+ the container e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the container e(e " ent.
" ode	de+au(t	Strin ,	Scro((bar di &(ay " odeE va(id va(ue are de+au(t and native.

Getting started with ScrollPanel

In order to set the content width and/or height you'd be defined.

```
<p:scrollPanel style="width:250px;height:200px">
    //any content
</p:scrollPanel>
```

Native ScrollBars

If you want to define the scrollbars mode to native, "mode" option to native defines the browser's native scrollbar.

```
<p:scrollPanel style="width:250px;height:200px" mode="native">
    //any content
</p:scrollPanel>
```

Skinning

Scrollers reside in a main container which *style* and *styleClass* attribute applies. A skinning, type class are, (obviously) the main thing in, selection or more in or action. #0(0'in, in the (int of structure(type class K

Style Class	Applies
.ui-scroller	8 main container element.
.ui-scroller-container	7 vertical scroll container.
.ui-scroller-content	* content element.
.ui-scroller-bar	. orientation scroll container.
.ui-scroller-bar-track	vertical scroll container.
.ui-scroller-bar-drag	drag element.
.ui-scroller-bar-drag,	Drag element.

3.99 SelectBooleanButton

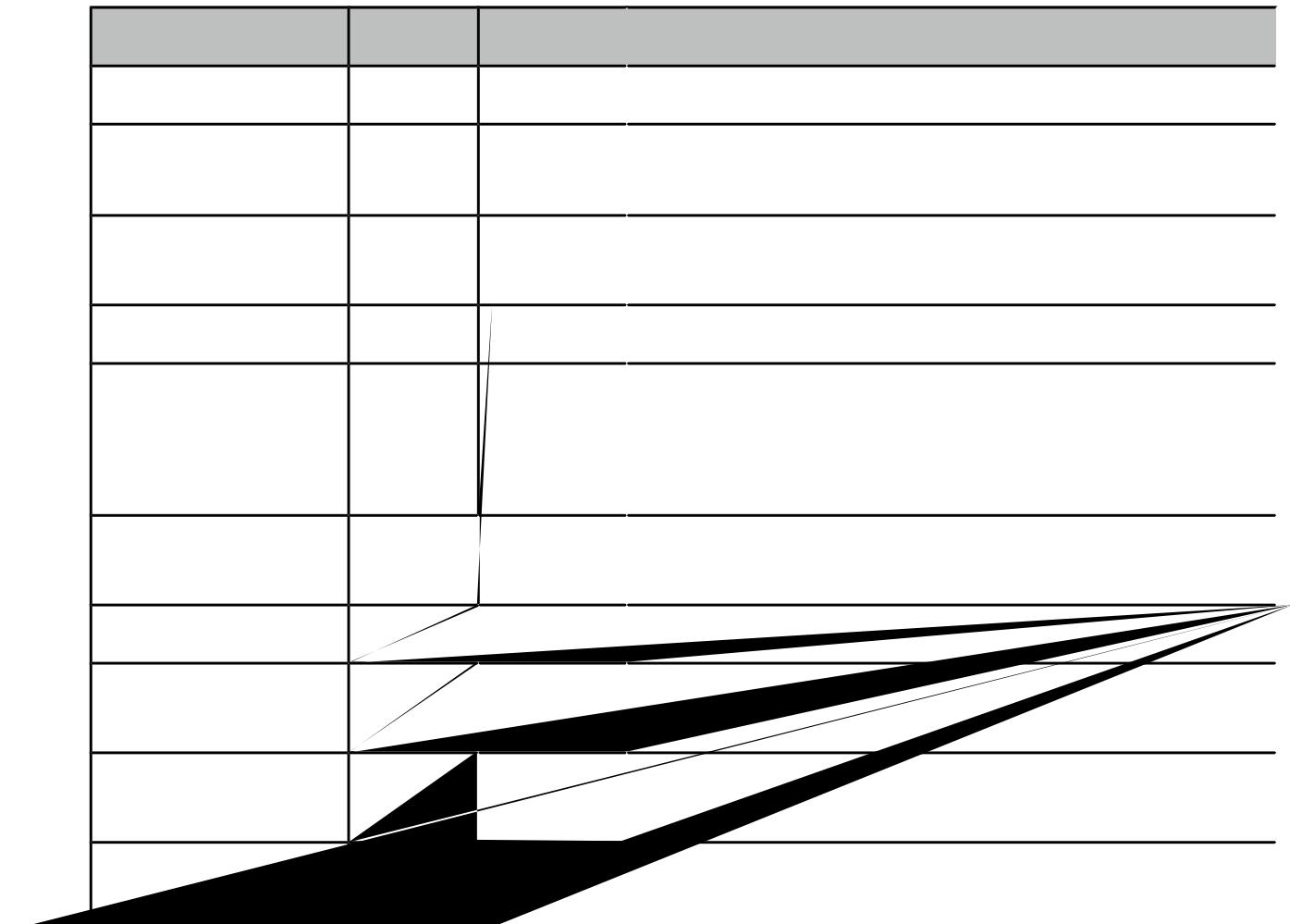
Select one button intended to select a binary decision with a touch button.

I accept terms and conditions: No

Info

9a,	selectBooleanButton
*o "&onent *(a	org.primefaces.component.selectbooleanbutton.SelectBooleanButton
*o "&onent 9y&e	org.primefaces.component.SelectBooleanButton
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectBooleanButtonRenderer
Renderer *(a	org.primefaces.component.selectbooleanbutton.SelectBooleanButtonRenderer

Attributes



Name	Default	Type	Description
converter	null	String	Value to be displayed when conversion failed.
validator	null	String	Value to be displayed when validation failed.
id,etc>ar	null	String	ar".
disabled	false	Boolean	Disables the component.
(abe)	null	String	User-defined tab name.
onchange	null	String	*attribute to execute on value change.
ty(e	null	String	Inline style of the component.
ty(e*(a	null	String	Style class of the container.
on6abe(null	String	6abe(to display when button is selected.
o++6abe(null	String	6abe(to display when button is unselected.
onIcon	null	String	Icon to display when button is selected.
o++Icon	null	String	Icon to display when button is unselected.

Getting started with SelectBooleanButton

SelectBooleanButton uses "input type="checkbox" instead of selectBooleanButton.

```
<p:selectBooleanButton id="value2" value="#{bean.value}" onLabel="Yes"
    offLabel="No" onIcon="ui-icon-check" offIcon="ui-icon-close" />
```

```
public class Bean {
    private boolean value;
    //getter and setter
}
```

Skinning

SelectBooleanButton reuses in a "main container" which *style* and *styleClass* attribute apply. A 3innin, ty(e c(a e are ,obabE ee the "main the" in, ection +or "ore in+or "ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.ui-selectbooleanbutton	main container element.

3.100 SelectBooleanCheckbox

SelectBooleanCheckbox is an extended version of the standard checkbox with the "e" integration.



Info

9a,	selectBooleanCheckbox
*o "&onent *(a	org.primefaces.component.selectbooleancheckbox.SelectBooleanCheckbox
*o "&onent 9y&e	org.primefaces.component.selectbooleancheckbox.SelectBooleanCheckbox

Name	Default	Type	Description
converter	null	String	Value to be displayed when conversion failed.
validator	null	String	Value to be displayed when validation failed.
clientId	null	String	<a href="#" onclick="clientide('id,etc')."
disabled	false	Boolean	Disable the component.
(abe)	null	String	User &reditable value.
onchange	null	String	*action bac3 to execute on value change.
type	null	String	Input type of the component.
type*(a	null	String	Style class of the container.
ite"6abe	null	String	6abe(disabled &(ayed ne0t to chec3bo0.
tabindex	null	String	Specifies tab order for key navigation.

Getting started with SelectBooleanCheckbox

SelectBooleanCheckbox is a standard one.

Client Side API

Widget PrimeFaces.widget.SelectBooleanCheckbox

Method	Params	Return Type	Description
check()	Boolean	void	*check the checkbox.
unchecked()	Boolean	void	Uncheck the checkbox.
toogle()	Boolean	void	Toggle checked state.

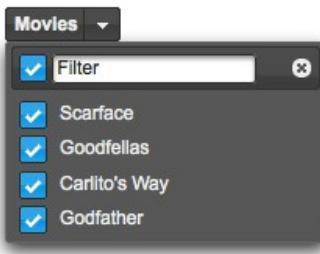
Skinning

SelectBooleanCheckbox resides in a main container which style and styleClass attribute apply. A skinning, type class are, (obad) ee the main the "in, action or " ore in+or " ation. #o((o ' in, it the (it o+ tructura(type class K

Style Class	Applies
.uiBch3bo0	Main container element.
.uiBch3bo0Bbo0	*ontainer of checkbox icon.
.uiBch3bo0Bicon	*checkbox icon.
.uiBch3bo0Bicon	*checkbox (abe).

3.101 SelectCheckboxMenu

Select checkbox menu is a "utility component that displays a dropdown in an overlay.



Info

9a,	selectCheckboxMenu
*o " &onent *(a	org.primefaces.component.selectcheckboxmenu.SelectCheckboxMenu
*o " &onent 9y&e	org.primefaces.component.SelectCheckboxMenu
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectCheckboxMenuRenderer
Renderer *(a	org.primefaces.component.selectcheckboxmenu.SelectCheckboxMenuRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	True (value to display the renderer, or the component) when set to false (the component will not be rendered).
binding	null	Object	An expression that binds to a server-side UI component in a backing bean.
value	null	Object	The value of the component rendered to a field.
converter	null	Converter	An expression or an iterator that defines a converter for the component. When it's an iterator it's resolved to a converter instance. In case it's a static iterator it's used directly to a converter id.
immediate	false	Boolean	When set true, processing validation (which is executed at the same time as the component).
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method expression that refers to a method that performs validation, the input



Getting started with SelectCheckboxMenu

Select*checkboxMenu is a component that extends the standard select*menu and adds the ability to filter items.

```
<p:selectCheckboxMenu value="#{bean.selectedOptions}" label="Movies">
    <f:selectItems value="#{bean.options}" />
</p:selectCheckboxMenu>
```

Filtering

When filtering is enabled, the filter option is rendered at the top header and on the event executed on client side in, *filterMatchMode*. Default mode options that start with and end with a dot are filtered. To require a specific function to do the filtering, .

```
<p:selectCheckboxMenu value="#{bean.selectedOptions}" label="Movies"
    filterMatchMode="custom" filterFunction="customFilter">
    <f:selectItems value="#{bean.options}" />
</p:selectCheckboxMenu>
```

```
function customFilter(itemLabel, filterValue) {
    //return true to accept and false to reject
}
```

Ajax Behavior Events

In addition to the "on" events (like change, etc) the select*checkboxMenu provides a *toggleSelect* event.

Event	Listener Parameter	Fired
to, (select or, .and event.90, , (selectEvent)	When to, (select change event.)	

Skinning

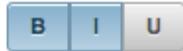
Select*checkboxMenu is rendered in a "main container" which has style and styleClass attributes available. A skinning, typically are, (based on the "main" in, action or "more in" action. #0(0 in, it the (to+ structure typically K

Style Class	Applies
.ui-selectcheckboxmenu" enu	main container element.
.ui-selectcheckboxmenu" enuB(container	label container.
.ui-selectcheckboxmenu" enuB(abe	label.
.ui-selectcheckboxmenu" enuBtri, ,er	Drop-down icon.
.ui-selectcheckboxmenu" enuB&ane(Trigger button.
.ui-selectcheckboxmenu" enuBsite "	Selection (to+ container.

Style Class	Applies
.ui-selectcheckboxenvelope	Each option in the selection.
.ui-ch3bo0	* container of a checkbox.
.ui-ch3bo0bo0	* container of checkbox icon.
.ui-ch3bo0bicon	* checkbox icon.

3.102 SelectManyButton

Select any button in a "button" element contained within, button UI.



Info

9a,	selectManyButton
*o "&onent *(a	org.primefaces.component.selectmanybutton.SelectManyButton
*o "&onent 9y&e	org.primefaces.component.SelectManyButton
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectManyButton
Renderer *(a	org.primefaces.component.selectmanybutton.SelectManyButton

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	True (value to identify the rendering, or the component) when set to false the component will not be rendered.
binding	null	Object	An expression that refers to a server-side IDE UI component instance in a backing bean.
value	null	Object	The value of the component rendered to a string.
converter	null	Converter	An expression or an iterator that defines a converter for the component. When it's an Expression it's resolved to a converter instance. In case it's a static text it's used to refer to a converter id.
immediate	false	Boolean	When true & processes validation (which is executed at all levels) before setting the value and this component.
required	false	Boolean	Specifies a component as required.
validator	null	MethodExpression	A method expression that refers to a method validation, the input
valueChangeListener	null	MethodExpression	A method expression that refers to a method for handling a valuechange event
requiredMessage	null	String	The message to be displayed when required is violated.

Name	Default	Type	Description
converter	null	String	Value to be displayed when conversion failed.
validator	null	String	Value to be displayed when validation failed.
clientId	null	String	The client id of the component.
disabled	false	Boolean	Disables the component.
(abegin)	null	String	User-defined tab name.
onchange	null	String	JavaScript to execute on value change.
type	null	String	Input type of the component.
typeAhead	null	String	Style class of the container.

Getting started with SelectManyButton

Select any button using a "select" element with a "button" attribute. A "ui-select" style class applies to the container element.

Skinning

Select any button resides in a "ui-select" container which has style and styleClass attributes. A "ui-select" style class applies to the container element.

Style Class	Applies
.ui-select "anybutton	Container element.

3.103 SelectManyCheckbox

SelectManyCheckbox is an extended version of the standard SelectManyCheckbox with the "selection" feature.



Info

9a,	selectManyCheckbox
*o " &onent *(a	org.primefaces.component.selectmanycheckbox.SelectManyCheckbox
*o " &onent 9y&e	org.primefaces.component.SelectManyCheckbox
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectManyCheckboxRenderer
Renderer *(a	org.primefaces.component.selectmanycheckbox.SelectManyCheckboxRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression defining the binding to a server-side UI component instance in a bean.
value	null	Object	The value of the component rendered to a string.
converter	null	Converter	An expression or an iterator that defines a converter for the component. When it's an EL expression, it's resolved to a converter instance. In case it's a static object, it's used directly.
immediate	false	Boolean	When set to true, validation (optional) is executed at any point before the component's value is set.
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method expression that refers to a method for validation, the input.
valueChangeListener	null	Method	A method expression that refers to a method for handling a valuechange event.
requiredMessage	null	String	The message to be displayed when required is violated.

Name	Default	Type	Description
			va(idation +ai(.
converter8e a,e	nu((Strin ,	8e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator8e a,e	nu((Strin ,	8e a,e to be di &(ayed ' hen va(idation +ie(d .
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
di ab(ed	+a(e	1oo(ean	Di ab(e the co " &onent.
(abe(nu((Strin ,	U er &re entab(e na " e.
(ayout	(ineDirection	Strin ,	6ayout o+ the chec3bo0e E va(id va(ue are lineDirectionE pageDirection and grid.
co(u " n	0	Inte ,er	<u " ber o+ co(u " n in ,rid (ayout.
onchan,e	nu((Strin ,	*a((bac3 to e0ecute on va(ue chan,e.
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the container.

Getting started with SelectManyCheckbox

Select8 any*hec3bo0 u a,e i a" e a the standard one.

Skinning

Select8 any*hec3bo0 re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB e(ect " anychec3bo0	8 ain container e(e " ent.
.uiBch3bo0	*ontainer o+ a chec3bo0.
.uiBch3bo0Bbo0	*ontainer o+ chec3bo0 icon.
.uiBch3bo0Bicon	*hec3bo0 icon.

3.104 SelectManyMenu

Selects any number in an extended version of the standard SelectManyMenu.



Info

9a,	selectManyMenu
*o " &onent *(a	org.primefaces.component.selectmanymenu.SelectManyMenu
*o " &onent 9y&e	org.primefaces.component.SelectManyMenu
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectManyMenuRenderer
Renderer *(a	org.primefaces.component.selectmanymenu.SelectManyMenuRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Allows you to specify whether the component is rendered or not. If set to false, the component will not be rendered.
binding	null	Object	An expression that binds the component to a server-side UI component in a backing bean.
value	null	Object	The value of the component is set to the component's value.
converter	null	Converter	An expression or an iterator that defines a converter for the component. When it's an Iterator, it's resolved to a converter in advance. In case it's a static iterator, it's used directly to a converter id.
immediate	false	Boolean	When set to true, validation (optional) is executed at the same time as the value change and this component.
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method expression that refers to a method that performs validation, the input
valueChangeListener	null	Method	A method expression that refers to a method that performs validation, the output

Name	Default	Type	Description
			hand(in, a va(uechan,eevent
reCquired	nu((Strin,	8 e a,e to be di &(ayed ' hen reCquired +ie(d va(idation +ai(.
converter	nu((Strin,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator	nu((Strin,	8 e a,e to be di &(ayed ' hen va(idation +ie(d .
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
di ab(ed	+a(e	1oo(ean	Di ab(e the co " &onent.
(abe(nu((Strin,	U er &re entab(e na " e.
onchan,e	nu((Strin,	*a((bac3 to e0ecute on va(ue chan ,e.
onc(ic3	nu((Strin,	*a((bac3 +or c(ic3 event.
ondb(c(ic3	nu((Strin,	*a((bac3 +or db(c(ic3 event.
ty(e	nu((Strin,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the container.
tabinde0	nu((Strin,	!o ition o+ the in&ut e(e " ent in the tabbin , order.
var	nu((Strin,	<a " e o+ iterator to be u ed in cu to " content di &(ay.
ho ' *hec3bo0	+a(e	1oo(ean	When trueE a chec3bo0 i di &(ayed ne0t to each ite " .
+i(ter	+a(e	1oo(ean	Di &(ay an in&ut +i(ter +or the (i t.
+i(ter 8 atch 8 ode	nu((Strin,	8 atch " ode +or +i(terin,E va(id va(ue are tart With ?de+au(t@E contain E end With and cu to " .
+i(ter#unction	nu((Strin,	*(ient ide +unction to u e in cu to " +i(ter 8 atch 8 ode.
ca eSen itive	+a(e	1oo(ean	De+ine i+ +i(terin , ' ou(d be ca e en itive.
cro((. ei ,ht	nu((Inte ,er	De+ine the hei ,ht o+ the cro((ab(e area

Getting started with SelectManyMenu

Se(ect 8 any 8 enu u a,e i a" e a the standard one.

Custom Content

*u to " content can be di &(ayed +or each ite " u in, co(u " n co " &onent .

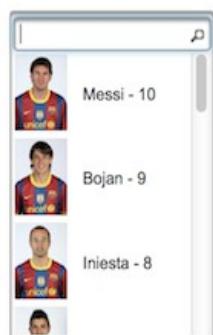
```
<p:selectManyMenu value="#{bean.selectedPlayers}" converter="player" var="p">
    <f:selectItems value="#{bean.players}" var="player"
        itemLabel="#{player.name}" itemValue="#{player}" />
    <p:column>
        <p:graphicImage value="/images/barca/#{p.photo}" width="40"/>
    </p:column>
    <p:column>
        #{p.name} - #{p.number}
    </p:column>
</p:selectManyMenu>
```



Filtering

#i(terin, i enab(ed by ettin, +i(ter attribute to true. 9here are +our +i(ter "ode K *startsWith*
contains endsWith and custom. In cu to " "ode filterFunction " u t be de+ined a the na " e o+ the
/ava cri&t +unction that ta3e the ite " va(ue and +i(ter a &ara " eter to return a boo(ean to acce&t or
re ect a va(ue. 9o add a +i(ter to &reviou e0a "&(eK

```
<p:selectOneMenu value="#{menuBean.selectedPlayer}" converter="player" var="p"
    filter="true" filterMatchMode="contains">
    ...
</p:selectOneMenu>
```



Checkbox

Select 8 any 8 enu ha bui(tBin u&&ort +or chec3bo0 ba ed "u(ti&(e e(ectionE ' hen enab(ed by
showCheckbox o&tionE chec3bo0e are di &(ayed ne0t to each co(u " n.

Skinning

Select any element in a container that `style` and `styleClass` attribute already. A skinning, type are, (otherwise the "ain the" in, ection or "ore in+or" ation. #o(o 'in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.ui-select-any-enumeration	8 ain container e(e " ent.
.ui-select-item-highlight	Each ite" in (i t.

3.105 SelectOneButton

SelectOneButton is an input component to do a single select.



Info

9a,	selectOneButton
*o " &onent *(a	org.primefaces.component.selectonebutton.SelectOneButton
*o " &onent 9y&e	org.primefaces.component.SelectOneButton
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectOneButtonRenderer
Renderer *(a	org.primefaces.component.selectonebutton.SelectOneButtonRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression or ion that binds to a server-side UI component instance in a backing bean.
value	null	Object	The value of the component rendered to a string.
converter	null	Converter	An expression or iterator that defines a converter for the component. When it's an EL expression, it's resolved to a converter instance. In case it's a static method, it's used to refer to a converter id.
immediate	false	Boolean	When set to true, validation (optional) is executed at any point before the component's value is set.
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method expression that refers to a method for validation, the input.
valueChangeListener	null	Method	A method expression that refers to a method for handling value change events.
requiredMessage	null	String	The message to be displayed when required is violated.

Name	Default	Type	Description
			va(idation +ai(.
converter8e a,e	nu((Strin,	8e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator8e a,e	nu((Strin,	8e a,e to be di &(ayed ' hen va(idation +ie(d .
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
di ab(ed	+a(e	1oo(ean	Di ab(e the co " &onent.
(abe(nu((Strin,	U er &re entab(e na " e.
onchan,e	nu((Strin,	*a((bac3 to e0ecute on va(ue chan,e.
ty(e	nu((Strin,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the container.

Getting started with SelectOneButton

SelectOneButton u a,e i a" e a ectOneRadio co " &onent button /u t re&(ace the radio .

Skinning

SelectOneButton re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB e(ectonebutton	8 ain container e(e " ent.

3.106 SelectOneListbox

SelectOneListbox is an extended version of the standard SelectListbox component.



Info

9a,	selectOneListbox
*o " &onent *(a	org.primefaces.component.selectonelistbox.SelectOneListbox
*o " &onent 9y&e	org.primefaces.component.SelectOneListbox
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectOneListboxRenderer
Renderer *(a	org.primefaces.component.selectonelistbox.SelectOneListboxRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression or function that binds the component to a server-side UI component in a backing bean.
value	null	Object	The value of the component rendered to a string.
converter	null	Converter	An expression or function that defines a converter for the component. When it's an EL expression, it's resolved to a converter in the context. In case it's a static method, it's used to resolve to a converter id.
immediate	false	Boolean	When set to true, validation (optional) is executed at any time before the component's value is set.
required	false	Boolean	A boolean attribute indicating whether the component is required.
validator	null	Method	A method expression that refers to a method for validation, the input

Name	Default	Type	Description
valueChangeListener	null	String	A " method expression that refers to a " method for hand(in, a valuechange event)
requiredValidator	null	String	Validator to be displayed when required validation failed.
converter	null	String	Validator to be displayed when conversion failed.
validator	null	String	Validator to be displayed when validation failed.
id,etc>ar	null	String	<a " element of the client-side id attribute.
disabled	false	boolean	Disabled the component.
(abeginning)	null	String	User-defined tab(abeginning name).
onchangetime	null	String	*action(listener) to execute on value change.
onclick	null	String	*action(listener) or click event.
ondblclick	null	String	*action(listener) or double click event.
type	null	String	Input type of the component.
type*(a)	null	String	Style class of the container.
tabindex	null	String	TabIndex of the input element in the tabbing order.
value	null	String	<a " iterator to be used in context content display.
var	null	String	<a " iterator to be used in context content display.
+iterator	false	boolean	Display an inner iterator for the item.
+iterator#atch#ode	null	String	atch "#ode or +iterator, start with value are start With ?de+au(t@E contain End With and context".
+iterator#unction	null	String	*(client-side function to use in context "+iterator#atch#ode).
caseSensitive	false	boolean	Defines if +iterator, 'oud be case sensitive.
croff(.height)	null	Integer	Defines the height of the dropdown area.

Getting started with SelectOneListbox

SelectOneListbox is a "element the standard one.

Custom Content

*Context content can be displayed for each item in, conditionally.

```
<p:selectOneListbox value="#{bean.player}" converter="player" var="p">
    <f:selectItems value="#{bean.players}" var="player"
        itemLabel="#{player.name}" itemValue="#{player}" />

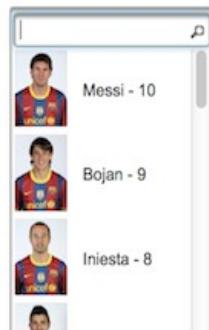
    <p:column>
        <p:graphicImage value="/images/barca/#{p.photo}" width="40"/>
    </p:column>
    <p:column>
        #{p.name} - #{p.number}
    </p:column>
</p:selectOneListbox>
```



Filtering

#i(terin, i enab(ed by ettin, +i(ter attribute to true. 9here are +our +i(ter "ode K *startsWith*
contains endsWith and custom. In cu to " "ode filterFunction " u t be de+ined a the na " e o+ the
/ava cri&t +unction that ta3e the ite " va(ue and +i(ter a &ara " eter to return a boo(ean to acce&t or
re/ect a va(ue. 9o add a +i(ter to &reviou e0a "&(ek

```
<p:selectOneMenu value="#{menuBean.selectedPlayer}" converter="player" var="p"
    filter="true" filterMatchMode="contains">
    ...
</p:selectOneMenu>
```



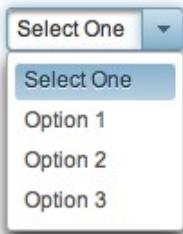
Skinning

Se(ect7ne6i tbo0 re ide in a "ain container ' hich style and styleClass attribute a&&y. A
3innin, ty(e c(a e are ,oba(ee the "ain the " in, ection +or " ore in+or " ation. #o((o ' in, i
the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.ui-selectone(tbo0	8 ain container e(e " ent.
.ui-selectect(i tbo0Bite "	Each ite " in (i t.

3.107 SelectOneMenu

SelectOneMenu is an extended version of the standard SelectOneMenu.



Info

9a,	selectOneMenu
*o " &onent *(a	org.primefaces.component.selectonemenu.SelectOneMenu
*o " &onent 9y&e	org.primefaces.component.SelectOneMenu
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectOneMenuRenderer
Renderer *(a	org.primefaces.component.selectonemenu.SelectOneMenuRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	True (value to display) the renderer, or the component when set to false (the component will not be rendered).
binding	null	Object	An expression that refers to a server-side UIComponent instance in a backing bean.
value	null	Object	The value of the component.
converter	null	Converter	An expression or an iterator that defines a converter for the component. When it's an EL expression it's resolved to a converter instance. In case it's a static iterator it's converted to a converter id.
immediate	false	Boolean	When set true, processing validation (if any) is executed at the same time as the value of this component.
required	false	Boolean	Specifies if the component is required.
validator	null	Method	A method expression that refers to a method that performs validation, the input

Name	Default	Type	Description
valueChangeListener	null	Method	A method called after a value change. It receives the previous value and the new value.
required	null	String	Specifies if the field is required. If true, the field must be filled before the form can be submitted.
converter	null	String	Specifies the converter to be used for this field. It must implement the Converter interface.
validator	null	String	Specifies the validator to be used for this field. It must implement the Validator interface.
validatorMessage	-00	Integer	Duration of the validation message in milliseconds.
disabled	false	Boolean	Specifies if the field is disabled.
(abefore)	null	String	Used for event delegation. It specifies the name of the event to be triggered on the parent component.
onchange	null	String	Specifies the client-side JavaScript code to be executed when the value changes.
on3seyu&	null	String	Specifies the client-side JavaScript code to be executed when the value changes.
on3eyodo'n	null	String	Specifies the client-side JavaScript code to be executed when the value changes.
type	null	String	Specifies the type of the input field, such as "text", "password", or "checkbox".
typeahead	null	String	Specifies the type of the input field, such as "text", "password", or "checkbox".
var	null	String	Specifies the variable name for the iterator.
height	auto	Integer	Specifies the height of the input field.
tabindex	null	String	Specifies the tab index of the input field.
editable	false	Boolean	Specifies if the field is editable.
+item	false	Boolean	Render an item.
+item#atch#ode	startWith	String	Atch "ode" or "+item" value are startWith contain End With and cu to".
+item#unction	null	String	Specifies the function to use in cu to" +item,..
caSeSensitive	false	Boolean	Defines if +item, 'oud be case sensitive.
"a0(en,th	null	Integer	<u "ber o "a0i "u" character a((o 'ed in editable element 8 enum.
and9o	null	String	And the over(ay to the element defined by search e0&re ion. Default to document "ent body.

Getting started with SelectOneMenu

1. Create SelectOneMenu using the standard one.

Custom Content

Select one menu can display content in overlay pane by using, containing "n content and the var option to refer to each item".

```
public class MenuBean {
    private List<Player> players;
    private Player selectedPlayer;

    public OrderListBean() {
        players = new ArrayList<Player>();

        players.add(new Player("Messi", 10, "messi.jpg"));
        //more players
    }

    //getters and setters
}
```

```
<p:selectOneMenu value="#{menuBean.selectedPlayer}" converter="player" var="p">
    <f:selectItem itemLabel="Select One" itemValue="" />
    <f:selectItems value="#{menuBean.players}" var="player"
        itemLabel="#{player.name}" itemValue="#{player}"/>
    <p:column>
        <p:graphicImage value="/images/barca/#{p.photo}" width="40" height="50"/>
    </p:column>

    <p:column>
        #{p.name} - #{p.number}
    </p:column>
</p:selectOneMenu>
```



Effects

An animation is executed to show and hide the overlay pane. Effect mode and options are available (or effect attribute) by binding bounce, click, drop, open, close, hide, height, and none.

Editable

`<selectOneMenu>` provides a UI to either choose one of the predefined option or enter a new value. Set `editable` to true to use this feature.



Filtering

When filtering, it enables filtering on an input field rendered at the header and on keyup event filtering, it executes on client side in `filterMatchMode`. Default mode of filtering is exact match. To contain ends with and contains ". * up to" mode requires a JavaScript function to do the filtering..

```
<p:selectOneMenu value="#{bean.selectedOptions}"
    filterMatchMode="custom" filterFunction="customFilter">
    <f:selectItems value="#{bean.options}" />
</p:selectOneMenu>
```

```
function customFilter(itemLabel, filterValue) {
    //return true to accept and false to reject
}
```



Client Side API

Widget `PrimeFaces.widget.SelectOneMenu`

Method	Params	Return Type	Description
show?@	B	void	Show over(ay " enu.
hide?@	B	void	.ide over(ay " enu.
b(ur)?@	B	void	Invo3e b(ur event.
+ocu ?@	B	void	Invo3e +ocu event.
enab(e)?@	B	void	Enab(e co " &onent.
di ab(e)?@	B	void	Di ab(ed co " &onent.
e(ect>a(ue)?@	va(ue) ite " >a(ue	void	Se(ect ,iven va(ue.
,etSe(ected>a(ue)?@	B	7b/ect	Return va(ue o+ e(ected ite " .
,etSe(ected6abe(?@		Strin,	Return (abe(o+ e(ected ite " .

Skinning

Select 7 ne 8 enu re ide in a container e(e " ent that *style* and *styleClass* attribute a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB e(ectone " enu	8 ain container.
.uiB e(ectone " enuB(abe(6abe(o+ the co " &onent.
.uiB e(ectone " enuBtri , ,er	*ontainer o+ dro&do ' n icon.
.uiB e(ectone " enuBite "	Ite " (i t.
.uiB e(ectone " enuBite "	Each ite " in the (i t.

3.108 SelectOneRadio

SelectOneRadio is an extended version of the standard SelectOneRadio with the "selection" feature.



Info

9a,	selectOneRadio
*o " &onent *(a	org.primefaces.component.selectoneradio.SelectOneRadio
*o " &onent 9y&e	org.primefaces.component.SelectOneRadio
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SelectOneRadioRenderer
Renderer *(a	org.primefaces.component.selectoneradio.SelectOneRadioRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	Unique identifier of the component.
rendered	true	1oo(ean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin,	nu((7b/ect	An expression that adds to a server-side UI component in a backing bean.
va(ue	nu((7b/ect	Value of the component rendered to a field.
converter	nu((*onverter()	An expression or an iterator that defines a converter for the component. When it's an E6 expression, it's resolved to a converter in context. In case it's a static text, it's used as a converter id.
i " " ediate	0	1oo(ean	When set to true, validation (optional) is executed at any request time and the component.
reCquired	0	1oo(ean	Specifies the component as required.
va(idator	nu((8 ethodE0&r	A method expression that refers to a method validation, the input
va(ue*han,e6i tener	nu((8 ethodE0&r	A method expression that refers to a method for handling, a valuechange event
reCquired 8 e a,e	nu((Strin,	Specifies a field to be displayed when required instead of validation.

Name	Default	Type	Description
converter	null	String	Specifies a converter to be used when conversion between the component value and the database value is required.
validator	null	String	Specifies a validator to be used when validation of the component value is required.
clientId	null	String	Specifies the client-side id of the component.
disabled	false	Boolean	Disables the component.
label	null	String	Used for generating the label for the component.
layout	(none)	String	Specifies the layout of the radiobutton. Valid values are lineDirection, pageDirection, custom and grid.
order	0	Integer	Specifies the order of the radio button in the layout.
onchange	null	String	Specifies a JavaScript function to execute on value change.
type	null	String	Specifies the type of the radio button.
typeGroup	null	String	Specifies the style class of the container.
tabIndex	-1	Integer	Specifies the tab order of the component in tab navigation.
labelSeparator	" "	String	Specifies the label separator for the radio button.

Getting started with SelectOneRadio

SelectOneRadio uses a standard one.

Custom Layout

Standard SelectOneRadio component only supports horizontal and vertical rendering, or the radio button with a strict tab order. To face SelectOneRadio on the other hand & provide a flexible layout option so that radio button can be located anywhere on the page. This is achieved by setting layout option to "custom" and with tandem one radio button component. Note that in custom mode SelectOneRadio does not render any output.

```
<p:selectOneRadio id="customRadio" value="#{formBean.option}" layout="custom">
    <f:selectItem itemLabel="Option 1" itemValue="1" />
    <f:selectItem itemLabel="Option 2" itemValue="2" />
    <f:selectItem itemLabel="Option 3" itemValue="3" />
</p:selectOneRadio>
```

```
<h:panelGrid columns="3">
    <p:radioButton id="opt1" for="customRadio" itemIndex="0"/>
    <h:outputLabel for="opt1" value="Option 1" />
    <p:spinner />

    <p:radioButton id="opt2" for="customRadio" itemIndex="1"/>
    <h:outputLabel for="opt2" value="Option 2" />
    <p:inputText />

    <p:radioButton id="opt3" for="customRadio" itemIndex="2"/>
    <h:outputLabel for="opt3" value="Option 3" />
    <p:calendar />
</h:panelGrid>
```

Radio button' +or attribute 'hou(d re+er to a 'e(ect7neRadio co " &onent and ite " Inde0 &oint to the inde0 o+ the 'e(ectlte ". When u in, cu to " (ayout o&tionE 'e(ect7neRadio co " &onent hou(d be &(aced above any radio button that &ooint to the 'e(ect7neRadio.

Skinning

'e(ect7neRadio re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB e(ectoneradio	8 ain container e(e " ent.
.uiBradiobutton	*ontainer o+ a radio button.
.uiBradiobuttonBboO	*ontainer o+ radio button icon.
.uiBradiobuttonBicon	Radio button icon.

3.109 Separator

Separator di &(ay a hori=onta(ine to e&arate content.

Info

9a,	separator
*o " &onent *(a	org.primefaces.component.separator.Separator
*o " &onent 9y&e	org.primefaces.component.Separator
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.Separator
Renderer *(a	org.primefaces.component.separator.Separator

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Logical value to specify if the component should be rendered. If false, it will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
title	null	String	Alternative text for the separator.
type	null	String	Style class of the separator.
type*(a	null	String	Style class of the separator.

Getting started with Separator

In it i " &(e t +or " E e&arator i u ed a K

```
//content
<p:separator />
//content
```

Dimensions

Separator render a *<hr />* ta, ' hich ty(e and ty(e*(a o&tion a&&(y.

```
<p:separator style="width:500px;height:20px" />
```

Special Separators

Separator can be used in side other components such as menu when unsorted.

```
<p:menu>
    //submenu or menuitem
    <p:separator />
    //submenu or menuitem
</p:menu>
```

Skinning

A mentioned in diagram section, ty(e and ty(e*(a option can be used to style the separator. #0((o in, in the (i to+ structure(ty(e c(a e K

Class	Applies
.uiB separator	Separator element

A skinning, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or " ore in+or " ation.

3.110 SlideMenu

Slide menu is used to display nested menus with transition animation.



Info

9a,	slideMenu
*o "&onent *(a	org.primefaces.component.slidemenu.SlideMenu
*o "&onent 9y&e	org.primefaces.component.SlideMenu
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SlideMenuRenderer
Renderer *(a	org.primefaces.component.slidemenu.SlideMenuRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	Unique identifier of the component.
rendered	true	booleean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin ,	nu((7b/ect	An event delegation that adds to a server-side UI component interaction in a backing bean.
' id, et>ar	nu((Strin ,	<a href="#" onclientclick="id, et.
" ode(nu((8 enu 8 ode(8 enum 8 ode(interaction for, r, ra " atic " enu.
ty(e	nu((Strin ,	Inline style of the component.
ty(e *(a	nu((Strin ,	Style class of the component.
bac36abe(1 ac3	Strin ,	9event for bac3(in3.
tri , ,er	nu((Strin ,	Id of the component whose tri , ,erEvent is triggered by the dynamic action.
" y	nu((Strin ,	Marker of " enum to add with tri , ,er event.

Name	Default	Type	Description
at	null	String,	* orner o+ tri,,er to a(i,n ' ith " enu e(e" ent.
over(ay	+a(e	1oo(ean	Define &o itionin,E ' hen enab(ed " enu i di &(ayed ' ith ab o(ute &o ition re(ative to the tri,,er. De+au(t i +a(eE " eanin,, tatic &o itionin,.
tri,,erEvent	c(ic3	String,	Event na" e o+ tri,,er that ' i((ho ' the dyna" ic &o itioned " enu.

Getting started with the SlideMenu

S(ide 8 enu con i t o+ ub "enu and "enuite" E ub "enu can be ne ted and each ne ted ub "enu ' i((be di &(ayed ' ith a (ide ani "ation.

```
<p:slideMenu>
    <p:submenu label="Ajax Menuitems" icon="ui-icon-refresh">
        <p:menuItem value="Save" actionListener="#{buttonBean.save}"
            update="messages" icon="ui-icon-disk" />
        <p:menuItem value="Update" actionListener="#{buttonBean.update}"
            update="messages" icon="ui-icon-arrowrefresh-1-w" />
    </p:submenu>

    <p:submenu label="Non-Ajax MenuItem" icon="ui-icon-newwin">
        <p:menuItem value="Delete" actionListener="#{buttonBean.delete}"
            update="messages" ajax="false" icon="ui-icon-close"/>
    </p:submenu>

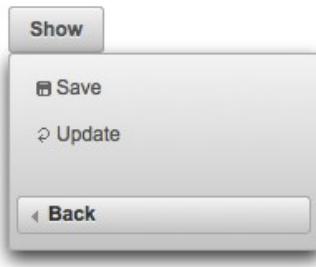
    <p:separator />

    <p:submenu label="Navigations" icon="ui-icon-extlink">
        <p:submenu label="Prime Links">
            <p:menuItem value="Prime" url="http://www.prime.com.tr" />
            <p:menuItem value="PrimeFaces" url="http://www.primefaces.org" />
        </p:submenu>
        <p:menuItem value="Mobile" url="/mobile" />
    </p:submenu>
</p:slideMenu>
```

Overlay

S(ide 8 enu can be &o itioned re(ative to a tri,,er co " &onentE +o((o ' in, a" &(e attache a (ide 8 enu to the button o that ' henever the button i c(ic3ed " enu ' i((be di &(ayed in an over(ay it e(+.

```
<p:commandButton type="button" value="Show" id="btn" />  
  
<p:slideMenu trigger="btn" my="left top" at="left bottom">  
    //content  
</p:slideMenu>
```



Client Side API

Wid, etn `PrimeFaces.widget.SlideMenu`

Method	Params	Return Type	Description
<code>show()</code>	B	void	Show over(ay " enu.
<code>hide()</code>	B	void	. ide over(ay " enu.
<code>animate(n)</code>	B	void	A(i, n over(ay " enu ' ith tri, ,er.

Skinning

S(ide 8 enu re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the
(i t o+ tructura(ty(e c(a e K

Style Class	Applies
<code>.ui-slide(ide " enu .ui-slide(ide " enu</code>	*ontainer e(e " ent o+ " enu.
<code>.ui-slide(ide " enuB ' ra&&er</code>	Wra&&er e(e " ent +or content.
<code>.ui-slide(ide " enuBcontent</code>	*ontent container.
<code>.ui-slide(ide " enuBbac3 ' ard</code>	1ac3 navi ,ator.
<code>.ui-slide(ide " enuB(i t</code>	6i t container
<code>.ui-slide(ide " enuB" enuite "</code>	Each " enu ite "
<code>.ui-slide(ide " enuB" enuite " B(in3</code>	Anchor e(e " ent in a (in3 ite "
<code>.ui-slide(ide " enuB" enuite " Bte0t</code>	9e0t e(e " ent in an ite "

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.111 Slider

Slider is used to provide input with various controls to "iate orientation (like orientation) and binding.



Info

9a,	slider
*o " &onent *(a	org.primefaces.component.slider.Slider
*o " &onent 9y&e	org.primefaces.component.Slider
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SliderRenderer
Renderer *(a	org.primefaces.component.slider.SliderRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify if the component is rendered. If set to false, the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
for	null	String	Id of the input field that the slider will be used for.
disabled	null	String	Id of the component to disable the slider value.
"in>a(ue	0	Integer	Initial value of the slider.
"a0>a(ue	100	Integer	Maximum value of the slider.
type	null	String	Inline type of the container element.
type*(a	null	String	Style class of the container element.
animate	true	Boolean	Boolean value to enable/disable the animation of the slider when it is moved.
type	horizontal	String	Set the type of the slider. It can be vertical or horizontal.
step	1	Integer	Number of increments that the slider moves in.
disabled	0	Boolean	Disable or enable the slider.
onSlideStart	null	String	Script code to execute when the slider begins.

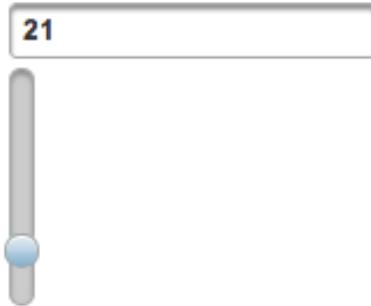
DH ygN

Name	Default	Type	Description
onS(ide	null	String	*(isOnSide,

Vertical Slider

By default the orientation is horizontal, but it can be set to vertical by setting the `type` attribute.

```
<h:inputText id="number" value="#{sliderController.number}" />
<p:slider for="number" type="vertical" minValue="0" maxValue="200"/>
```



Step

Step defines the interval between each point during the animation. By default the value is one and it is controlled via the `step` option.

```
<h:inputText id="number" value="#{sliderBean.number}" />
<p:slider for="number" step="10" />
```

Animation

Setting the `animate` attribute to true turns on the `animate` option to enable animation.

Boundaries

The minimum and maximum boundaries for the slider are defined using the `minValue` and `maxValue` attributes. Both can be between -100 and 100.

```
<h:inputText id="number" value="#{sliderBean.number}" />
<p:slider for="number" minValue="-100" maxValue="100"/>
```

Range Slider

Selecting a range with a value is supported by enabling this feature by setting the `range` attribute to true and providing a separate input field to attach the range value. Doing so creates a range slider with a double handle and a track bar.

```

<h:outputText id="displayRange"
    value="Between #{sliderBean.number6} and #{sliderBean.number7}" />

<p:slider for="txt6,txt7" display="displayRange" style="width:400px" range="true"
    displayTemplate="Between {min} and {max}" />

<h:inputHidden id="min" value="#{sliderBean.min}" />
<h:inputHidden id="max" value="#{sliderBean.max}" />

```

Client Side Callbacks

Sider provide three callback methods to handle the start and end events. A slider receives the start event and the ui object contains information about the event.

```

<h:inputText id="number" value="#{sliderBean.number}" />

<p:slider for="number" onSlideEnd="handleSlideEnd(event, ui)"/>

```

```

function handleSlideEnd(event, ui) {
    //ui.helper = Handle element of slider
    //ui.value = Current value of slider
}

```

Ajax Behavior Events

Sider provides one additional behavior event called *slideEnd* that is fired when the slider completes. If you have a listener defined it will be called by a message in, *org.primefaces.event.SlideEndEvent* instance. To add a behavior and define it using, *update="msgs"*.

```

<h:inputText id="number" value="#{sliderBean.number}" />

<p:slider for="number">
    <p:ajax event="slideEnd" listener="#{sliderBean.onSlideEnd}" update="msgs" />
</p:slider>

<p:messages id="msgs" />

```

```

public class SliderBean {

    private int number;

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    public void onSlideEnd(SlideEndEvent event) {
        int value = event.getValue();
        //add faces message
    }
}

```

Client Side API

Wid, et PrimeFaces.widget.Slider

Method	Params	Return Type	Description
, et>a(ue?)	B	<u " ber	Return the current va(ue
et>a(ue?)va(ue?)	va(ue) >a(ue to et	void	U&date (ider va(ue ' ith &rovided one.
di ab(e)?	indeON IndeO o+ tab to di ab(e	void	Di ab(e (ider.
enab(e)?	indeON IndeO o+ tab to enab(e	void	Enab(e (ider.

Skinning

S(ider re ide in a "ain container ' hich style and styleClass attribute a&&(y. 9he e attribute are handy to &eci;y the di " en ion o+ the (ider. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.uiB (ider	8 ain container e(e " ent
.uiB (iderBhori=onta(8 ain container e(e " ent o+ hori=onta((ider
.uiB (iderBvertica(8 ain container e(e " ent o+ vertica((ider
.uiB (iderBhand(e	S(ider hand(e

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or " ore in+or " ation.

3.112 Spotlight

S&ot(i , ht hi , h(i , ht a certain co " &onent on &a , e.

Spotlight

Spotlight highlights a certain component on page.

New User

Firstname: * ✖ Firstname: Validation Error: Value is required.

Surname: * ✖ Surname: Validation Error: Value is required.

Source

```
spotlight.xhtml
```

Info

9a,	spotlight
*o " &onent *(a	org.primefaces.component.spotlight.Spotlight
*o " &onent 9y&e	org.primefaces.component.Spotlight
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SpotlightRenderer
Renderer *(a	org.primefaces.component.spotlight.SpotlightRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	#{!empty}	Boolean	Boolean value to specify whether the component is rendered or not. If set to false, the component will not be rendered.
binding	null	Object	An expression indicating that it is bound to a server-side UI component in a backing bean.
' id, etc>ar	null	String	ar" etc>ar>
target	null	String	*o " &onent to hi , h(i , ht.
animate	true	Boolean	*ontrol(animation used during, hi , h(i , ht.

Getting started with Spotlight

When a command button is clicked, the button highlights the element it is contained in.

```
<p:panel id=" pnl" header="Panel">
    //content
</p:panel>

<p:commandButton value="Highlight" onclick="PF('spot').show()" />

<p:spotlight target="#pnl" widgetVar="spot" />
```

Client Side API

Widget `PrimeFaces.widget.Spotlight`

Method	Params	Return Type	Description
<code>highlight()</code>	None	void	Highlights the target element.
<code>hide()</code>	None	void	Hides the highlighted element.

Skinning

Style rules defined in a CSS container which `style` and `styleClass` attributes apply. These attributes are handy to specify the style of the container. A class name, type class are, obviously, the main "in, effect or "more important" action. #of(o ' in, in the (into) structure(type class)

Class	Applies
.ui-highlighted	Applies to all elements.
.ui-highlighted-to	Applies to the element.
.ui-highlighted-button	Applies to a button element.
.ui-highlighted-elem	Applies to an element.
.ui-highlighted-rib	Applies to the ribbon element.

3.114 Socket

Soc3et co " &onent i an a,ent that create a &u h channe(bet ' een the erver and the c(ient.

Info

9a,	socket
*o " &onent *(a	org.primefaces.component.socket.Socket
*o " &onent 9y&e	org.primefaces.component.Socket
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SocketRenderer
Renderer *(a	org.primefaces.component.socket.SocketRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7b/ect	An e(eore ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
channe(nu((7b/ect	*hanne(na " e o+ the connection.
tran &ort	' eb oc3et	Strin,	De ired &rotoco(to be u ed va(id va(ue are ' eb oc3et ?de+au(t@E eE trea " in,E(on,B&o((in,E / on&.
+a((bac39ran &ort	(on,B&o((in,	Strin,	#a((bac3 &rotoco(to be u ed ' hen de ired tran &ort i not u&&orted in environ " entE va(id va(ue are ' eb oc3etE eE trea " in,E(on,B&o((in,E ?de+au(t@E / on&.
on 8 e a,e	nu((Strin,	Aava cri&t event hand(er that i &roce ed ' hen erver &ub(i he data.
onError	nu((Strin,	Aava cri&t event hand(er that i &roce ed ' hen there i an error.
on*(o e	nu((Strin,	Aava cri&t event hand(er +or on*(o e ca((bac3 o+ at " o &here.
on7&en	nu((Strin,	Aava cri&t event hand(er +or on7&en ca((bac3 o+ at " o &here.
onReconnect	nu((Strin,	Aava cri&t event hand(er +or onReconnect ca((bac3 o+ at " o &here.

Name	Default	Type	Description
onBeforeEvent	null	String	A Java script event handler or onBeforeEvent listener callback function at "onBeforeEvent".
onError	null	String	A Java script event handler or onError listener callback function at "onError".
onFocus	null	String	A Java script event handler or onFocus listener callback function at "onFocus".
autoConnect	true	Boolean	*connect to channel on &enable when enabled.

Getting Started with the Socket

See chapter 5 for more information or detailed information.

3.115 Spacer

Spacer is used to put space between elements.

Info

9a,	spacer
*o "&onent *(a	org.primefaces.component.spacer.Spacer
*o "&onent 9y&e	org.primefaces.component.Spacer
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SpacerRenderer
Renderer *(a	org.primefaces.component.spacer.SpacerRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
title	null	String	Alternative text for the spacer.
type	null	String	Style class of the spacer.
width	null	String	Width of the spacer.
height	null	String	Height of the spacer.

Getting started with Spacer

Spacer is used by either width or height of the spacer.

Spacer in this example separates this text <p:spacer width="100" height="10"> and <p:spacer width="100" height="10"> this text.

Spacer in this example separates this text and this text.

3.116 Spinner

Spinner is an input component to provide a numeric value via increment and decrement buttons.



Info

9a,	spinner
*o " &onent *(a	org.primefaces.component.spinner.Spinner
*o " &onent 9y&e	org.primefaces.component.Spinner
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SpinnerRenderer
Renderer *(a	org.primefaces.component.spinner.SpinnerRenderer

Attributes

Name	Default	Type	Description
id	nu((String	Unique identifier for the component.
rendered	true	Boolean	Value indicating whether the component should be rendered. If set to false, the component will not be rendered.
binding	nu((Object	An expression or function that maps to a server-side IDE UI component instance in a backing bean.
value	nu((Object	The value of the component, which can be either an E6 expression or an iterator.
converter	nu((Converter	An expression or function that defines a converter for the component. When it's an E6 expression, it refers to a converter instance. In case it's a static method, it refers to a converter id.
immediate	#a(e	Boolean	A value that indicates whether the component is immediately processed by the event loop. If true, the component's value is updated before the event is fired at the Ia&&(y reCue t value IE i+ immediate is set to true, the value is updated before the event is fired in I&roce va(idation I &ha e
required	#a(e	Boolean	Specifies if the component is required.
validator	nu((Method	A method binding expression that refers to a validation method, the input

Name	Default	Type	Description
va(ue*han,e6i tener	nu()	8ethod E0&r	A " method bindin, e0&re ion that re+er to a " method +or hand(in, a va(uchan,eevent
reCquired 8 e a,e	nu()	Strin,	8 e a,e to be di &(ayed ' hen reCquired +ie(d va(idation +ai(.
converter 8 e a,e	nu()	Strin,	8 e a,e to be di &(ayed ' hen conver ion +ai(.
va(idator 8 e a,e	nu()	Strin,	8 e a,e to be di &(ayed ' hen va(idation +ie(d .
' id,et>ar	nu()	Strin,	<a " e o+ the c(ient ide ' id,et.
te&#actor	1	Doub(e	Ste&&in, +actor +or each incre " ent and decre " ent
" in	nu()	Doub(e	8 ini " u " boundary va(ue
" a0	nu()	Doub(e	8 a0i " u " boundary va(ue
&re+i0	nu()	Strin,	!re+i0 o+ the in&ut
u+i0	nu()	Strin,	Su+i0 o+ the in&ut
acce 3ey	nu()	Strin,	Acce 3ey that ' hen &re ed tran +er +ocu to the in&ut e(e " ent.
a(t	nu()	Strin,	A(ternate te0tua(de cri&tion o+ the in&ut +ie(d.
autoco " &(ete	nu()	Strin,	*ontro(bro ' er autoco " &(ete behavior.
dir	nu()	Strin,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
di ab(ed	+a(e	1oo(ean	Di ab(e in&ut +ie(d
(abe(nu()	Strin,	A(oca(i=ed u er &re entab(e na " e.
(an,	nu()	Strin,	*ode de cribin, the (an,ua,e u ed in the ,enerated " ar3u& +or thi co " &onent.
" a0(en,th	nu()	Inte,er	8 a0i " u " nu " ber o+ character that " ay be entered in thi +ie(d.
onb(ur	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu .
onchan,e	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent (o e +ocu and it va(ue ha been " odi+ied inc ,ainin, +ocu .
onc(ic3	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i c(ic3ed.
ondb(c(ic3	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent i doub(e c(ic3ed.
on+ocu	nu()	Strin,	*(ient ide ca((bac3 to e0ecute ' hen in&ut e(e " ent receive +ocu .

Name	Default	Type	Description
on3eydo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over in&ut e(e " ent.
on3ey&re	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over in&ut e(e " ent.
on3eyu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over in&ut e(e " ent.
on " ou edo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over in&ut e(e " ent
on " ou e " ove	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin in&ut e(e " ent.
on " ou eout	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " in&ut e(e " ent.
on " ou eover	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto in&ut e(e " ent.
on e(ect	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin in&ut e(e " ent i e(ected by u er.
&(aceho(der	nu((Strin,	S&eci+ie a hort hint.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&event chan,e by the u er.
i=e	nu((Inte,er	<u " ber o+ character u ed to deter " ine the ' idth o+ the in&ut e(e " ent.
ty(e	nu((Strin,	In(ine ty(e o+ the in&ut e(e " ent.
ty(e*(a	nu((Strin,	Sty(e c(a o+ the in&ut e(e " ent.
tabinde0	nu((Inte,er	!o ition o+ the in&ut e(e " ent in the tabbin, order.
tit(e	nu((Strin,	Advi ory too(ti& in+or " aton.

Getting Started with Spinner

Spinner is an in&ut co " &onent and u ed /u t (i3e a standard in&ut te0t.

```
public class SpinnerBean {
    private int number;
    //getter and setter
}
```

```
<p:spinner value="#{spinnerBean.number}" />
```

Step Factor

Other than integer and inner attributes, the fraction and part can be controlled with the stepFactor attribute to specify the increment amount. For example, setting stepFactor="0.25" will result in a step of 0.25.

```
<p:spinner value="#{spinnerBean.number}" stepFactor="0.25"/>
```

```
public class SpinnerBean {  
    private double number;  
    //getter and setter  
}
```

After this inner output is displayed:



After an increase event has been triggered on the input field:



Prefix and Suffix

Prefix and Suffix option enables this functionality. Note that you would need to use a converter to avoid conversion error since the prefix is to be added.

```
<p:spinner value="#{spinnerBean.number}" prefix="$" />
```



Boundaries

In order to restrict the boundary values use min and max option.

```
<p:spinner value="#{spinnerBean.number}" min="0" max="100"/>
```

Ajax Spinner

Spinner can be animated using client behavior (like `onchange` or `onblur`). In addition, an ajax update is done to update the output whenever a inner button is clicked.

```
<p:spinner value="#{spinnerBean.number}">  
    <p:ajax update="display" />  
</p:spinner>  
  
<h:outputText id="display" value="#{spinnerBean.number}" />
```

Skinning

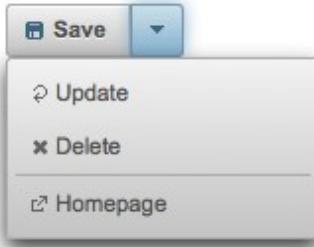
Skinnable components are those that have a container element that uses `style` and `styleClass` attributes. These attributes can be used to change the visual structure of the component.

Class	Applies
.uiB &inner	Span in container element of inner
.uiB &innerbin&ut	Input + button
.uiB &innerBbutton	Skinnable button
.uiB &innerBbuttonBu&	Increment button
.uiB &innerBbuttonBdo 'n	Decrement button

A skinning template can be applied to the span element, section or input+button section.

3.117 SplitButton

Split button di & (ay a co " " and by default and additiona(one in an over(ay.



Info

9a,	splitButton
*o " &onent *(a	org.primefaces.component.splitbutton.SplitButton
*o " &onent 9y&e	org.primefaces.component.SplitButton
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SplitButtonRenderer
Renderer *(a	org.primefaces.component.splitbutton.SplitButtonRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify if the component should be rendered. If false, it will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
value	null	String,	Value for the button.
action	null	MethodExpression or String,	A method expression or a String, outcome that's to be processed when the button is clicked.
actionListener	null	MethodExpression	An action listener that's to be processed when the button is clicked.
immediate	false	Boolean	Boolean value that determines if the action is to be processed when the button is clicked.
type	button	String,	Set the behavior of the button.
auto	true	Boolean	Specifies the button's type to true?default:button if the button is not defined.

Name	Default	Type	Description
async	+a(e	1oo(ean	When set to trueE a/a0 reCue t are not Cueued.
&roce	nu((Strin,	*o " &onent? @ to &roce &artia((y in tead o+ ' ho(e vie ' .
u&date	nu((Strin,	*o " &onent? @ to be u&dated ' ith a/a0.
on tart	nu((Strin,	*(ient ide ca((bac3 to e0ecute be+ore a/a0 reCue t i be,in .
onco " &(ete	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a/a0 reCue t i co " &(eted.
on ucce	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a/a0 reCue t ucceed .
onerror	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a/a0 reCue t +ai(.
,(oba(true	1oo(ean	De+ine ' hether to tri , ,er a/a0Statu or not.
de(ay	nu((Strin,	I+ (e than delay " i((i econd e(a& e bet ' een ca((to request() on(y the " o t recent one i ent and a((other reCue t are di carded. I+ thi o<ption i not &eci+iiedE or i+ the va(ue o+ delay i the (itera(trin, lnone) ' ithout the Cuote E no de(ay i u ed.
&artia(Sub " it	+a(e	1oo(ean	Enab(e eria(i=ation o+ va(ue be(on,in, to the &artia((y &roce ed co " &onent on(y.
re et>a(ue	+a(e	1oo(ean	I+ trueE oca(va(ue o+ in&ut co " &onent to be u&dated ' ithin the a/a0 reCue t ' ou(d be re et.
i,noreAutoU&date	+a(e	1oo(ean	I+ trueE co " &onent ' hich autoU&datePItrueI ' i((not be u&dated +or thi reCue t. I+ not &eci+iiedE or the va(ue i +a(eE no uch indication i " ade.
ty(e	nu((Strin,	In(in e ty(e o+ the button e(e " ent.
ty(e*(a	nu((Strin,	Sty(e*(a o+ the button e(e " ent.
onb(ur	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen button (o e +ocu .
onchan,e	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen button (o e +ocu and it va(ue ha been " odi+iied ince ,ainin ,+ocu .
onc(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen button i c(ic3ed.
ondb(c(ic3	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen button i doub(e c(ic3ed.
on+ocu	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen button receive +ocu .
on3eydo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed do ' n over button.
on3ey&re	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i &re ed and re(ea ed over button.

Name	Default	Type	Description
on3eyu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a 3ey i re(ea ed over button.
on " ou edo ' n	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i &re ed do ' n over button.
on " ou e " ove	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved ' ithin button.
on " ou eout	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved a ' ay +ro " button.
on " ou eover	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i " oved onto button.
on " ou eu&	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen a &ointer button i re(ea ed over button.
on e(ect	nu((Strin,	*(ient ide ca((bac3 to e0ecute ' hen te0t ' ithin button i e(ected by u er.
acce 3ey	nu((Strin,	Acce 3ey that ' hen &re ed tran +er +ocu to the button.
a(t	nu((Strin,	A(ternate te0tua(de cri&tion o+ the button.
dir	nu((Strin,	Direction indication +or te0t that doe not inherit directiona(ity. >a(id va(ue are 69R and R96.
di ab(ed	+a(e	1oo(ean	Di ab(e the button.
i " a,e	nu((Strin,	Sty(e c(a +or the button icon. ?de&recatedN u e icon@
(abe(nu((Strin,	A (oca(i=ed u er &re entab(e na " e.
(an,	nu((Strin,	*ode de cribin, the (an, ua, e u ed in the ,enerated " ar3u& +or thi co " &onent.
tabinde0	nu((Inte,er	!o ition o+ the button e(e " ent in the tabbin, order.
tit(e	nu((Strin,	Advi ory too(ti& in+or " ation.
readon(y	+a(e	1oo(ean	#(a, indicatin, that thi co " &onent ' i((&revent chan,e by the u er.
icon	nu((Strin,	Icon o+ the button a a c c(a .
icon !o	(e-t	Strin,	!o ition o+ the icon.
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et.
a&&end9o	nu((Strin,	A&&end the over(ay to the e(e " ent de+ined by earch e0&re ion. De+au(t to docu " ent body.

Getting started with SplitButton

SplitButton is a button that contains a menu. Additionally, it can contain other buttons like the Save, Update, Delete and Home buttons shown below. The menu items are defined in the `menuitem` tag.

```
<p:splitButton value="Save" actionListener="#{buttonBean.save}" update="messages"
    icon="ui-icon-disk">
    <p:menuitem value="Update" actionListener="#{buttonBean.update}" update="messages" icon="ui-icon-arrowrefresh-1-w"/>
    <p:menuitem value="Delete" actionListener="#{buttonBean.delete}" ajax="false" icon="ui-icon-close"/>
    <p:separator />
    <p:menuitem value="Homepage" url="http://www.primefaces.org" icon="ui-icon-extlink"/>
</p:splitButton>
```

Client Side API

Widet `PrimeFaces.widget.SplitButton`

Method	Params	Return Type	Description
<code>show()</code>	B	void	Display overlay.
<code>hide()</code>	B	void	Hide overlay.

Skinning

SplitButton renders a container element which has style and styleClass attributes. Inside the container, there is the following structure:

Style Class	Applies
.ui-splitbutton	* container element.
.ui-splitbutton	SplitButton element
.ui-splitbutton-button	Drop-down button
.ui-splitbutton-button-only	SplitButton element which has icon but no button
.ui-splitbutton-button	Label for button
.ui-splitbutton-enu	* container element of "envelope" type
.ui-splitbutton-enu-item	Container for item
.ui-splitbutton-enu-item	Each "envelope" item
.ui-splitbutton-enu-item	Anchor element in a "envelope" item
.ui-splitbutton-enu-item	Text of element in an item

A skinning template can be applied to the "envelope" in, selection or "more information" action.

3.118 Submenu

Submenu is nested in menu component and represents a sub-menu item.

Info

9a,	submenu
*o "&onent *(a	org.primefaces.component.submenu.Submenu
*o "&onent 9y&e	org.primefaces.component.Submenu
*o "&onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component.
rendered	true	Boolean	Value indicating whether the component is rendered. If false, it will not be rendered.
binding	nu((Object	An expression that binds to a server-side UI component in a backing bean.
(abefore	nu((String,	Value of the sub-menu header.
icon	nu((String,	Icon of a sub-menu item.
type	nu((String,	Inline style of the sub-menu.
type*(a	nu((String,	Style class of the sub-menu.

Getting started with Submenu

Learn how to bind out-put from a sub-menu to a backing bean.

3.119 Stack

Stack is a navigation component that implements the tac3 feature in 8 ac 7S F.



Info

9a,	stack
*o " &onent *(a	org.primefaces.component.stack.Stack
*o " &onent 9y&e	org.primefaces.component.Stack
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.StackRenderer
Renderer *(a	org.primefaces.component.stack.StackRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	Unique identifier of the component.
rendered	true	1oo(ean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin ,	nu((7b/ect	An expression that allows to a server-side UI component in tance in a bac3in , bean.
icon	nu((Strin ,	An optional attribute to contain tac3ed ite" .
o&enS&eed)00	Strin ,	Seed of the animation when o&enin , the tac3.
c(o eS&eed)00	Inte ,er	Seed of the animation when o&enin , the tac3.
' id ,et>ar	nu((Strin ,	<a " e of the client side ' id ,et.
" ode(nu((8 enu 8 ode(8 enu 8 ode(in tance to create " enu &ro ,ra " " atica((y
e0&anded	+a(e	1oo(ean	Whether to display tac3 a e0&anded or not.

Getting started with Stack

Each item in the stack is represented with "envelope". Stack below has five items with different icon and label.

```
<p:stack icon="/images/stack/stack.png">
    <p:menuitem value="Aperture" icon="/images/stack/aperture.png" url="#" />
    <p:menuitem value="Photoshop" icon="/images/stack/photoshop.png" url="#" />
    //...
</p:stack>
```

Initially stack will be rendered in collapsed mode:



Location

Stack is positioned by default and location can be changed via CSS. Here's one important selector for stack class .ui-stack. To override this style to change the location of stack.

```
.ui-stack {
    bottom: 28px;
    right: 40px;
}
```

Dynamic Menus

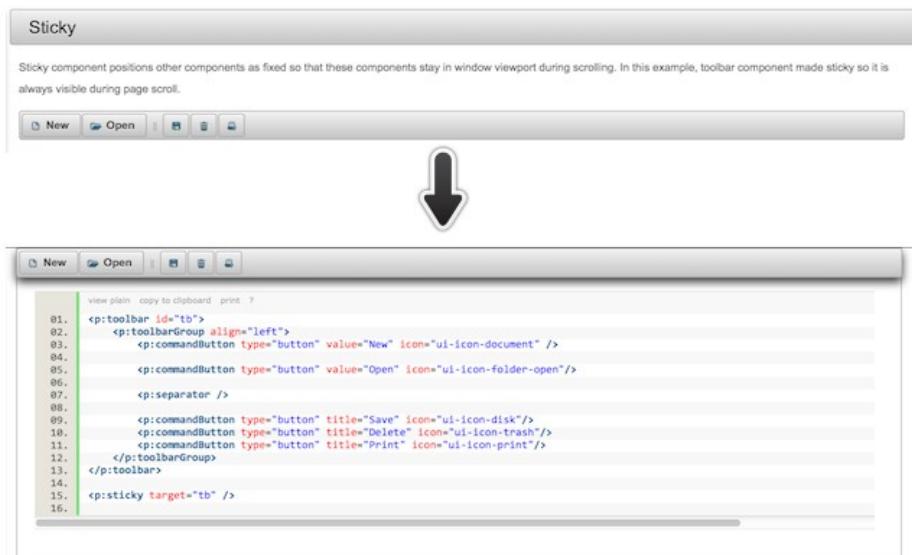
Menus can be created dynamically using a specific dynamic menu component "menu" & part of "menu" component. It can be created in several ways or in combination and an example is given.

Skinning

Class	Applies
.ui-stack	Main container element of stack
.ui-stack ul(i)	Each item in stack
.ui-stack ul(i) i	Icon of a stack item
.ui-stack ul(i) &an	Label of a stack item

3.120 Sticky

Sticky component positions other components as fixed so that these components stay in window viewport during scrolling. In this example, toolbar component made sticky so it is always visible during page scroll.



Info

9a,	sticky
*o " &onent *(a	org.primefaces.component.sticky.Sticky
*o " &onent 9y&e	org.primefaces.component.Sticky
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.StickyRenderer
Renderer *(a	org.primefaces.component.sticky.StickyRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	null	Object	An expression reference that binds to a server-side UI component instance in a backing bean.
target	null	String	*o " &onent to " a3e tic3y.
sharing	0	Integer	Sharing information between components during scrolling.

Getting started with Sticky

Sticky requires a toolbar to see it in view & sort on scroll. There is a sticky toolbar.

```
<p:toolbar id="tb">
    <p:toolbarGroup align="left">
        <p:commandButton type="button" value="New" icon="ui-icon-document" />
        <p:commandButton type="button" value="Open" icon="ui-icon-folder-open"/>

        <p:separator />

        <p:commandButton type="button" title="Save" icon="ui-icon-disk"/>
        <p:commandButton type="button" title="Delete" icon="ui-icon-trash"/>
        <p:commandButton type="button" title="Print" icon="ui-icon-print"/>
    </p:toolbarGroup>
</p:toolbar>

<p:sticky target="tb" />
```

Skinning

There are no visual styles or themes for the *ui-sticky* component attached to the toolbar, so when the position is changed. When the toolbar is repositioned to its original location, it is "oved.

3.121 SubTable

SubTable is a header component for data tables.

FCB Statistics		
Player	Goals	Assists
Messi		
2005-2006	4	2
2006-2007	10	7
2007-2008	16	10
2008-2009	32	15
2009-2010	51	22
2010-2011	55	30
Totals:	168	86
Xavi		
2005-2006	6	15
2006-2007	10	20
2007-2008	12	22
2008-2009	9	24
2009-2010	8	21
2010-2011	10	25
Totals:	55	127
Iniesta		
2005-2006	4	12
2006-2007	7	9
2007-2008	10	14
2008-2009	15	17
2009-2010	14	16
2010-2011	17	22
Totals:	67	90

Info

9a,	subTable
*o " &onent *(a	org.primefaces.component.subtable.SubTable
*o " &onent 9y&e	org.primefaces.component.SubTable
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SubTableRenderer
Renderer *(a	org.primefaces.component.subtable.SubTableRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	Unique identifier of the component
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
bindin ,	nu((7b/ect	An expression that refers to a server-side UI component instance in a backing bean
va(ue	nu((7b/ect	Data of the component.
var	nu((Strin ,	<a " e of the data iterator.

Getting started with SubTable

See DataTable section for more information.

3.122 SummaryRow

Summary Row is a header row containing summary information.

Model	Year	Manufacturer	Color
20b7dd32	1983	Volvo	Orange
93583964	1962	Volvo	White
6e68d612	1970	Volvo	Brown
a127d75d	1968	Volvo	Black
3d5ba523	1994	Volvo	Red
Total:			51545\$
4d784acf	2002	Volkswagen	Red
0e43ef6e	1978	Volkswagen	Black
4b0ee961	1960	Volkswagen	Red
8b1bddef	2008	Volkswagen	White
Total:			80121\$
40b0c19d	2000	Renault	Green
a56ff6ee	1967	Renault	Maroon
ec645794	1983	Renault	Green
Total:			67468\$

Info

9a,	summaryRow
*o " &onent *(a	org.primefaces.component.summaryrow.SummaryRow
*o " &onent 9y&e	org.primefaces.component.SummaryRow
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.SummaryRowRenderer
Renderer *(a	org.primefaces.component.summaryrow.SummaryRowRenderer

Attributes

Name	Default	Type	Description
id	nu((String	Unique identifier of the component
rendered	true	Boolean	Value to specify whether the component is rendered or not. If set to false, the component will not be rendered.
binding	nu((Object	An expression that refers to a server-side UI component instance in a backing bean
(tener	nu((Method	Method expression to execute before rendering, such as to calculate total.

Getting started with SummaryRow

See DataTable section for more information.

3.123 Tab

The Tab component is a generic container component used by other PrimeFaces components such as tabView and accordionPanel.

Info

Value	tab
Component Type	org.primefaces.component.TabView.Tab
Parent Component	org.primefaces.component.Tab
Class Name	org.primefaces.component

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	Boolean value to specify whether the component is rendered. If set to false, the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
title	null	String	Title of the tab.
titleStyle	null	String	Inline style for the tab.
titleStyleClass	null	String	Style class for the tab.
disabled	false	Boolean	Disables the tab element.
disabled	false	Boolean	Enables the tab component.
title	null	String	Title of the tab header.

Getting started with the Tab

See the section on component 'how to use tab component' for more information. A tab is a shared component not associated with any specific component that uses it.

3.124 TabMenu

Tab menu is a navigation component that displays "enuite" a tab.



Info

9a,	tabMenu
*o "&onent *(a	org.primefaces.component.tabmenu.TabMenu
*o "&onent 9y&e	org.primefaces.component.TabMenu
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TabMenuRenderer
Renderer *(a	org.primefaces.component.tabmenu.TabMenuRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier o+ the co "&onent
rendered	true	Boolean	Boolean value to &ecify the renderin, o+ the co "&onentE ' hen et to +a(e co "&onent ' i((not be rendered.
bindin,	nu((Object	An e(eore ion that "a& to a erver ide UI*o "&onent in tance in a bac3in, bean
" ode(nu((String	8 enu 8 ode(in tance to bui(d " enu dyna " ica((y.
ty(e	nu((String,	In(ine ty(e o+ the co "&onent.
ty(e*(a	nu((String,	Style class o+ the co "&onent.
activeIndex	0	Integer	Index o+ the active tab.
' id,et>ar	nu((String,	<a " e o+ the c(ient ide ' id,et.

Getting started with TabMenu

Tab menu require "enuite" a chi(dren co "&onent E each "enuite" i rendered a tab. Au t (i3e in any other "enu co "&onentE "enuite" can be uti(i=ed to do a/a0 reCue t E nonBa/a0 reCue t and i "&e GE9 navi ,ation .

```
<p:tabMenu activeIndex="0">
    <p:menuItem value="Overview" outcome="main" icon="ui-icon-star"/>
    <p:menuItem value="Demos" outcome="demos" icon="ui-icon-search" />
    <p:menuItem value="Documentation" outcome="docs" icon="ui-icon-document"/>
    <p:menuItem value="Support" outcome="support" icon="ui-icon-wrench"/>
    <p:menuItem value="Social" outcome="social" icon="ui-icon-person" />
</p:tabMenu>
```

Skinning TabMenu

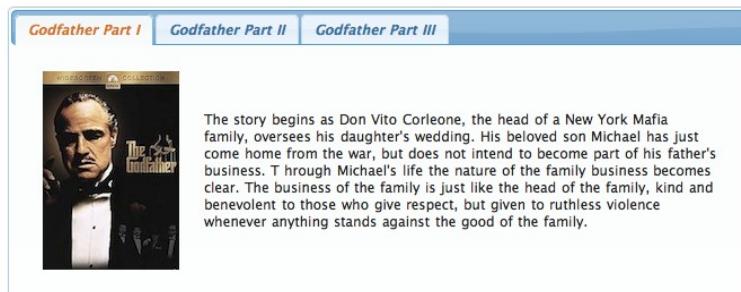
9ab 8 enu re ide in a " ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the
(i t o+ tructura(ty(e c(a e K

Style Class	Applies
.ui-tab " enu	8 ain container e(e " ent.
.ui-tab " enuBnav	*ontainer +or tab .
.ui-tab " enuite "	8 enuite " container.
.uiB " enuite "	Anchor o+ a " enuite " .

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.125 TabView

tabView is a container component that contains content in tabs.



Info

9a,	tabView
*o "&onent *(a	org.primefaces.component.tabview.TabView
*o "&onent 9y&e	org.primefaces.component.TabView
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TabViewRenderer
Renderer *(a	org.primefaces.component.tabview.TabViewRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component when set to false or if not rendered.
binding	null	Object	An expression that refers to a server-side UI component instance in a backing bean.
clientId	null	String	Client identifier of the component.
activeIndex	0	Integer	Index of the active tab.
effect	null	String	Effect of the transition effect.
effectDuration	null	String	Duration of the transition effect.
dynamic	false	Boolean	Enable (or disable) inactive tab.

Name	Default	Type	Description
cache	true	boolean	When tab content are (a=y loaded by a/a0 to , ,(e 8 odee cachin, on(y retrieve the tab content once and ub eCuent to , ,(e o+ a cached tab doe not co " " unicate ' ith erver. I+ cachin, i turned onE tab content are re(oaded +ro " erver each ti " e tab i c(ic3ed.
onTabSelect	null	String	*(ient ide ca((bac3 to e0ecute ' hen a tab i c(ic3ed.
onTabShow	null	String	*(ient ide ca((bac3 to e0ecute ' hen a tab i ho ' n.
onTabSelectOne	null	String	*(ient ide ca((bac3 to e0ecute on tab c(o e.
tab	null	String	In(ine ty(e o+ the " ain container.
tabCaption	null	String	Sty(e c(a o+ the " ain container.
var	null	String	<a " e o+ iterator to re+er an ite " in co((ection.
value	null	Object	*o((ection " ode(to di &(ay dyna " ic tab .
orientation	top	String	7rientation o+ tab header .
dir	(tr	String	De+ine te0t directionE va(id va(value are ltr and rtl.
cropping	+a(e	boolean	When enab(ed) tab header can be cro((ed hori=onta((y in tead o+ ' ra&&in,,.
renderId	true	boolean	9ab>ie ' i a na" in, container thu &re&end it id to it chi(dren by de+au(tE a +a(e va(value turn thi behavior o++ e0ce&t +or dyna " ic tab .

Getting started with the TabView

9ab>ie ' reCuire one " ore chi(d tab co " &onent to di &(ay. 9it(e can a(o be de+ined by u in, "tit(e" +acet.

```
<p:tabView>
    <p:tab title="Tab One">
        <h:outputText value="Lorem" />
    </p:tab>
    <p:tab title="Tab Two">
        <h:outputText value="Ipsum" />
    </p:tab>
    <p:tab title="Tab Three">
        <h:outputText value="Dolor" />
    </p:tab>
</p:tabView>
```

Dynamic Tabs

Here're two tabs in tabview: *non-dynamic* and *dynamic*. Only dynamic tabs' content are rendered to the client; on the other hand in dynamic tabs only the active tab content are rendered and when an inactive tab header is selected its content is loaded with ajax. Dynamic tabs are handy in reducing page size when inactive tabs are already loaded and are enabled by setting the `dynamic="true"` option to true.

```
<p:tabView dynamic="true">
    //tabs
</p:tabView>
```

Content Caching

Dynamic tabs cache their content by default by doing so reactivating a tab doesn't reuse it in an ajax reload since content are cached. If you want to reload content of a tab each time a tab is selected turn off caching by *cache="false"*.

Effects

Content transition effects are controlled with `effect` and `effectDuration` attributes. EffectDuration specifies the speed of the effect: `normal` and `fast` are the valid options.

```
<p:tabView effect="fade" effectDuration="fast">
    //tabs
</p:tabView>
```

Ajax Behavior Events

`tabChange` and `tabClose` are the ajax behavior events that are executed when a tab is changed and closed respectively. Here is an example of a tab behavior in configuration:

```
<p:tabView>
    <p:ajax event="tabChange" listener="#{bean.onChange}" />
    //tabs
</p:tabView>
```

```
public void onChange(TabChangeEvent event) {
    //Tab activeTab = event.getTab();
    //...
}
```

Moreover it is defined to be invoked with an `org.primefaces.event.TabChangeEvent` instance that contains a reference to the new active tab and the accordion item or tab whose event triggered it. It is also an instance of `org.primefaces.event.TabCloseEvent`.

Dynamic Number of Tabs

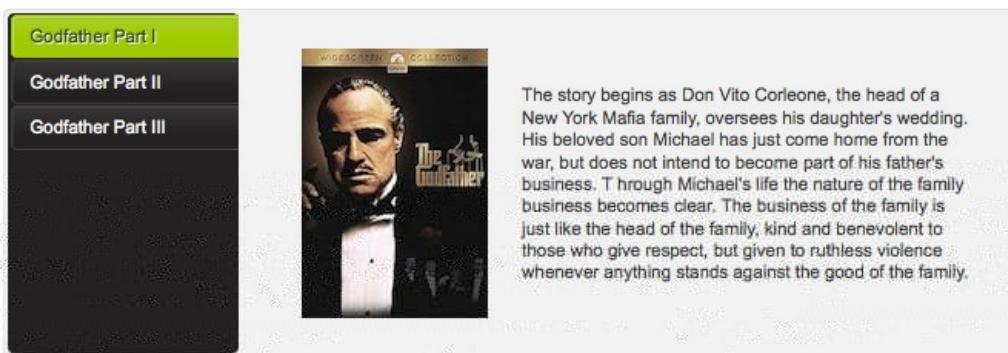
When the tabs to display are not static use the `ui:repeat` iteration feature in combination with `ui:insert`.

```
<p:tabView value="#{bean.list}" var="listItem">
    <p:tab title="#{listItem.propertyA}">
        <h:outputText value= "#{listItem.propertyB}"/>
        ...More content
    </p:tab>
</p:tabView>
```

Orientations

#our different orientation are available (top (default), left, right and bottom).

```
<p:tabView orientation="left">
    //tabs
</p:tabView>
```



Scollable Tabs

9ab header ' relate to the next (ine in case there is not enough space at header area by default. Use scrollable="true" to scroll header area horizontally and navigation button to access hidden header.

```
<p:tabView scrollable="true">
    //tabs
</p:tabView>
```



Client Side Callbacks

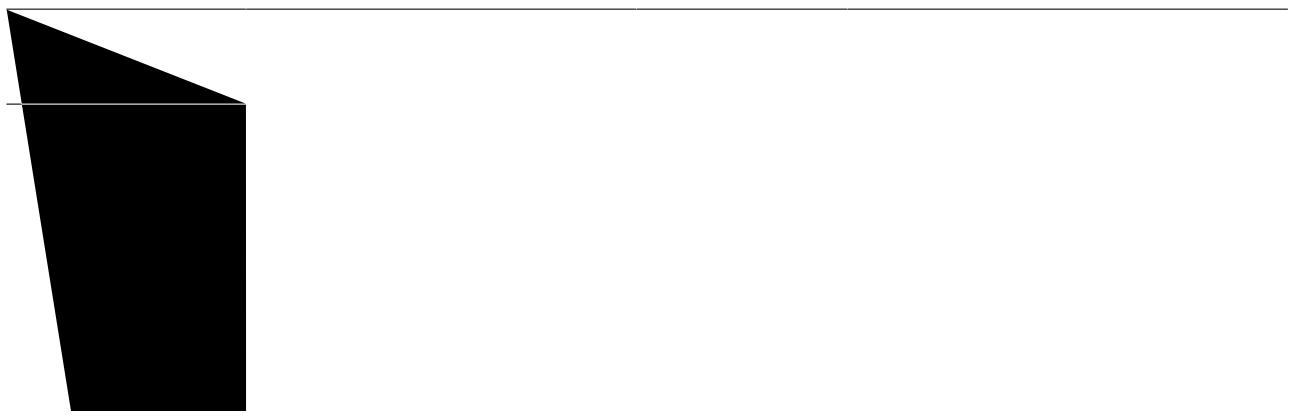
9abview has three callbacks for client side. *onTabChange* is executed when an inactive tab is selected. *onTabShow* is executed when an inactive tab becomes active to be shown and *onTabClose* when a tab is closed. All the callbacks receive index and target a the index of tab.

```
<p:tabView onTabChange="handleTabChange(index)">
    //tabs
</p:tabView>

function handleTabChange(i) {
    //i = Index of the new tab
}
```

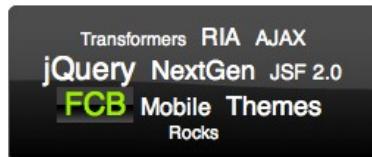
Client Side API

Wid, etn PrimeFaces.widget.TabView



3.126 TagCloud

9a, *(oud di &(ay a co((ection o+ ta, ' ith di+erent tren,th .



Info

9a,	tagCloud
*o " &onent *(a	org.primefaces.component.tagcloud.TagCloud
*o " &onent 9y&e	org.primefaces.component.TagCloud
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TagCloudRenderer
Renderer *(a	org.primefaces.component.tagcloud.TagCloudRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in, bean
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
" ode(nu((9a, *(oud 8 ode(1ac3in, ta, coud " ode(.
ty(e	nu((Strin ,	In(ine ty(e o+ the container e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the container e(e " ent.

Getting started with the TagCloud

9a, *(oud reCuire a bac3end 9a, *(oud " ode(to di &(ay.

```
<p:tagCloud model="#{tagCloudBean.model}" />
```

```

public class TagCloudBean {

    private TagCloudModel model;

    public TagCloudBean() {
        model = new DefaultTagCloudModel();
        model.addTag(new DefaultTagCloudItem("Transformers", "#", 1));
        //more
    }

    //getter
}

```

Selecting Tags

An item in the tag cloud can be selected using the `select` action behavior. Note that only one item with the same URL can be selected.

```

<h:form>
    <p:growl id="msg" showDetail="true" />

    <p:tagCloud model="#{tagCloudBean.model}">
        <p:ajax event="select" update="msg" listener="#{tagCloudBean.onSelect}" />
    </p:tagCloud>
</h:form>

```

```

public class TagCloudBean {

    //model, getter and setter

    public void onSelect(SelectEvent event) {
        TagCloudItem item = (TagCloudItem) event.getObject();
        FacesMessage msg = new FacesMessage(FacesMessage.SEVERITY_INFO,
            "Item Selected", item.getLabel());
        FacesContext.getCurrentInstance().addMessage(null, msg);
    }
}

```

TagCloud API

`org.primefaces.model.tagcloud.TagCloudModel`

Method	Description
<code>getCloud()</code>	Return a tag cloud in the form.
<code>void addTag(TagCloudItem item)</code>	Add a tag.
<code>void removeTag(TagCloudItem item)</code>	Remove a tag.
<code>void clear()</code>	Remove all tags.

The `TagCloudModel` interface provides the default implementation for the tag cloud.

org.primefaces.model.tagcloud.TagCloudItem

Method	Description
String, getLabel()@	Return the label of the tag.
String, getUrl()@	Return the URL of the tag.
int, getStrength()@	Return the strength of the tag, between 1 and 5.

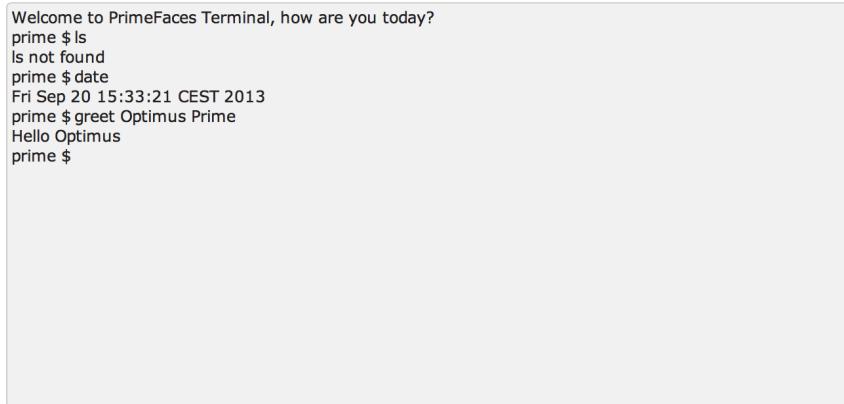
org.primefaces.model.tagcloud.DefaultTagCloudItem is provided with the default implementation.

Skinning

If you're using a container element that has the `style` and `styleClass` attributes set to `.ui-tagcloud`, then each tag will have a class of `.ui-tagcloud-strength-[1,5]` assigned to it. A skinning style can then be applied to each tag, based on its strength value. For example, if you want to make the strongest tags larger, you could use the following CSS:

3.127 Terminal

9er " ina(i an a/a0 &o ' ered ' eb ba ed ter " ina(that brin , de 3to& ter " ina(to AS#.



Info

9a,	terminal
*o "&onent *(a	org.primefaces.component.terminal.Terminal
*o "&onent 9y&e	org.primefaces.component.Terminal
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TerminalRenderer
Renderer *(a	org.primefaces.component.terminal.TerminalRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1 oo(ean	1 oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a server ide UI*o " &onent in tance in a bac3in , bean
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.
' e(co " e 8 e a,e	nu((Strin ,	We(co " e " e a,e to be di &(ayed on initia((oad.
&ro " &t	&ri " e _	Strin ,	! ri " ary &ro " &t te0t.
co " " and . and(er	nu((8 ethodE0&r	8 ethod to be ca((ed ' ith ar,u " ent to &roce .
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.

Getting started with the Terminal

A command and handler is required to interpret commands and enter them in the terminal.

```
<p:terminal commandHandler="#{terminalBean.handleCommand}" />
```

```
public class TerminalBean {

    public String handleCommand(String command, String[] params) {
        if(command.equals("greet"))
            return "Hello " + params[0];
        else if(command.equals("date"))
            return new Date().toString();
        else
            return command + " not found";
    }
}
```

Whenever a command is sent to the server, the `handleCommand()` method is invoked with the command and parameters and returns a String array.

Client Side API

The client-side API includes methods for clearing and focusing the terminal. The `clear()` and `focus()` methods are used to add focus on a terminal instance defined in a dialog.

```
<p:commandButton type="button" value="Apply Focus" onclick="PF(term).focus();"/>
<p:terminal widgetVar="term" commandHandler="#{terminalBean.handleCommand}" />
```

Skinning

Styles are applied in a skin container with `style` and `styleClass` attributes applied to the structural elements of the terminal.

Style Class	Applies
.ui-terminal	Terminal container element.
.ui-terminal-content	Content area of the terminal.
.ui-terminal-row & .ui-terminal-cell	Row and cell elements.

3.128 ThemeSwitcher

The theme switcher enables switching themes, allowing the user to switch themes without reloading the page.



Info

9a,	themeSwitcher
*o "&onent *(a	org.primefaces.component.themeswitcher.ThemeSwitcher
*o "&onent 9y&e	org.primefaces.component.ThemeSwitcher
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ThemeSwitcherRenderer
Renderer *(a	org.primefaces.component.themeswitcher.ThemeSwitcherRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
' id ,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id ,et.
eff ect	+ade	Strin ,	<a " e o+ the ani " ation.
eff ectS&eed	-00	Inte ,er	Duration o+ the ani " ation in " i((i econd .
di ab(ed	+a(e	1oo(ean	Di ab(e the co " &onent.
(abe(nu((Strin ,	U er &re entab(e na " e.
onchan ,e	nu((Strin ,	*(ient ide ca((bac3 to e0ecute on the " e chan ,e.
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e *(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.
var	nu((Strin ,	>ariab(e na " e to re+er to each ite " .
hei ,ht	nu((Inte ,er	. ei ,ht o+ the &ane(.

Name	Default	Type	Description
tabindex0	null	Integer	The position of the element in the tabbin, order.

Getting Started with the ThemeSwitcher

The themeSwitcher is very similar to select7ne 8 enu.

```
<p:themeSwitcher style="width:150px">
    <f:selectItem itemLabel="Choose Theme" itemValue="" />
    <f:selectItems value="#{bean.themes}" />
</p:themeSwitcher>
```

Stateful ThemeSwitcher

By default the themeSwitcher /u t chan,e the the "e on the +y' ith no &a, e re+re hE in ca e you'd (i3e to ,et noti+iied ' hen a u er chan,e the the "e ?e.,.. to u&date u er &reference @E you can u e an a/a0 behavior.

```
<p:themeSwitcher value="#{bean.theme}" effect="fade">
    <f:selectItem itemLabel="Choose Theme" itemValue="" />
    <f:selectItems value="#{themeSwitcherBean.themes}" />
    <p:ajax listener="#{bean.saveTheme}" />
</p:themeSwitcher>
```

Advanced ThemeSwitcher

The themeSwitcher supports di &(ayin, cu to " content o that you can ho ' the "e &revie ' .

```
<p:themeSwitcher>
    <f:selectItem itemLabel="Choose Theme" itemValue="" />
    <f:selectItems value="#{themeSwitcherBean.advancedThemes}" var="theme"
        itemLabel="#{theme.name}" itemValue="#{theme}" />

    <p:column>
        <p:graphicImage value="/images/themes/#{t.image}"/>
    </p:column>

    <p:column>
        #{t.name}
    </p:column>
</p:themeSwitcher>
```

3.129 TieredMenu

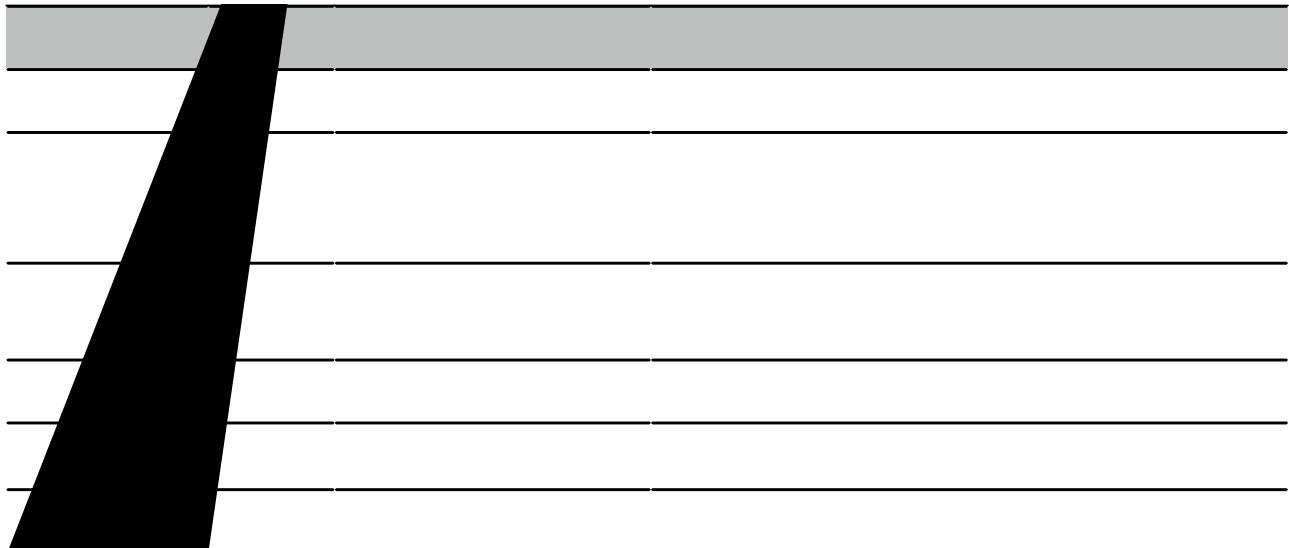
Tiered menu is used to display nested menus with overlapping.



Info

9a,	tieredMenu
*o " &onent *(a	org.primefaces.component.tieredmenu.TieredMenu
*o " &onent 9y&e	org.primefaces.component.TieredMenu
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TieredMenuRenderer
Renderer *(a	org.primefaces.component.tieredmenu.TieredMenuRenderer

Attributes



Name	Default	Type	Description
over(ay	+a(e	1oo(ean	Define &o itionin, E ' hen enab(ed " enu i di &(ayed ' ith ab o(ute &o ition re(ative to the tri, ,er. De+au(t i +a(eE " eanin, static &o itionin,..
tri, ,erEvent	c(ic3	Strin,	Event na " e o+ tri, ,er that ' i((ho ' the dyna " ic &o itioned " enu.

Getting started with the TieredMenu

Tiered menu con i t o+ ub "enu and "enuite" E ub "enu can be ne ted and each ne ted ub "enu ' i((be di &(ayed in an over(ay.

```
<p:tieredMenu>
    <p:submenu label="Ajax Menuitems" icon="ui-icon-refresh">
        <p:menuitem value="Save" actionListener="#{buttonBean.save}"
        update="messages" icon="ui-icon-disk" />
        <p:menuitem value="Update" actionListener="#{buttonBean.update}"
        update="messages" icon="ui-icon-arrowrefresh-1-w" />
    </p:submenu>

    <p:submenu label="Non-Ajax MenuItem" icon="ui-icon-newwin">
        <p:menuitem value="Delete" actionListener="#{buttonBean.delete}"
        update="messages" ajax="false" icon="ui-icon-close"/>
    </p:submenu>

    <p:separator />

    <p:submenu label="Navigations" icon="ui-icon-extlink">
        <p:submenu label="Prime Links">
            <p:menuitem value="Prime" url="http://www.prime.com.tr" />
            <p:menuitem value="PrimeFaces" url="http://www.primefaces.org" />
        </p:submenu>
        <p:menuitem value="Mobile" url="/mobile" />
    </p:submenu>
</p:tieredMenu>
```

AutoDisplay

If default ub "enu are di &(ayed ' hen " ou e i over root " enuitem" E et autoDi &(ay to +a(e to reCuire a c(ic3 on root " enuitem" to enab(e autoDi &(ay " ode.

```
<p:tieredMenu autoDisplay="false">
    //content
</p:tieredMenu>
```

Overlay

Tiered menu can be &o itioned re(ative to a tri, ,er co " &onentE +o((o ' in, a" &(e attache a tiered menu to the button o that ' henever the button i c(ic3ed tiered menu ' i((be di &(ayed in an over(ay it e(+.

```
<p:commandButton type="button" value="Show" id="btn" />
<p:tieredMenu autoDisplay="false" trigger="btn" my="left top" at="left bottom">
    //content
</p:tieredMenu>
```



Client Side API

Wid, etn *PrimeFaces.widget.TieredMenu*

Method	Params	Return Type	Description
show()	B	void	Show / over(ay " enu.
hide()	B	void	. ide over(ay " enu.
at(i,n)	B	void	A(i,n over(ay " enu ' ith tri,,er.

Skinning

Tiered menu re ide in a "ain container ' hich *style* and *styleClass* attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiB"enu .uibtiered "enu	*ontainer e(e " ent o+ " enu.
.uiB"enuB(i t	6i t container
.uiB"enuite "	Each " enu ite "
.uiB"enuite " B(in3	Anchor e(e " ent in a (in3 ite "
.uiB"enuite " Bte0t	9e0t e(e " ent in an ite "

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or " ore in+or " ation.

3.130 Toolbar

A toolbar is a horizontal container component that can contain other components such as buttons, links, and other content.



Info

9a,	toolbar
*o " &onent *(a	org.primefaces.component.toolbar.Toolbar
*o " &onent 9y&e	org.primefaces.component.Toolbar
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.ToolbarRenderer
Renderer *(a	org.primefaces.component.toolbar.ToolbarRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify if the component is rendered. If false, it will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
type	null	String,	Defines the type of the container element.
type*(a	null	String,	Style class of the container element.

Getting Started with the Toolbar

A toolbar has two facets and a header that are defined with facets.

```
<p:toolbar>
    <f:facet name="left"></f:facet>
    <f:facet name="right"></f:facet>
</p:toolbar>
```

Any content can be placed in the toolbar.

```

<p:toolbar>
    <f:facet name="left">
        <p:commandButton type="push" value="New" image="ui-icon-document" />
        <p:commandButton type="push" value="Open" image="ui-icon-folder-open"/>

        <span class="ui-separator">
            <span class="ui-icon ui-icon-grip-dotted-vertical" />
        </span>

        <p:commandButton type="push" title="Save" image="ui-icon-disk"/>
        <p:commandButton type="push" title="Delete" image="ui-icon-trash"/>
        <p:commandButton type="push" title="Print" image="ui-icon-print"/>
    </f:facet>

    <f:facet name="right">
        <p:menuButton value="Navigate">
            <p:menuItem value="Home" url="#" />
            <p:menuItem value="Logout" url="#" />
        </p:menuButton>
    </f:facet>
</p:toolbar>

```

Skinning

900(bar re ide in a container e(e " ent ' hich style and styleClass o<ion a&�((o ' in, i the (i t o+ tructura(ty(e c(a e K

Style Class	Applies
.uiBtoo(bar	8 ain container
.uiBtoo(bar .uiB e&arator	Divider in a too(bar
.uiBtoo(barB, rou&B(e+t	6e+t too(barGrou& container
.uiBtoo(barB, rou&Bri, ht	Ri , ht too(barGrou& container

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.131 ToolbarGroup (Deprecated)

This component is deprecated and will be removed in a future release. Use *left* and *right* facets instead.

Info

9a,	toolbarGroup
*o " &onent *(a	org.primefaces.component.toolbar.ToolbarGroup
*o " &onent 9y&e	org.primefaces.component.ToolbarGroup
*o " &onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier of the component.
rendered	true	Boolean	True (value) to specify the rendering of the component. If false, it will not be rendered.
binding	null	Object	An expression that binds to a server-side UI component in a backing bean.
align	null	String	Defines the alignment of the toolbar within its container (left or right).
type	null	String	Defines the type of the container element.
type*(a	null	String	Style class of the container element.

Getting Started with the ToolbarGroup

See the documentation for more information about how to use this component.

Deprecated

This component is deprecated and will be removed in a future release. Use *left* and *right* facets instead.

3.132 Tooltip

9oo(ti& ,oe beyond the (e,acy ht " (tit(e attribute by &rovidin, cu to " effect E ht " (content and advance the " e u&&ort.



Info

9a,	tooltip
*o " &onent *(a	org.primefaces.component.tooltip.Tooltip
*o " &onent 9y&e	org.primefaces.component.Tooltip
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TooltipRenderer
Renderer *(a	org.primefaces.component.tooltip.TooltipRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1 oo(ean	1 oo(ean va(ue to &eci+y the renderin, o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin,	nu((7 b/ect	An e(e0&re ion that " a& to a erver ide UI *o " &onent in tance in a bac3in, bean
va(ue	nu((7 b/ect	>a(ue o+ the co " &onent than can be either an E6 e0&re ion o+ a (itera(te0t
converter	nu((*onverter() Strin ,	An e(e0&re ion or a (itera(te0t that define a converter +or the co " &onent. When it' an E6 e0&re ionE it' re o(ved to a converter in tance. In ca e it' a static te0tE it " u t re+er to a converter id
' id,et>ar	nu((Strin ,	<a " e o+ the c(ient ide ' id,et.
ho ' Event	" ou eover	Strin ,	Event di &(ayin, the too(ti&.
ho ' E+ect	+ade	Strin ,	E+ect to be u ed +or di &(ayin,..
hideEvent	" ou eout	Strin ,	Event hidin, the too(ti&.
hideE+ect	+ade	Strin ,	E+ect to be u ed +or hidin,..
ho ' De(ay	150	Inte ,er	De(ay ti " e to ho ' too(ti& in " i((i econd .

Name	Default	Type	Description
hideDelay	0	Integer	Delay time to hide tooltip in milliseconds.
for	null	String	* optional to attach the tooltip.
type	null	String	Inline type option for the tooltip.
title*	null	String	Style class for the tooltip.
,(obaseector	null	String	/Query selector for the tooltip to be applied to input button.
ca&e	true	Boolean	Defines whether the field is checked or not.
trackMouseOver	false	Boolean	Follows the mouse cursor on mouseover.

Getting started with the Tooltip

Value can be used by attaching it to a target component "optional. Value can also be retrieved through target element's title attribute, are as follows:

```
<h:inputSecret id="pwd" value="#{myBean.password}" />
<p:tooltip for="pwd" value="Only numbers"/>
```

```
<h:inputSecret id="pwd" value="#{myBean.password}" title="Only numbers"/>
<p:tooltip for="pwd"/>
```

Global Tooltip

Global tooltip binds to elements with title attributes. Ajax updates are supported as well, meaning if target component is updated with ajax, tooltip can still bind. Additionally, are more efficient since only one instance of tooltip is used across all target elements. It is intended to be used instead of explicit tooltip when you have a context change, e.g., different option has context content.

```
<p:tooltip />

<p:inputText id="focus" title="Tooltip for an input"/>
<h:outputLink id="fade" value="#" title="Tooltip for a link">
    <h:outputText value="Fade Effect" />
</h:outputLink>
<p:commandButton value="Up" title="Up" />
```

IE10 Issue

Due to a bug in IE10, the title attribute in a native input element receives focus via tabbing, and the title attribute is not be displayed at once. Solution is to use a through attribute in instead of title.

```

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:pt="http://xmlns.jcp.org/jsf/passthrough"
      xmlns:p="http://primefaces.org/ui">

    <h:head></h:head>

    <h:body>
        <p:inputText pt:data-tooltip="Title here"/>
        <p:inputText title="Works fine except tabbed on IE10"/>
    </h:body>
</html>

```

Events and Effects

A tooltip is shown on mouseover event and hidden when mouseout by default. If you need to change this behavior use the mouseEvent feature. You can be notified when the input, mouseover and hidden with onmouseover.

```

<h:inputSecret id="pwd" value="#{myBean.password}" />
<p:tooltip for="pwd" value="Password must contain only numbers"
           showEvent="focus" hideEvent="blur" showEffect="blind" hideEffect="explode" />

```

Available options for effect are blind, bounce, clip, drop, explode, fold, highlight, puff, pulsate, scale, shake, size and slide.

HTML Content

Another feature of tooltip is the ability to display content as a tooltip.

```

<h:outputLink id="lnk" value="#">
    <h:outputText value="PrimeFaces Home" />
</h:outputLink>

<p:tooltip for="lnk">
    <p:graphicImage value="/images/prime_logo.png" />
    <h:outputText value="Visit PrimeFaces Home" />
</p:tooltip>

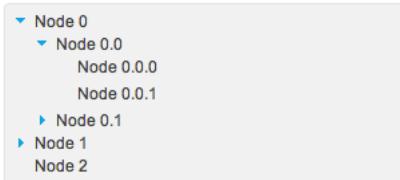
```

Skinning

There is also a style class ui-tooltip and its styling is based on the selector .ui-tooltip. It has "background-color: #333; color: white; border-radius: 5px; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto; font-size: 1em; font-weight: bold; text-decoration: none; text-align: center; border: 1px solid black; opacity: 0.8; transition: opacity 0.2s ease-in-out;".

3.133 Tree

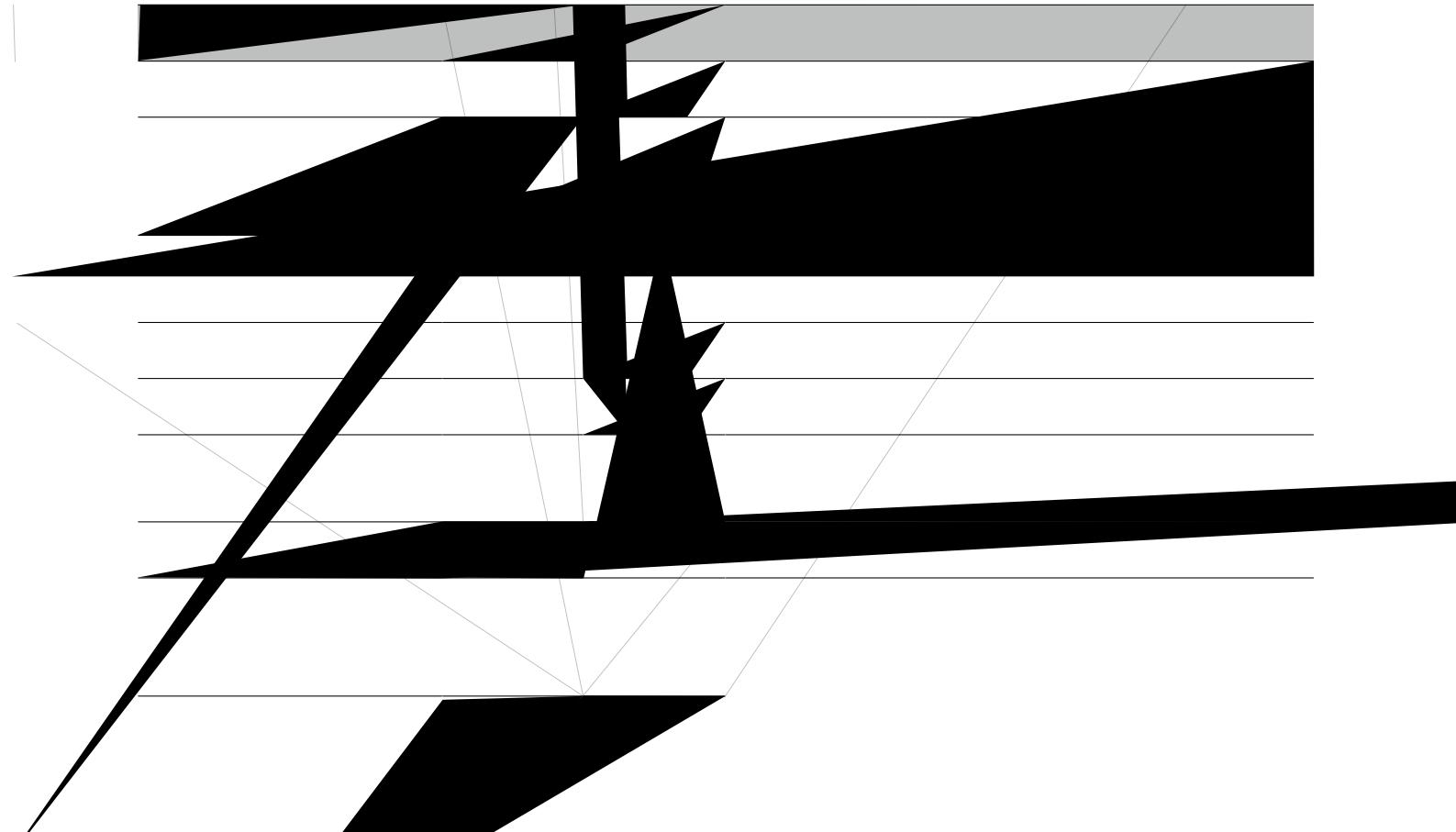
Tree is used for displaying hierarchical data and creating site navigation.



Info

9a,	tree
*o " &onent *(a	org.primefaces.component.tree.Tree
*o " &onent 9y&e	org.primefaces.component.Tree
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TreeRenderer
Renderer *(a	org.primefaces.component.tree.TreeRenderer

Attributes



Name	Default	Type	Description
ty(e	nu((Strin ,	Sty(e o+ the " ain container e(e " ent o+ tree
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the " ain container e(e " ent o+ tree
e(ection 8 ode	nu((Strin ,	De+ine the e(ection 8 ode
hi ,h(i ,ht	true	1oo(ean	. i,h(i ,ht node on hover 'hen e(ection i enab(ed.
data3ey	nu((7b/ect	UniCue 3ey o+ the data &re ented by node .
ani " ate	+a(e	1oo(ean	When enab(ed di &(ay (ide effect on to , ,(e.
orientation	vertica(Strin ,	7rientation o+ (ayout <i>vertical or horizontal.</i>
&ro&a ,ateSe(ectionU&	true	1oo(ean	De+ine u& ' ard e(ection &ro&a ,ation +or chec3boo " ode.
&ro&a ,ateSe(ectionDo ' n	true	1oo(ean	De+ine do ' n ' ard e(ection &ro&a ,ation +or chec3boo " ode.
dir	(tr	Strin ,	De+ine te0t directionE va(id va(ue are <i>ltr</i> and <i>rtl</i> .
dra , ,ab(e	+a(e	1oo(ean	8 a3e tree node dra , ,ab(e.
dro&&ab(e	+a(e	1oo(ean	8 a3e tree dro&&ab(e.
dra ,dro&Sco&e	nu((Strin ,	Sco&e 3ey to ,rou&a et o+ tree co " &onent +or tran +errin, node u in, dra , and dro&.
dra , 8 ode	e(+	Strin ,	De+ine &arentBchi(d re(ation hi& ' hen a node i dra , ,edE va(id va(ue are e(+ ?de+au(t@ &arent and ance tor.
dro&Re trict	none	Strin ,	De+ine &arentBchi(d re triction ' hen a node i dro&&ed va(id va(ue are none ?de+au(t@ and ib(in ,.
reCuired	+a(e	1oo(ean	>a(idation con traint +or e(ection.
reCuired 8 e a,e	nu((Strin ,	8 e a,e +or reCuired e(ection va(idation.

Getting started with the Tree

Tree is &o&u(ated ' ith a *org.primefaces.model.TreeNode* in tance ' hich corre &ond to the root.

```
<p:tree value="#{treeBean.root}" var="node">
    <p:treeNode>
        <h:outputText value="#{node}" />
    </p:treeNode>
</p:tree>
```

```

public class TreeBean {

    private TreeNode root;

    public TreeBean() {
        root = new TreeNode("Root", null);
        TreeNode node0 = new TreeNode("Node 0", root);
        TreeNode node1 = new TreeNode("Node 1", root);
        TreeNode node2 = new TreeNode("Node 2", root);

        TreeNode node00 = new TreeNode("Node 0.0", node0);
        TreeNode node01 = new TreeNode("Node 0.1", node0);

        TreeNode node10 = new TreeNode("Node 1.0", node1);
        TreeNode node11 = new TreeNode("Node 1.1", node1);

        TreeNode node000 = new TreeNode("Node 0.0.0", node00);
        TreeNode node001 = new TreeNode("Node 0.0.1", node00);
        TreeNode node010 = new TreeNode("Node 0.1.0", node01);

        TreeNode node100 = new TreeNode("Node 1.0.0", node10);
    }

    //getter of root
}

```

TreeNode vs p:TreeNode

Tree<ode A! i u ed to create the node "ode(and con i t o+ org.primefaces.model.TreeNode in tance E on the other hand <p:treeNode /> re&re ent a co " &onent o+ ty&e org.primefaces.component.tree.UITreeNode. Mou can bind a 9ree<ode to a &articu(ar &Ntree<ode u in, the type na " e. Docu " ent 9ree e0a " &(e in u&co " in, ection de " on trate a a " &(e u a,e.

TreeNode API

9ree<ode ha a i" &(e A! I to u e 'hen bui(din, the bac3in, "ode(. #or e0a" &(e i+ you ca((node. etE0&anded?true@ on a &articu(ar nodeE tree ' i((render that node a e0&anded.

Property	Type	Description
ty&e	Strin,	ty&e o+ the tree<ode na " eE de+au(t ty&e na " e i Ide+au(tI.
data	7b/ect	Enca& u(ated data
chi(dren	6i tS9ree<odeT	6i t o+ chi(d node
&arent	9ree<ode	!arent node
e0&anded	1oo(ean	#(a, indicatin, ' hether the node i e0&anded or not

Dynamic Tree

Three types of dynamic trees can be defined and used, depending on client-side behavior. In order to enable a tree to be dynamic, it must be set to true.

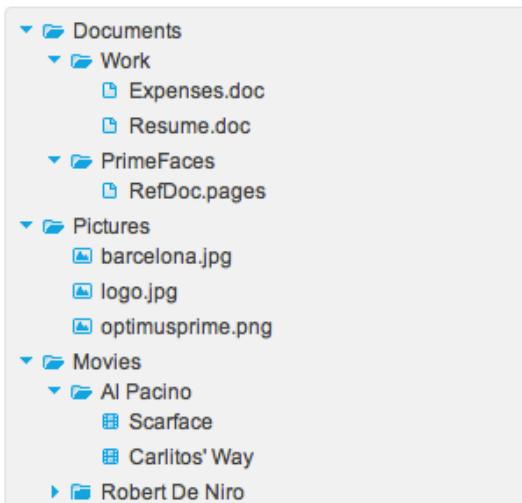
```
<p:tree value="#{treeBean.root}" var="node" dynamic="true">
    <p:treeNode>
        <h:outputText value="#{node}" />
    </p:treeNode>
</p:tree>
```

Non-Dynamic: When set to true, it is set to client-side, so the tree nodes in the code are rendered to the client and tree is created. This mode is suitable for relative data and provides a smoother interaction. On the other hand, it is not suitable for large data since all the data is sent to the client along with the client-side tree structure.

Dynamic: Dynamic mode uses ajax to fetch the tree node from the server side and is intended to be used with the client-side dynamic mode. It has advantages over static mode because it only sends the changes in the tree structure to the client instead of sending the entire tree structure.

Multiple TreeNode Types

It is common to encounter different types of trees with different UI elements, icons, etc. You're using a tree view to visualize a collection of files and folders. Different parts of the tree have different data representations. For example, a folder icon is used for a folder node, while a file icon is used for a file node. In order to achieve this, you can define multiple tree nodes for each type of data.



Documentation is provided for each type of tree node, and additional resources are available online.

```
<p:tree value="#{bean.root}" var="doc">
    <p:treeNode expandedIcon="ui-icon ui-icon-folder-open"
        collapsedIcon="ui-icon ui-icon-folder-collapsed">
        <h:outputText value="#{doc.name}" />
    </p:treeNode>
    <p:treeNode type="document" icon="ui-icon ui-icon-document">
        <h:outputText value="#{doc.name}" />
    </p:treeNode>
    <p:treeNode type="picture" icon="ui-icon ui-icon-image">
        <h:outputText value="#{doc.name}" />
    </p:treeNode>
    <p:treeNode type="mp3" icon="ui-icon ui-icon-video">
        <h:outputText value="#{doc.name}" />
    </p:treeNode>
</p:tree>
```

```

public class Bean {

    private TreeNode root;

    public Bean() {
        root = new TreeNode("root", null);

        TreeNode documents = new TreeNode("Documents", root);
        TreeNode pictures = new TreeNode("Pictures", root);
        TreeNode music = new TreeNode("Music", root);

        TreeNode work = new TreeNode("Work", documents);
        TreeNode primefaces = new TreeNode("PrimeFaces", documents);

        //Documents
        TreeNode expenses = new TreeNode("document", "Expenses.doc", work);
        TreeNode resume = new TreeNode("document", "Resume.doc", work);
        TreeNode refdoc = new TreeNode("document", "RefDoc.pages", primefaces);

        //Pictures
        TreeNode barca = new TreeNode("picture", "barcelona.jpg", pictures);
        TreeNode primelogo = new TreeNode("picture", "logo.jpg", pictures);
        TreeNode optimus = new TreeNode("picture", "optimus.png", pictures);

        //Music
        TreeNode turkish = new TreeNode("Turkish", music);
        TreeNode cemKaraca = new TreeNode("Cem Karaca", turkish);
        TreeNode erkinKoray = new TreeNode("Erkin Koray", turkish);
        TreeNode mogollar = new TreeNode("Mogollar", turkish);

        TreeNode nemalacak = new TreeNode("mp3", "Nem Alacak Felek Benim", cemKaraca);
        TreeNode resimdeki = new TreeNode("mp3", "Resimdeki Goz Yaslari", cemKaraca);

        TreeNode copculer = new TreeNode("mp3", "Copculer", erkinKoray);
        TreeNode oylebirgecer = new TreeNode("mp3", "Oyle Bir Gecer", erkinKoray);

        TreeNode toprakana = new TreeNode("mp3", "Toprak Ana", mogollar);
        TreeNode bisiyapmali = new TreeNode("mp3", "Bisi Yapmali", mogollar);
    }

    //getter of root
}

```

Integration between a Tree and a Tree in the type attribute or else "use icon" in document enter or are represented in, Tree with type I & there are also a Tree component which has its own type I. This is reflected in rendering, all "use icon" node in, that articulates Tree representation which displays a note icon. If "is (y) document and picture have their own Tree representation.

Moreover on the other hand have two state icons are defined by *expandedIcon* and *collapsedIcon* attribute.

Ajax Behavior Events

Provide various behavior events.

Event	Listener Parameter	Fired
e0&and	or,.&ri " e+ace .event.<odeE0&andEvent	When a node i e0&anded.
co((a& e	or,.&ri " e+ace .event.<ode*o((a& eEvent	When a node i co((a& ed.
e(ect	or,.&ri " e+ace .event.<odeSe(ectEvent	When a node i e(ected.
co((a& e	or,.&ri " e+ace .event.<odeUn e(ectEvent	When a node i un e(ected.

#o((o ' in , tree ha three (i tener ¶

```
<p:tree value="#{treeBean.model}" dynamic="true">
    <p:ajax event="select" listener="#{treeBean.onNodeSelect}" />
    <p:ajax event="expand" listener="#{treeBean.onNodeExpand}" />
    <p:ajax event="collapse" listener="#{treeBean.onNodeCollapse}" />
    ...
</p:tree>
```

```
public void onNodeSelect(NodeSelectEvent event) {
    String node = event.getTreeNode().getData().toString();
}

public void onNodeExpand(NodeExpandEvent event) {
    String node = event.getTreeNode().getData().toString();
}

public void onNodeCollapse(NodeCollapseEvent event) {
    String node = event.getTreeNode().getData().toString();
}
```

Event (i tener are a(o u e+u(' hen dea(in , ' ith hu,e a " ount o+ data. 9he idea +or i " &(e " entin , uch a u e ca e ' ou(d be &rovidin , on(y the root and chi(d node to the treeE u e event (i tener to ,et the e(ected node and add ne ' node to that &articu(ar tree at runti " e.

Selection

<ode e(ection i a bui(tBn +eature o+ tree and it u&&ort three di+erent " ode . Se(ection hou(d be a 9ree<ode +or in ,(e ca e and an array o+ 9ree<ode +or " u(ti&(e and chec3bo0 ca e E tree +ind the e(ected node and a i,n the " to your e(ection " ode(.

single¶ 7n(y one at a ti " e can be e(ectedE e(ection hou(d be a 9ree<ode re+erence.

multiple¶ 8u(ti&(e node can be e(ectedE e(ection hou(d be a 9ree<ode`a re+erence.

checkbox¶ 8u(ti&(e e(ection i done ' ith chec3bo0 UIE e(ection hou(d be a 9ree<ode`a re+erence.

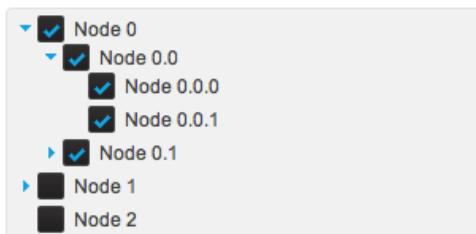
```
<p:tree value="#{treeBean.root}" var="node"
        selectionMode="checkbox"
        selection="#{treeBean.selectedNodes}">
    <p:treeNode>
        <h:outputText value="#{node}" />
    </p:treeNode>
</p:tree>
```

```
public class TreeBean {

    private TreeNode root;
    private TreeNode[] selectedNodes;

    public TreeBean() {
        root = new CheckboxTreeNode("Root", null);
        //populate nodes
    }
    //getters and setters
}
```

What's interesting about the checkbox tree is that when the "checkbox" button is checked, it triggers an event that can be handled by a listener. When the "checkbox" button is checked, it triggers an event that can be handled by a listener. When the "checkbox" button is checked, it triggers an event that can be handled by a listener. When the "checkbox" button is checked, it triggers an event that can be handled by a listener.



Node Caching

When caching is turned on by default, it's important to understand how it works. When a node is not found in memory, a search is performed. In case it's not found, the node is loaded from the database and its children are loaded. This causes the children of the node to be fetched along with the node itself.

Handling Node Click

If you need to execute code when a treenode is clicked, use the `onNodeClick` attribute. This attribute is invoked when the `click` event is triggered on the node and the `onNodeClick` event is fired. In case you have data defined, you can access data on client side by using `node.attrs['data']`.

DragDrop

Tree nodes can be reordered within a tree and can even be transferred between different trees using drag-and-drop. #or a tree enabled, dragging and dropping is supported.

```
<p:tree value="#{treeBean.root}" var="node" draggable="true" droppable="true">
    <p:treeNode>
        <h:outputText value="#{node}"/>
    </p:treeNode>
</p:tree>
```

#or " u(ti&e tree E u e a co&e attribute to " attach the " and con+i, ure dra,dro& o&tion de&endin, on your ca el+o((o ' in, e0a "&e ha % tree ' here one i the ource and other i the tar,et. 9ar, et can a(o be reordered ' ithin it e(+.

```
<p:tree value="#{treeBean.root1}" var="node" draggable="true" droppable="false"
        dragdropScope="myscope">
    <p:treeNode>
        <h:outputText value="#{node}"/>
    </p:treeNode>
</p:tree>

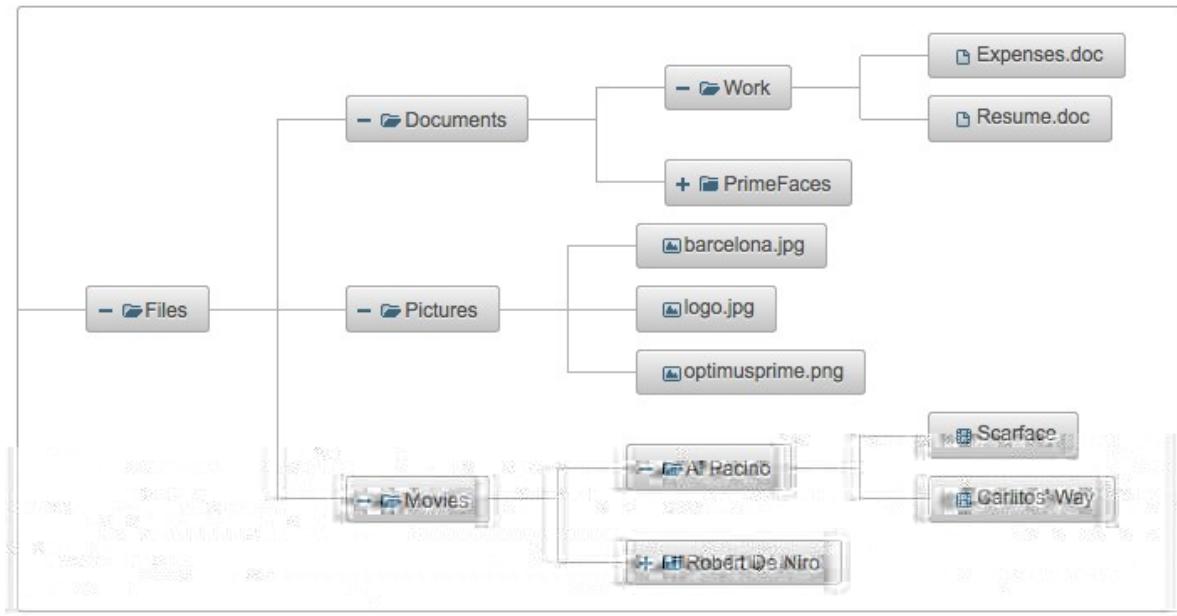
<p:tree value="#{treeBean.root2}" var="node" draggable="true" droppable="true"
        dragdropScope="myscope">
    <p:treeNode>
        <h:outputText value="#{node}"/>
    </p:treeNode>
</p:tree>
```



9 ' o additiona(o&tion e0i t +or +urther con+i, urationE *dragMode* de+ine the tar,et node that ' ou(d be dro&&edE de+au(t va(ue i *self* and other va(ue are *parent* and *ancestor*. *dropRestrict* on the other handE can re trict the dro& tar,et to be ' ithin the &arent by ettin, it to *sibling*.

Horizontal Tree

De+au(t orientation o+ tree i vertica(E ettin, it to hori=onta(di &(ay node in an hori=onta(layout. A((+eature o+ vertica(tree e0ce&t dra,dro& i avai(ab(e +or hori=onta(tree a ' e(.



ContextMenu

Three have been added to the context menu of the tree component. You can even attach different contexts to different tree nodes. In, *nodeType* option of the tree node that attaches the tree node type. Note that selection must be enabled in tree component or context menu integration.

```

<p:contextMenu for="tree">
    <p:menuItem value="View" update="messages"
        actionListener="#{bean.view}" icon="ui-icon-search" />
    <p:menuItem value="View" update="tree"
        actionListener="#{bean.delete}" icon="ui-icon-close" />
</p:contextMenu>

<p:tree id="tree" value="#{bean.root}" var="node"
    selectionMode="single" selection="#{bean.selectedNode}">

    <p:treeNode>
        <h:outputText value="#{node}" />
    </p:treeNode>

</p:tree>
  
```

Skinning

Three ready-made containers each with style and styleClass options available. Note that the structure type can be

Style Class	Applies
.ui-tree	Span container
.ui-treeContainer	Root node container.
.ui-treeNode	Tree node
.ui-treeNodeContent	Tree node content

Style Class	Applies
.ui-tree-node-bicon	Tree node icon
.ui-tree-to-, (er	Root, (e icon
.ui-tree-node-(abe(Tree node (abe(
.ui-tree-node-&arent	<ode ' ith chi(dren
.ui-tree-node-(ea+	<ode ' ithout chi(dren

A 3innin, ty(e c(a e are ,(oba(E ee the "ain the " in, ection +or "ore in+or " ation.

3.134 TreeNode

Tree code is used with tree component to represent a node in tree.

Info

9a,	treeNode
*o "&onent *(a	org.primefaces.component.tree.UITreeNode
*o "&onent 9y&e	org.primefaces.component.UITreeNode
*o "&onent #a " i(y	org.primefaces.component

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component
rendered	true	Boolean	Value indicating whether the component is rendered. If false, it will not be rendered.
binding	nu((Object	A binding expression that maps to a server-side UI component instance in a backing bean.
type	default	String,	Type of the tree node
type*(a	nu((String,	Style class applied to a specific tree node type.
icon	nu((String,	Icon of the node.
expandedIcon	nu((String,	Expanded icon of the node.
collapsedIcon	nu((String,	Collapsed icon of the node.

Getting started with the TreeNode

Tree code is used by tree and tree9ab(e component). Refer to section on the e component for more information.

3.135 TreeTable

9reetab(e i i u ed +or di &(ayin , hierarchica(data in tabu(ar +or " at.

Document Viewer			
Name	Size	Type	
▼ Documents	-	Folder	▷
▶ Work	-	Folder	▷
▶ PrimeFaces	-	Folder	▷
▶ Pictures	-	Folder	▷
▶ Movies	-	Folder	▷

Info

9a,	treeTable
*o "&onent *(a	org.primefaces.component.treetable.TreeTable
*o "&onent 9y&e	org.primefaces.component.TreeTable
*o "&onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.TreeTableRenderer
Renderer *(a	org.primefaces.component.treetable.TreeTableRenderer

Attributes

Name	Default	Type	Description
id	null	String,	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify if the component is rendered or not. If set to false, the component will not be rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.
value	null	Object	A tree node instance in the backing bean.
var	null	String,	<a> element containing the variable used to render each treenode.
id, et>ar	null	String,	<a> element containing the client id.
type	null	String,	Inline type of the container element.
style*(a	null	String,	Style of the container element.

Name	Default	Type	Description
e(ection	nu((7b/ect	Se(ection re+erence.
e(ection 8 ode	nu((Strin,	9y&e o+ e(ection " ode.
cro((ab(e	+a(e	1oo(ean	Whether or not the data hou(d be cro((ab(e.
cro((. ei, ht	nu((Inte,er	. ei, ht o+ cro((ab(e data.
cro((Width	nu((Inte,er	Width o+ cro((ab(e data.
tab(eSty(e	nu((Strin,	In(ine ty(e o+ the tab(e e(e " ent.
tab(eSty(e*(a	nu((Strin,	Sty(e c(a o+ the tab(e e(e " ent.
e " &ty 8 e a,e	<o record +ound	Strin,	9e0t to di &(ay ' hen there i no data to di &(ay.
re i=ab(e*o(u " n	+a(e	1oo(ean	De+ine i+ co(u " can be re i=ed or not.
ro ' Sty(e*(a	nu((Strin,	Sty(e c(a +or each ro '.
(iveRe i=e	+a(e	1oo(ean	*o(u " n are re i=ed (ive in thi " ode ' ithout u in, a re i=e he(&er.
reCured	+a(e	1oo(ean	>a(idation con traint +or e(ection.
reCured 8 e a,e	nu((Strin,	8 e a,e +or reCured e(ection va(idation.
ort1y	nu((>a(ueE0&r	E0&re ion +or de+au(t ortin,.
ort7rder	a cendin,	Strin,	De+ine de+au(t ortin, order.
ort#unction	nu((8 ethodE0&r	*u to " &(u ,ab(e ort#unction +or de+au(t ortin,.
nativeE(e " ent	+a(e	1oo(ean	In native " ode E treeTab(e u e native chec3bo0e .
data6oca(e	nu((7b/ect	6oca(e to be u ed in +eature such a ortin,E de+au(t to vie ' (oca(e.
ca eSen itiveSort	+a(e	1oo(ean	*a e en itivity +or ortin,E in en itive by de+au(t.

Getting started with the TreeTable

Si " i(ar to the 9reeE 9ree9ab(e i &o&u(ated ' ith an *org.primefaces.model.TreeNode* in tance that corre &ond to the root node. 9ree<ode A !I ha a hierarchica(data tructure and re&re ent the data to be &o&u(ated in tree. #or an e0a " &(eE " ode(to be di &(ayed i a co((ection o+ docu " ent i " i(ar a in tree ection.

```
public class Document {

    private String name;
    private String size;
    private String type;
    //getters, setters
}
```

```
<p:treeTable value="#{bean.root}" var="document">
    <p:column>
        <f:facet name="header">
            Name
        </f:facet>
        <h:outputText value="#{document.name}" />
    </p:column>

    //more columns
</p:treeTable>
```

Selection

`ode` e(ection i a bui(tBin +eature o+ tree and it u&&ort t 'o di+erent "ode . Se(ection hou(d be a 9ree<ode +or in,(e ca e and an array o+ 9ree<ode +or " u(ti&(e ca e) tree +ind the e(ected node and a i,n the " to your e(ection "ode).

`single` 7n(y one at a ti " e can be e(ected) e(ection hou(d be a 9ree<ode re+erence.

`multiple` or `checkbox` 8 u(ti&(e node can be e(ected) e(ection hou(d be a 9ree<ode`a re+erence.

A chec3bo0 e(ection have a &ecia(hierarchy u e *CheckboxTreeNode* in chec3bo0 "ode.

Ajax Behavior Events

9ree9ab(e &rovide variou a/a0 behavior event to re &ond u er action .

Event	Listener Parameter	Fired
e0&and	or,.&ri " e+ace .event.<odeE0&andEvent	When a node i e0&anded.
co((a& e	or,.&ri " e+ace .event.<ode*o((a& eEvent	When a node i co((a& ed.
e(ect	or,.&ri " e+ace .event.<odeSe(ectEvent	When a node i e(ected.
un e(ect	or,.&ri " e+ace .event.<odeUn e(ectEvent	When a node i un e(ected.
co(Re i=e	or,.&ri " e+ace .event.*o(u " nRe i=eEvent	When a co(u " n i re i=ed.

ContextMenu

9ree9ab(e ha &ecia(inte ,ration ' ith conte0t " enu you can even " atch di+erent conte0t " enu ' ith di+erent tree node u in, *nodeType* o&tion o+ conte0t " enu that " atche the tree node ty&e.

Scrolling

Scro((ab(e 9ree9ab(e i " &(e " entation i a" e a Data9ab(e Scro((ab(e re+er to cro((in, &art in Data9ab(e ection +or detai(ed in+or " ation.

Sorting

Sortin, i enab(ed by ettin, *sortBy* e0&re ion at co(u " n (eve(.

```
<p:treeTable value="#{bean.root}" var="document">
    <p:column sortBy="#{document.name}">
        <h:outputText value="#{document.name}" />
    </p:column>

    //more columns
</p:treeTable>
```

In ca e yould (i3e to di &(ay tree9ab(e a orted on &a,e (oad u e ort1y attribute o+ tree9ab(e o&tiona(*sortOrder* and *sortFunction* attribute are &rovided to de+ine the de+au(t ort order ?a cendin, or de cindin, l and a /ava " ethod to do the actua(ortin, re &ective(y. Re+er to databab(e ortin, ection +or an e0a " &(e u a,e o+ *sortFunction*.

Skinning

9ree9ab(e content re ide in a container e(e " ent ' hich ty(e and ty(e*(a attribute a&&(y. #o((o ' in, i the (i t o+ tructura(ty(e c(a e K

Class	Applies
.uiBreetab(e	8 ain container e(e " ent.
.uiBreetab(eBheader	. eader o+ treetab(e.
.uiBreetab(eBdata	1ody e(e " ent o+ the tab(e containin , data

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

3.136 Watermark

Watermark is a hint on an input field.

Info

9a,	watermark
*o " &onent *(a	org.primefaces.component.watermark.Watermark
*o " &onent 9y&e	org.primefaces.component.Watermark
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.WatermarkRenderer
Renderer *(a	org.primefaces.component.watermark.WatermarkRenderer

Attributes

Name	Default	Type	Description
id	nu((String,	Unique identifier of the component.
rendered	true	Boolean	Boolean value to specify the rendering of the component. If set to false, the component will not be rendered.
binding	nu((Object	An expression that refers to a server-side UI component in a backing bean.
value	nu((Object	Default value of the input field.
for	nu((String,	Identifier to attach the watermark.

Getting started with Watermark

Watermark requires a target element or the input component. In case you don't need to support legacy browsers, consider using the attribute `placeholder` over `watermark`.

```
<h:inputText id="txt" value="#{bean.searchKeyword}" />
<p:watermark for="txt" value="Search with a keyword" />
```

Form Submissions

Water "ar3 i et a the te0t o+ an in&ut +ie(d 'hich hou(dn't be ent to the erver 'hen an enc(o in, +or i ub" itted. 9hi 'ou(d re u(t in u&datin, bean &ro&ertie 'ith 'ater" ar3 va(ue . Water"ar3 co "&onent i c(ever enou,h to hand(e thi ca eE by de+au(t in nonBa/a0 +or" ub" i ion E 'ater"ar3 are c(eared. . o 'ever a/a0 ub" i ion reCuire a(itt(e "anua(e+ort.

!(ea e note that thi on(y a&&(ie to (e,acy bro ' er E a 'ater" ar3 u e . 9 8 65 &(aceho(der o&tion ' hen avai(ab(e.

```
<h:inputText id="txt" value="#{bean.searchKeyword}" />
<p:watermark for="txt" value="Search with a keyword" />
<p:commandButton value="Submit" onclick="PrimeFaces.cleanWatermarks()"
    oncomplete="PrimeFaces.showWatermarks()" />
```

Skinning

#or bro ' er that do not u&&ort &(aceho(der there' on(y one c ty(e c(a a&&(yin, 'ater" ar3 'hich i '.ui-watermark' you can override thi c(a to brin, in your o 'n ty(e.

3.137 Wizard

Wi=ard &rovide an a/a0 enhanced UI to i " &(e " ent a ' or3+(o' ea i(y in a in,(e &a,e. Wi=ard con i t o+ evera(chi(d tab co " &onent ' here each tab re&re ent a te& in the &roce .

The screenshot shows a wizard component with four tabs: Personal, Address, Contact, and Confirmation. The Personal tab is selected and active. Inside the Personal tab, there is a panel titled 'Personal Details' containing four input fields: 'Firstname' (with a required asterisk), 'Lastname' (with a required asterisk), 'Age' (with a required asterisk), and a 'Skip to last' checkbox. At the bottom right of the panel is a 'Next' button.

Info

9a,	wizard
*o " &onent *(a	org.primefaces.component.wizard.Wizard
*o " &onent 9y&e	org.primefaces.component.Wizard
*o " &onent #a " i(y	org.primefaces.component
Renderer 9y&e	org.primefaces.component.WizardRenderer
Renderer *(a	org.primefaces.component.wizard.WizardRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent.
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onentE ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
te&	0	Strin ,	Id o+ the current te& in +(o '
ty(e	nu((Strin ,	Sty(e o+ the " ain ' i=ard container e(e " ent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the " ain ' i=ard container e(e " ent.
+o ' 6i tener	nu((8 ethodE0&r	Server ide (i tener to invo3e ' hen ' i=ard atte " &t to ,o +or ' ard or bac3.
ho ' <av1ar	true	1oo(ean	S&eci+ie vi ibi(ity o+ de+au(t navi ,ator arro ' .
ho ' Ste&Statu	true	1oo(ean	S&eci+ie vi ibi(ity o+ de+au(t te& tit(e bar.

Name	Default	Type	Description
onbac3	nu((Strin,	Aava cri&t event hand(er to be invo3ed ' hen +(o' ,oe bac3.
onneot	nu((Strin,	Aava cri&t event hand(er to be invo3ed ' hen +(o' ,oe +or ' ard.
ne0t6abe(nu((Strin,	6abe(o+ ne0t navi ,ation button.
bac36abe(nu((Strin,	6abe(o+ bac3 navi ,ation button.
' id,et>ar	nu((Strin,	<a " e o+ the c(ient ide ' id,et

Getting Started with Wizard

Each te& in the +(o ' i re&re ented ' ith a tab. A an e0a "&(e +o((o ' in, ' i=ard i u ed to create a ne ' u er in a tota(o+ - te& ' here (a t te& i +or con+ir " ation o+ the in+or " ation &rovided in +ir t) te&. 9o be,in ' ith create your bac3in, beanE it' i " &ortant that the bean (ive acro " u(ti&(e reCue t o avoid a reCue t co&e bean. 7&ti " a(co&e +or ' i=ard i vie ' Sco&e.

```
public class UserWizard {

    private User user = new User();

    public User getUser() {
        return user;
    }

    public void setUser(User user) {
        this.user = user;
    }

    public void save(ActionEvent actionEvent) {
        //Persist user
        FacesMessage msg = new FacesMessage("Successful",
            "Welcome :" + user.getFirstname());
        FacesContext.getCurrentInstance().addMessage(null, msg);
    }
}
```

User i a i "&(e &o/o ' ith &ro&ertie uch a +ir tna " eE(a tna " elE e " ai(and etc. #o((o ' in, ' i=ard reCuire) te& to ,et the u er data!er ona(Detai(E Addre Detai(and *ontact Detai(. <ote that (a t tab contain readBony data +or con+ir " ation and the ub " it button.

```

<h:form>

    <p:wizard>
        <p:tab id="personal">
            <p:panel header="Personal Details">

                <h:messages errorClass="error"/>

                <h:panelGrid columns="2">
                    <h:outputText value="Firstname: *" />
                    <h:inputText value="#{userWizard.user.firstname}" required="true"/>

                    <h:outputText value="Lastname: *" />
                    <h:inputText value="#{userWizard.user.lastname}" required="true"/>

                    <h:outputText value="Age: " />
                    <h:inputText value="#{userWizard.user.age}" />
                </h:panelGrid>
            </p:panel>
        </p:tab>

        <p:tab id="address">
            <p:panel header="Address Details">

                <h:messages errorClass="error"/>

                <h:panelGrid columns="2" columnClasses="label, value">
                    <h:outputText value="Street: " />
                    <h:inputText value="#{userWizard.user.street}" />

                    <h:outputText value="Postal Code: " />
                    <h:inputText value="#{userWizard.user.postalCode}" />

                    <h:outputText value="City: " />
                    <h:inputText value="#{userWizard.user.city}" />
                </h:panelGrid>
            </p:panel>
        </p:tab>

        <p:tab id="contact">
            <p:panel header="Contact Information">

                <h:messages errorClass="error"/>

                <h:panelGrid columns="2">
                    <h:outputText value="Email: *" />
                    <h:inputText value="#{userWizard.user.email}" required="true"/>

                    <h:outputText value="Phone: " />
                    <h:inputText value="#{userWizard.user.phone}"/>

                    <h:outputText value="Additional Info: " />
                    <h:inputText value="#{userWizard.user.info}"/>
                </h:panelGrid>
            </p:panel>
        </p:tab>
    </p:wizard>

```

```

<p:tab id="confirm">
    <p:panel header="Confirmation">

        <h:panelGrid id="confirmation" columns="6">
            <h:outputText value="Firstname: " />
            <h:outputText value="#{userWizard.user.firstname}" />

            <h:outputText value="Lastname: " />
            <h:outputText value="#{userWizard.user.lastname}" />

            <h:outputText value="Age: " />
            <h:outputText value="#{userWizard.user.age}" />

            <h:outputText value="Street: " />
            <h:outputText value="#{userWizard.user.street}" />

            <h:outputText value="Postal Code: " />
            <h:outputText value="#{userWizard.user.postalCode}" />

            <h:outputText value="City: " />
            <h:outputText value="#{userWizard.user.city}" />

            <h:outputText value="Email: " />
            <h:outputText value="#{userWizard.user.email}" />

            <h:outputText value="Phone " />
            <h:outputText value="#{userWizard.user.phone}" />

            <h:outputText value="Info: " />
            <h:outputText value="#{userWizard.user.info}" />

            <h:outputText />
            <h:outputText />
        </h:panelGrid>

        <p:commandButton value="Submit" actionListener="#{userWizard.save}" />
    </p:panel>
</p:tab>

</p:wizard>
</h:form>

```

AJAX and Partial Validations

S'itchin, bet'een te& i ba ed on a/aOE "eanin, each te& i (oaded dyna" ica((y 'ith a/a0. !artia(va(idation i a(o bui(tBInE by thi 'ay 'hen you c(ic3 ne0tE on(y the current te& i va(idatedE i+ the current te& i va(idE ne0t tab' content are (oaded 'ith a/a0. >a(idation are not e0ecuted 'hen +(o ' ,oe bac3.

Navigations

Wi=ard &rovide t' o icon to interact ' ithK ne0t and &rev. !(ea e ee the 3innin, ' i=ard ection to 3no ' " ore about ho ' to chan,e the (oo3 and +ee(o+ a ' i=ard.

Custom UI

If you'd like to hide the navigation bar and provide your own, set `showNavBar="false"` and use the provided client-side API.

```
<p:wizard showNavBar="false" widgetVar="wiz">
    ...
</p:wizard>

<h:outputLink value="#" onclick="PF('wiz').next();">Next</h:outputLink>
<h:outputLink value="#" onclick="PF('wiz').back();">Back</h:outputLink>
```

FlowListener

If you'd like to be notified on server side when a card is added or removed, define a flow listener.

```
<p:wizard flowListener="#{userWizard.handleFlow}">
    ...
</p:wizard>
```

```
public String handleFlow(FlowEvent event) {
    String currentStepId = event.getCurrentStep();
    String stepToGo = event.getNextStep();

    if(skip)
        return "confirm";
    else
        return event.getNextStep();
}
```

Steps here are defined by the `onTab` attribute. If you need to decide which tab to display based on other conditions, you can update the navigation bar using `RequestContext.update(String clientId)`.

Client Side Callbacks

With each card added or removed, the `onTab` attribute is called. You need to execute code to handle this event. You don't need to provide the name of the `onTab` function, just the value of the `on` attribute.

```
<p:wizard onnext="alert('Next')" onback="alert('Back')">
    ...
</p:wizard>
```

Client Side API

The `PrimeFaces.widget.Wizard`

Method	Params	Return Type	Description
ne0t?@	B	void	Proceed to ne0t te&.
bac3?@	B	void	Go bac3 in +(o ' .
,etSte&Inde0?@	B	<u " ber	Return the inde0 o+ current te&.
ho ' <e0t<av?@	B	void	Show ne0t button.
hide<e0t<av?@	B	void	. ide ne0t button.
ho ' 1ac3<av?@	B	void	Show bac3 button.
hide1ac3<av?@	B	void	. ide bac3 button.

Skinning

Wi=ard re ide in a container e(e " ent that *style* and *styleClass* attribute a&&(y. #o((o ' in, i the (i t o+ tructura(c c(a e .

Selector	Applies
.uiB ' i=ard	8 ain container e(e " ent.
.uiB ' i=ardBcontent	*ontainer e(e " ent o+ content.
.uiB ' i=ardB te&Btit(e	*ontainer o+ te& tit(e .
.uiB ' i=ardB te&Btit(e	Each te& tit(e.
.uiB ' i=ardBnavbar	*ontainer o+ navi ,ation contro(.
.uiB ' i=ardBnavBbac3	1ac3 navi ,ation contro(.
.uiB ' i=ardBnavBne0t	#or ' ard navi ,ation contro(.

A 3innin, ty(e c(a e are ,(oba(E ee the " ain the " in, ection +or " ore in+or " ation.

4. Partial Rendering and Processing

!ri "e#ace &rovide a &artia(renderin, and vie ' &roce in, +eature ba ed on standard AS# % A!I to enab(e choo in, 'hat to &roce in AS# (i+ecy(e and 'hat to render in the end 'ith a/a0.

4.1 Partial Rendering

In addition to co " &onent (i3e auto*o " &(eteE databab(eE (ider ' ith bui(tB in a/a0 ca&abi(itie E !ri "e#ace a(o &rovide a ,eneric !!R ?!artia(!a,e Renderin,@ " echani " to u&date AS# co " &onent ' ith a/a0. Severa(co " &onent are eCui&&ed ' ith the co " "on !!R attribute ?e.,. u&dateE &roce E on tartE onco " &(ete@.

4.1.1 Infrastructure

!ri "e#ace A/a0 #ra "e ' or3 i ba ed on standard erver ide A!I o+ AS# %. 9here are no additiona(art+act (i3e cu to " A/a0>ie ' RootE A/a0State 8 ana,erE A/a0>ie ' . and(erE Serv(et #i(ter E .t"(!ar er E !ha e6i tener and o on. !ri "e#ace ai " to 3ee& it c(eanE +a t and (i,ht ' ei, ht.

7n c(ient ide rather than u in, c(ient ide A!I i " &(e " entation o+ AS# i " &(e " entation (i3e 8 o/arra and 8 y#ace E !ri "e#ace cri&t are ba ed on the "o t &o&u(ar /ava cri&t (ibraryE /Duery ' hich i +ar "ore te tedE tab(e re,ardin, a/aOE do " hand(in,E do " tree traver in, than a AS# i " &(e " entation cri&t .

4.1.2 Using IDs

Getting Started

When u in, !!R you need to &eci+y ' hich co " &onent? @ to u&date ' ith a/a0. If the co " &onent that tri,,er !!R reCue t i at the a"e na " in,container ?e.,. +or "@ ' ith the co " &onent? @ it render E you can u e the erver id direct(y. In thi ection a(thou,h ' e'((be u in, co " " and1uttonE a"e a&&(ie to every co " &onent that' ca&ab(e o+ !!R uch a co " "and6in3E &o((E re " ote*o " " and and etc.

```
<h:form>
    <p:commandButton update="display" />
    <h:outputText id="display" value="#{bean.value}" />
</h:form>
```

PrependId

Settin, &re&endId ettin, o+ a +or " ha no e+ect on ho ' !!R i u ed.

```
<h:form prependId="false">
    <p:commandButton update="display" />
    <h:outputText id="display" value="#{bean.value}" />
</h:form>
```

ClientId

It is also possible to define the client id of the component to update.

```
<h:form id="myform">
    <p:commandButton update="myform:display" />
    <h:outputText id="display" value="#{bean.value}" />
</h:form>
```

Different NamingContainers

If your application has different naming containers (e.g., tabs or accordions) you also need to add the container id to each element so that PrimeFaces can handle regions that are triggered in different naming containers that update another naming container. Note that this uses a search algorithm standard AS# is "entation".

```
<h:form id="form1">
    <p:commandButton update=":form2:display" />
</h:form>

<h:form id="form2">
    <h:outputText id="display" value="#{bean.value}" />
</h:form>
```

(See also **findComponent** which is described in the Java EE 6 API documentation and is supported by both AS# core and PrimeFaces to update under the same component reference.)

<http://docs.oracle.com/javaee/6/api/javax/faces/component/UIComponent.html>

AS# provides a databinding component whose elements are naming containers in addition to tabs, accordions, data grids, data tables, carousels, accordions, sheets and tabs. PrimeFaces supports this component to update components that are contained within a naming container.

Multiple Components

Updating multiple components can be achieved either by providing a single update attribute separated by commas or by using both.

```
<h:form>
    <p:commandButton update="display1,display2" />
    <p:commandButton update="display1 display2" />
    <h:outputText id="display1" value="#{bean.value1}" />
    <h:outputText id="display2" value="#{bean.value2}" />
</h:form>
```

4.1.3 Notifying Users

By default the component notify the user about the status of **global** and **reCue**. See the `action` section to learn more information about the component.

Global vs Non-Global

If you want to do a silent reCue not to trigger another status in tandem with, then set `global="false"` with both and button followed by:

```
<p:commandButton value="Silent" global="false" />
<p:commandButton value="Notify" global="true" />
```

4.1.4 Bits&Pieces

PrimeFaces Ajax Javascript API

See the `oncomplete` section to learn more about the PrimeFaces Ajax API.

4.2 Partial Processing

In `<h:form>` only defined components are rendered initially. Only defined components are rendered initially. In, only defined components are rendered initially. In, only defined components are rendered initially. In, only defined components are rendered initially.

High feature is that you can do this without using `value="#{bean.city}"`. Instead, you can use `value="#{bean.cityChoices}"`. This will avoid initial rendering of the city dropdown. Instead, it will be populated via Ajax when the user selects a city. This is useful for performance reasons and also allows for better user experience by avoiding unnecessary initial rendering.

4.2.1 Partial Validation

A component can be validated independently of the rest of the form. You have a contact person whose city and suburb are required. When the city is selected, the suburb dropdown is populated with the corresponding city's suburbs. This is achieved using the `process="@all"` attribute on the `p:ajax` listener.

```
<h:form>

    <h:selectOneMenu id="cities" value="#{bean.city}">
        <f:selectItems value="#{bean.cityChoices}" />
        <p:ajax listener="#{bean.populateSuburbs}" update="suburbs"
            process="@all"/>
    </h:selectOneMenu>

    <h:selectOneMenu id="suburbs" value="#{bean.suburb}">
        <f:selectItems value="#{bean.suburbChoices}" />
    </h:selectOneMenu>

    <h:inputText value="#{bean.email}" required="true"/>

</h:form>
```

When the city dropdown is changed, an ajax request is sent to execute `populateSuburbs` method. This method updates the suburb dropdown with the corresponding suburbs for the selected city. If the city is not selected, the suburb dropdown is not populated. This is achieved using the `process="@all"` attribute on the `p:ajax` listener.

The solution is to define what to validate in `value="#{bean.city}"`. A primefaces component can be rendered only if its value is valid. This means that the city dropdown will only be rendered if the city is valid. This is useful for performance reasons and also allows for better user experience by avoiding unnecessary initial rendering.

```

<h:form>
    <h:selectOneMenu id="cities" value="#{bean.city}">
        <f:selectItems value="#{bean.cityChoices}" />
        <p:ajax actionListener="#{bean.populateSuburbs}"
            event="change" update="suburbs" process="@this"/>
    </h:selectOneMenu>

    <h:selectOneMenu id="suburbs" value="#{bean.suburb}">
        <f:selectItems value="#{bean.suburbChoices}" />
    </h:selectOneMenu>

    <h:inputText value="#{bean.email}" required="true"/>
</h:form>

```

9hat i it no ' &o&u(ateSuburb " ethod ' i((be ca((ed and uburb (i t ' i((be &o&u(ated. <ote that de+au(t va(ue +or &roce o&tion i b thi a(ready +or &Na/a0 a tated in A/a01ehavior docu " entation& it i e0&(icit(y de+ined here to ,ive a better under tandin, o+ ho ' &artia(&roce in, ' or3 .

4.2.2 Using Ids

!artia(!roce u e the a " e techniCue a&&(ied in &artia(u&date to &eci+y co " &onent identi+ier to &roce .

4.3 Search Expression Framework

*ore AS# co " &onent re+erencin, i ba ed on co " &onent identi+ier on(y ' ith ba ic 3ey ' ord u&&ort. !ri " e#ace Search E0&re ion #ra " e ' or3 ?SE#@ &rovide both erver ide and c(ient ide e0ten ion to " a3e it ea ier to re+erence co " &onent . SE# i uti(i=ed in &artia(u&dateE &roce and ' henever a co " &onent re+erence another co " &onent.

4.3.1 Keywords

; ey ' ord are the ea ier ' ay to re+erence co " &onent E they re o(ve to id o that i+ an id chan,e E the re+erence doe not need to chan,e. *ore AS# &rovide a cou&(e o+ 3ey ' ord and !ri " e#ace &rovide " ore a(on, ' ith co " &o ite e0&re ion u&&ort.

Keyword	Type	Description
b thi	Standard	*urrent co " &onent.
b a((Standard	Who(e vie ' .
b +or "	Standard	*(o e t ance tor +or " o+ current co " &onent.
b none	Standard	<o co " &onent.
b na " in, container	!ri " e#ace	*(o e t ance tor na " in, container o+ current co " &onent.
b &parent	!ri " e#ace	!arent o+ the current co " &onent.
b co " &o ite	!ri " e#ace	*(o e t co " &o ite co " &onent ance tor.
b chi(d?n@	!ri " e#ace	nth chi(d.
b &reviou	!ri " e#ace	!reviou ib(in,..
b ne0t	!ri " e#ace	<e0t ib(in,..
b ' id,et>ar?na" e@	!ri " e#ace	*o " &onent ' ith ,iven ' id,et>ar.

*on ider the +o((o ' in, ca e ' here id are u ed +or re+erencin,K

```
<h:form id="form1">
    <p:commandButton id="btn" update="form1" process="btn" />
    <h:outputText value="#{bean.value}" />
</h:form>
```

U in, 3ey ' ord E a" e can be ' ritten a K

```
<h:form id="form1">
    <p:commandButton id="btn" update="@form" process="@this" />
    <h:outputText value="#{bean.value}" />
</h:form>
```

Composite Expressions

8 u(ti&(e 3ey ' ord can be co " bined in a in,(e e0&re ion u in, co(on&
G b +or " N b &arent
G b co "&o item" ybuttonid
G b thi N b &arentN b &arent
G b +or " N b chi(d?%@

Usage Scenarios

SE# i not /u t at &artia(&roce and u&dateE they are a(o avai(ab(e ' henever a co "&onent i
re+erencin, another.

```
<h:form>
    <p:commandButton id="dynaButton" value="Show" type="button" />
    <p:menu overlay="true" trigger="@parent:dynaButton">
        //items
    </p:menu>
</h:form>
```

4.3.2 PrimeFaces Selectors (PFS)

!#S inte,rate /Duery Se(ector A!I ' ith AS# co "&onent re+erencin, " ode(o that re+erencin, can
be done u in, /Duery Se(ector A!I in tead o+ core id ba ed AS# " ode(. 1e t ' ay to e0&(ain the
&o ' er o+ !#S i e0a " &(e K

U&date a((+or "

```
update="@(@(form))"
```

U&date +ir t +or "

```
update="@(@(form:first))"
```

U&date a((co "&onent that ha ty(e*(a na " ed " y ty(e

```
update="@(@.mystyle))"
```

U&date and &roce a((in&ut

```
update="@(:input)" process="@(:input))"
```

Update a(datatab(e

```
update="@(.ui-datatable)"
```

!roce in&ut co " &onent in ide any &ane(and u&date a(&ane(

```
process="@(.ui-panel :input)" update="@(.ui-panel)"
```

!roce in&ut co " &onent but not e(ect co " &onent

```
process="@(:input:not(select))"
```

U&date in&ut co " &onent that are di ab(ed

```
update="@(:input:disabled)"
```

!#S can be u ed ' ith other re+erencin , a&&roache a ' e((

```
update="compId :form:compId @(:input) @parent:@child(2)"
```

```
<h:form>
  <p:commandButton id="dynaButton" value="Show" type="button" styleClass="btn"/>
  <p:menu overlay="true" trigger="@(.btn)">
    //items
  </p:menu>
</h:form>
```

!#S &rovide an a(ternativeE+(e0ib(eE ,rou&in , ba ed a&&roach to re+erence co " &onent to &artia((y &roce and u&date. 9here i (e * !U erver (oad co " &ared to re ,u(ar re+erencin , becaue AS# co " &onent tree i not traver ed on erver ide to +ind a co " &onent and +i ,ure out the c(ient id a !#S i i " &(e " ented on c(ient ide by (oo3in , at do " tree. Another advanta,e i avoidin , na " in , container (i " itation E/u t re " e " ber the ti " e you've +aced ' ith cannot +ind co " &onent e0ce&tion ince the co " &onent you are (oo3in , +or i in a di+erent na " in , container (i3e a +or " or a datatab(e. !#S can he(& you out in tric3y ituation by +o((o ' in , /Duery' " ' rite (e do " ore" ty(e.

#or !#S to +unction &ro&er(y and not to " i any co " &onentE it i reCquired to have e0&(icit(y de+ined id on the " atched et a core AS# co " &onent u ua((y do not render auto id . So even thou,h " anua((y de+ined id ' on't be re+erenced direct(yE they are ti((reCquired +or !#S to be co(ected and end in the reCue t.

#or +u((re+erence o+ /Duery e(ector a&iE eeK

```
http://api.jquery.com/category/selectors/
```

4.4 PartialSubmit

*ore AS# A/a0 i " &(e " entation and by de+au(t !ri " e#ace eria(i=e the ' ho(e +or" to bui(d the &o t data in a/a0 reCue t o the a"e data i &o ted /u t (i3e in a nonBa/a0 reCue t. 9hi ha a do 'n ide in (ar,e vie ' ' here you on(y need to &roce (e0ecute a " inor &art o+ the vie '. A u"e you have a +or" ' ith 100 in&ut +ie(d E there i an in&ut +ie(d ' ith a/a0behavior attached &roce in, on(y it e(+? b thi @ and then u&date another +ie(d onb(ur. A(thou,h on(y a &articu(ar in&ut +ie(d i &roce edE ' ho(e +or" data ' i((be &o ted ' ith the unnece ary in+or" ation that ' ou(d be i,nored durin, erver ide &roce in, but con u"e re ource .

!ri " e#ace &rovide &artia(Sub " it +eature to reduce the net ' or3 tra+ic and co " &utin, on c(ient ide. When &artia(Sub " it i enab(edE on(y data o+ co " &onent that ' i((be &artia((y &roce ed on the erver ide are eria(i=ed. 1y de+au(t &artia(Sub " it i di ab(ed and you can enab(e it ,(oba((y u in, a conteot &ara " eter.

```
<context-param>
    <param-name>primefaces.SUBMIT</param-name>
    <param-value>partial</param-value>
</context-param>
```

*o " &onent (i3e button and behavior (i3e &Na/a0 are eCui&&ed ' ith &artia(Sub " it o&tion o you can override the ,(oba(ettin, &er co " &onent.

```
<p:commandButton value="Submit" partialSubmit="true|false" />
```

5. Javascript API

This chapter describes the PrimeFaces Javascript API, which provides a set of methods for interacting with PrimeFaces components and their properties.

5.1 PrimeFaces Namespace

PrimeFaces is the main Java class that provides the API for interacting with PrimeFaces components.

Method	Description
<code>e.ca(&#39;id?&#39;)</code>	Creates a new instance of the PrimeFaces component with the specified ID.
<code>addSubComponent("name")</code>	Adds a new subcomponent with the specified name to the current component.
<code>,et("name")</code>	Returns the component with the specified name.
<code>et("name") = value</code>	Sets the value of the component with the specified name.
<code>enhance()</code>	Enhanceses the component by applying PrimeFaces styles and behaviors.
<code>isDebugEnabled(), isWarn(), isError()</code>	Checks if debug, warn, or error logging is enabled.
<code>cancel()</code>	Cancels the component's execution.
<code>cleanWatermark()</code>	Cleans the watermark of the component.
<code>showWatermark()</code>	Shows the watermark of the component.

Some components contain children components, such as `DataTable`, `Tree`, etc. These components have their own methods and properties.

PrimeFaces.widget.*

Contains methods for interacting with widget components.

```
B !ri "e#ace . ' id, et.DataTable
B !ri "e#ace . ' id, et.Tree
B !ri "e#ace . ' id, et.Overlay
B and "ore...
```

Some components have a corresponding container component, such as `Table`, `Form`, etc.

PrimeFaces.ajax.*

Contains methods for performing Ajax requests and handling responses.

5.2 Ajax API

!ri " e#ace A/a0 Aava cri&t A !I i &o ' ered by /Duery and o&ti " i=ed +or AS#. Who(e A !I con i t o+ three &ro&er(y na " e &aced i " &(e /ava cri&t +unction .

PrimeFaces.ajax.Request

Send a/a0 reCue t that e0ecute AS# (i+ecyc(e and retrieve &artia(out&ut. #unction i, nature i a +o((o ' K

```
PrimeFaces.ajax.Request.handle(cfg);
```

Configuration Options

Option	Description
+or " Id	Id o+ the +or " e(e " ent to eria(i=eE i+ not de+ined &arent +or " o+ ource i u ed.
a ync	#(a, to de+ine ' hether reCue t hou(d ,o in a/a0 Cueue or notE de+au(t i +a(e.
,(oba(#(a, to de+ine i+ &Na/a0Statu hou(d be tri, ,ered or notE de+au(t i true.
u&date	*o " &onent? @ to u&date ' ith a/a0.
&roce	*o " &onent? @ to &roce in &artia(reCue t.
ource	*(ient id o+ the ource co " &onent cau in, the reCue t.
&ara "	Additiona(&ara " eter to end in a/a0 reCue t.
on tart?@	Aava cri&t ca((bac3 to &roce be+ore endin, the a/a0 reCue t@ return +a(e to cancel the reCue t.
on ucce ?dataE tatu E OhrE ar, @	Aava cri&t ca((bac3 to &roce ' hen a/a0 reCue t return ' ith ucce code. 9a3e +our ar,u " ent E 0 "(re &on eE tatu codeE 0 "(htt&reCue t and o&tiona(ar,u " ent &rovvided by ReCue t*ontent A !I.
onerror?OhrE tatu E eOce&tion@	Aava cri&t ca((bac3 to &roce ' hen a/a0 reCue t +ai(. 9a3e three ar,u " ent E 0 "(htt&reCue t@ tatu trin, and eOce&tion thro ' n i+ any.
onco " &(ete?OhrE tatu E ar, @	Aava cri&t ca((bac3 to &roce ' hen a/a0 reCue t co " &(ete . 9a3e three ar,u " ent E 0 "(htt&reCue t@ tatu trin, and o&tiona(ar,u " ent &rovvided by ReCue t*onte0t A !I.

Examples

Suppose you have a JSF page containing `createUser` method:

```
<h:form id="userForm">
    <h:inputText id="username" value="#{userBean.user.name}" />
    ...
</h:form>
```

You can trigger the `createUser` action with a button like this:

```
PrimeFaces.ajax.Request.handle({
    formId:'userForm',
    source:'userForm',
    process:'userForm'
});
```

For example, with additional options:

```
PrimeFaces.ajax.Request.handle({
    formId: 'userForm',
    source: 'userForm',
    process: 'userForm',
    update: 'msgs',
    params:{
        'param_name1':'value1',
        'param_name2':'value2'
    },
    oncomplete:function(xhr, status) {alert('Done');}
});
```

We highlight the code that ends up in the browser's network tab. The code is generated by PrimeFaces to handle the action and generate an error message if there is one.

PrimeFaces.ajax.Response

The response object contains the updated state of the page. It includes the updated components and any errors or messages generated during the processing of the request.

6. Dialog Framework

Dia(o, #ra "e ' or3 ?D# i u ed to o&en an e0terna(Oht "(&a,e in a dia(o, that i ,enerated dyna " ica(y on runti " e. 9hi i Quite diifferent to re,u(ar u a,e o+ dia(o, ' ith dec(arative &Ndia(o, co " &onent a D# i ba ed on a &ro,ra " "atic A !I ' here dia(o, are created and de troyed at runti " e. <ote that D# and the dec(arative a&&roach are t' o diifferent 'ay and both can even be u ed to,ether. U a,e i Quite i "&(el ReCue t*onte0t ha o&enDia(o, and c/o eDia(o, " ethod K

```
/*
 * Open a view in dialog.
 * @param outcome The logical outcome used to resolve a navigation case.
 */
public abstract void openDialog(String outcome);

/**
 * Open a view in dialog.
 * @param outcome The logical outcome used to resolve a navigation case.
 * @param options Configuration options for the dialog.
 * @param params Parameters to send to the view displayed in a dialog.
 */
public abstract void openDialog(String outcome, Map<String, Object> options,
Map<String, List<String>> params);

/**
 * Close a dialog.
 * @param data Optional data to pass back to a dialogReturn event.
 */
public abstract void closeDialog(Object data);
```

Configuration

D# reCuire the +o((o ' in, con+i,uration to be &re ent in +ace con+i, +i(e.

```
<application>
    <action-listener>
        org.primefaces.application.DialogActionListener
    </action-listener>
    <navigation-handler>
        org.primefaces.application.DialogNavigationHandler
    </navigation-handler>
    <view-handler>
        org.primefaces.application.DialogViewHandler
    </view-handler>
</application>
```

Getting Started

Si " &e t u e ca e o+ D# i o&enin, an Oht "(vie ' (i3e *cars.xhtml* in a dia(o, K

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui">

    <h:head>
        <title>Cars</title>
    </h:head>

    <h:body>
        <p:dataTable var="car" value="#{tableBean.cars}">
            //columns
        </p:dataTable>
    </h:body>
</html>
```

In the header, the `Cars` title is defined and the `tableBean.cars` data source is specified.

```
<p:commandButton value="View Cars" actionListener="#{hostBean.view}" />
```

```
public void view() {
    RequestContext.getCurrentInstance().openDialog("viewCars");
}
```

Once the request is received (e.g., the button is clicked), the dialog will be generated with the content of the view. At this point, the dialog is retrieved (the title and content are retrieved).

Dialog Configuration

Overloaded `openDialog` method provides advanced configuration regarding the visual options of the dialog, such as its title and content.

```
<p:commandButton value="View Cars" actionListener="#{hostBean.viewCustomized}" />
```

```
public void view() {
    Map<String, Object> options = new HashMap<String, Object>();
    options.put("modal", true);
    options.put("draggable", false);
    options.put("resizable", false);
    options.put("contentHeight", 320);

    RequestContext.getCurrentInstance().openDialog("viewCars", options, null);
}
```

Here is the full configuration option list:

Name	Default	Type	Description
"oda(0	Boolean	*ontrols the visibility of the dialog.
re i=ab(e	1	Boolean	When enabled "a3e dialog, re i=ab(e.
dra,,ab(e	1	Boolean	When enabled "a3e dialog, dra,,ab(e.
'idth	auto	Integer	Width of the dialog.
hei,ht	auto	Integer	. ei, ht of the dialog.
contentWidth	400	Integer	Width of the dialog content.
content.ei,ht	auto	Integer	. ei, ht of the dialog content.
c(ab(e	true	Boolean	Whether the dialog can be closed or not.

Data Communication

In addition to the dialog, can also pass data back to the parent page. The trick is, however, to have a dialogReturn behavior event to handle data returned from the dialog.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html"
      xmlns:p="http://primefaces.org/ui">

    <h:head>
        <title>Cars</title>
    </h:head>

    <h:body>
        <p: dataTable var="car" value="#{tableBean.cars}">
            //columns
            <p: column headerText="Select">
                <p: commandButton icon="ui-icon-search"
                    actionListener="#{tableBean.selectCarFromDialog(car)}" />
            </p: column>
        </p: dataTable>
    </h:body>
</html>
```

```
public void selectCarFromDialog(Car car) {
    RequestContext.getCurrentInstance().closeDialog(car);
}
```

At the button that triggered the dialog, should have a dialogReturn event.

```
<p:commandButton value="View Cars" actionListener="#{hostBean.viewCars}">
    <p:ajax event="dialogReturn" listener="#{hostBean.handleReturn}" />
</p:commandButton>
```

```
public void view() {
    RequestContext.getCurrentInstance().openDialog("viewCars");
}

public void handleReturn(SelectEvent event) {
    Car car = (Car) event.getObject();
}
```

Remarks on Dialog Framework

- At the "o" ent^E &Nco " "and1utton and &Nco " "and6in3 u&&ort *dialogReturn*.
- G* <e ted dia(o, are not u&&orted.
- G* *a((to Dia(o,#ra "e ' or3 A !I ' ithin a nonBa/a0 are i ,nored.

Dialog Messages

Di &ayin, #ace 8e a,e in a Dia(o, i a co " "on ca e ' here a +ace "e a,e need to be added to the conte0t+ir tE dia(o, content containin, a "e a,e co "&onent need to be u&dated and +ina((y dia(o, ,et ho ' n ' ith c(ient ide a&i. D# ha a i" &(e uti(ity to by&a thi &roce by &rovidin, a hortcut

```
/***
 * Displays a message in a dialog.
 * @param message FacesMessage to be displayed.
 */
public abstract void showMessageInDialog(FacesMessage message);
```

U in, thi hortcut it i /u t one (ine to i " &(e " ent the a " e +unctiona(ity

```
<p:commandButton value="Show" actionListener="#{bean.save}" />
```

```
public void save() {
    //business logic
    RequestContext.getCurrentInstance().showMessageInDialog(new
    FacesMessage(FacesMessage.SEVERITY_INFO,
        "What we do in life", "Echoes in eternity."));}
}
```



7. Client Side Validation

!ri "e#ace *(ient Side >a(idation ?*S>@ #ra "e' or3 i the "o t co" &(ete and advanced *S> o(ution +or AavaServer #ace and Aava EE. *S> u&&ort +or AS# i not an ea y ta 3E it i not i" &(e a inte,ratin, a)rd &arty /ava cri&t &(u,in a AS# ha it o 'n (i-ecyc(eE conce&t (i3e conver ion and then va(idationE &artia(&roce in,E +ace "e a,e and "any "ore. Rea(*S> +or AS# hou(d be co" &atib(e 'ith erver ide i"&(e"entationE hou(d do 'hat AS# doe E o that u er do not e0&erience di+ference behavior on c(ient ide and erver ide.

- G *o" &atib(e ' ith Server Side I " &(e " entation.
 - G *onver ion and >a(idation ha&&en at c(ient ide.
 - G !artia(!roce : U&date u&&ort +or A/a0.
 - G I12n u&&ort a(on, ' ith co " &onent &eci+ic " e a,e .
 - G *(ient ide Renderer +or " e a,e co " &onent .
 - G Ea y to ' rite cu to" c(ient converter and va(idator .
 - G G(oba(or *o" &onent ba ed enab(e\ldi ab(e.
 - G Advanced 1ean >a(idation Inte , ration.
 - G 6itt(e +oot&rint u in, . 9 8 65.

7.1 Configuration

Non-Ajax

In non-Ajax cases validation and editing in the form are validated and "empty" and "empty" are updated in the form.

Ajax

Since the validation and update logic in the client side is enabled by the validation attribute that is used to define the validation logic at the server side, it is validated at the client side. If the validation attribute is defined only "empty" and "empty" in the updated part, it is rendered. Who performs the validation depends on the validation logic.

7.3 Events

Provide a behavior called client behavior to do in tandem validation in case you do not want to wait for the user to click in the form and hit the "Save" button. In this case, client behavior and custom event *S> for a particular component can run with event such as change, validate, keyup, etc.

```
<h:form>
    <p:panel header="Validate">
        <h:panelGrid columns="4" cellpadding="5">
            <h:outputLabel for="text" value="Text: (Change)" />
            <p:inputText id="text" value="#{validationBean.text}" required="true">
                <f:validateLength minimum="2" maximum="5" />
                <p:clientValidator />
            </p:inputText>
            <p:message for="text" display="icon" />
            <h:outputText value="#{validationBean.text}" />

            <h:outputLabel for="integer" value="Integer: (Keyup)" />
            <p:inputText id="integer" value="#{validationBean.integer}">
                <p:clientValidator event="keyup"/>
            </p:inputText>
            <p:message for="integer" display="icon" />
            <h:outputText value="#{validationBean.integer}" />
        </h:panelGrid>

        <p:commandButton value="Save" ajax="false" icon="ui-icon-check"
            validateClient="true"/>
    </p:panel>
</h:form>
```

7.4 Messages

Validation errors are displayed at the client side in the validation text or retrieved from the client side bundle and "empty" and "empty" are required for the display.

I18N

Default translations in English or the *S> "empty" and "empty" or other translations or to customize "empty" and "empty" messages need to be present at the client side you'd like to provide translation. For example, "empty" message on "empty" field ace locale is visible at http://code.google.com/p/primefaces/source/browse/trunk/primefaces/src/main/resources/pt_BR/primefaces/messages_pt_BR.properties.

Rendering

!ri "e#ace " e a,e co " &onent have c(ient ide renderer +or *S> u&&ortE the e are &N" e a,e&N" e a,e and &N,ro '(. *o "&onent o&tion (i3e ho ' Su " "aryE ho ' DetaiE ,(oba(7n(yE " ode are a((i " &(e " ented by c(ient ide renderer +or co " &atibi(ity.

7.5 Bean Validation

*S> ha bui(tBin inte,ration ' ith Bean >a(idation by va(idatin, the con traint de+ined ' ith annotation at c(ient ide.

```
<h:form>
    <p:growl />
    <h:panelGrid>
        <h:outputLabel for="name" value="Name:" />
        <p:inputText id="name" value="#{bean.name}" label="Name"/>
        <p:message for="name" />

        <h:outputLabel for="age" value="Age: (@Min(10) @Max(20))" />
        <p:inputText id="age" value="#{bean.age}" label="Age"/>
        <p:message for="age" />
    </h:panelGrid>
    <p:commandButton value="Save" validateClient="false" ajax="false" />
</h:form>
```

```
public class Bean {

    @Size(min=2,max=5)
    private String name;

    @Min(10) @Max(20)
    private Integer age;
}
```

All the standard constraint are u&&orted.

7.6 Extending CSV

U in, *S> A !I Eit i ea y to ' rite your o ' n cu to " converter and va(idator .

Email Validator with JSF

Mour cu to " va(idator " u t i " &(e " ent *(ient>a(idator interface to &rovide the c(ient va(idator id and the o&tiona(" etadata.

```

package org.primefaces.examples.validate;

import java.util.Map;
import java.util.regex.Pattern;
import javax.faces.application.FacesMessage;
import javax.faces.component.UIComponent;
import javax.faces.context.FacesContext;
import javax.faces.validator.FacesValidator;
import javax.faces.validator.Validator;
import javax.faces.validator.ValidatorException;
import org.primefaces.validate.ClientValidator;

@FacesValidator("custom.emailValidator")
public class EmailValidator implements Validator, ClientValidator {

    private Pattern pattern;

    private static final String EMAIL_PATTERN = "^[_A-Za-z0-9-\\]+(\\.[_A-Za-z0-9-]*@[\\w.][A-Za-z0-9-]+(\\.\\w[A-Za-z0-9-]*)*(\\.\\w{2,})$";

    public EmailValidator() {
        pattern = Pattern.compile(EMAIL_PATTERN);
    }

    public void validate(FacesContext context, UIComponent component, Object value)
            throws ValidatorException {
        if(value == null) {
            return;
        }

        if(!pattern.matcher(value.toString()).matches()) {
            throw new ValidatorException(new
FacesMessage(FacesMessage.SEVERITY_ERROR, "Validation Error",
                    value + " is not a valid email;"));
        }
    }

    public Map<String, Object> getMetadata() {
        return null;
    }

    public String getValidatorId() {
        return "custom.emailValidator";
    }
}

```

>a(idator i &(u,,edBn u in, the standard 'ay.

```

<h:form>
    <p:messages />
    <p:inputText id="email" value="#{bean.value}">
        <f:validator validatorId="custom.emailValidator" />
    </p:inputText>
    <p:message for="email" />
    <p:commandButton value="Save" validateClient="true" ajax="false"/>
</h:form>

```

If a client-side validation fails, the validator at client-side and continues it.

```
PrimeFaces.validator['custom.emailValidator'] = {

    pattern: /\S+@\S+/,

    validate: function(element, value) {
        //use element.data() to access validation metadata, in this case there is
        none.
        if(!this.pattern.test(value)) {
            throw {
                summary: 'Validation Error',
                detail: value + ' is not a valid email.'
            }
        }
    }
};
```

In order to use your validator "isEmail", you need to add metadata or extend the validator to include constraints. Client-side validation can be overridden by overriding the `getMetadata()` method by providing a "map" of validation rules. At client-side, the rules are accessed via `element.data(key)`.

```
public Map<String, Object> getMetadata() {
    Map<String, Object> data = new HashMap<String, Object>();
    data.put("data-prime", 10);
    return data;
}
```

```
validate: function(element, value) {
    var prime = element.data("prime"); //10

    //validate
}
```

Similarly, a client-side converter can be written by extending `ClientConverter` and overriding the `convertFunction(element, submittedValue)` method to return a value or object.

Email Validator with Bean Validation

Bean validation is often used here in an example of a bean validator.

```
//imports
import org.primefaces.validate.bean.ClientConstraint;

@Target({METHOD,FIELD,ANNOTATION_TYPE})
@Retention(RUNTIME)
@Constraint(validatedBy=EmailConstraintValidator.class)
@ClientConstraint(resolvedBy=EmailClientValidationConstraint.class)
@Documented
public @interface Email {
    String message() default "{org.primefaces.examples.primefaces}";
    Class<?>[] groups() default {};
    Class<? extends Payload>[] payload() default {};
}
```

b *on traint i the re ,u(ar va(idator +ro " lean >a(idation A !I and b *(ient*on raint i +ro " *S> A !I to re o(ve " etadata.

```
public class EmailConstraintValidator
    implements ConstraintValidator<Email, String> {

    private Pattern pattern;

    private static final String EMAIL_PATTERN = "^[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-
9-]+)*@"
                                                + "[A-Za-z0-9-]+(\\. [A-Za-z0-
9-]+)*(\\. [A-Za-z]{2,})$";

    public void initialize(Email a) {
        pattern = Pattern.compile(EMAIL_PATTERN);
    }

    public boolean isValid(String value, ConstraintValidatorContext cvc) {
        if(value == null)
            return true;
        else
            return pattern.matcher(value.toString()).matches();
    }

}
```

```
public class EmailClientValidationConstraint implements ClientValidationConstraint {

    public static final String MESSAGE_METADATA = "data-p-email-msg";

    public Map<String, Object> getMetadata(ConstraintDescriptor constraintDescriptor)
    {
        Map<String, Object> metadata = new HashMap<String, Object>();
        Map attrs = constraintDescriptor.getAttributes();
        Object message = attrs.get("message");
        if(message != null) {
            metadata.put(MESSAGE_METADATA, message);
        }

        return metadata;
    }

    public String getValidatorId() {
        return Email.class.getSimpleName();
    }

}
```

#ina(&art i i" &(e " entin, the c(ient ide va(idator

```
PrimeFaces.validator['Email'] = {

    pattern: /\S+@\S+/,
    MESSAGE_ID: 'org.primefaces.examples.validate.email.message',
    validate: function(element, value) {
        var vc = PrimeFaces.util.ValidationContext;

        if(!this.pattern.test(value)) {
            var msgStr = element.data('p-email-msg'),
                msg = msgStr ? {summary:msgStr, detail: msgStr} :
                vc.getMessage(this.MESSAGE_ID);

            throw msg;
        }
    }
};
```

Using a standard constraint

```
<h:form>
    <p:messages />
    <p:inputText id="email" value="#{bean.value}" />
    <p:message for="email" />
    <p:commandButton value="Save" validateClient="true" ajax="false"/>
</h:form>
```

```
public class Bean {

    @Email
    private String value;

    //getter-setter
}
```

8. Themes

!ri " e#ace i inte, rated ' ith &o ' er+u(9he " eRo((er *SS #ra " e ' or3. *urrent(y there are)OH &reB de i, ned the " e that you can &revie ' and do ' n(oad +ro " !ri " e#ace the " e ,a((ery.

<http://www.primefaces.org/themes.html>



8.1 Applying a Theme

A&&(yin, a the "e to your !ri "e#ace &ro/ect i very ea y. Each the "e i &ac3a,ed a a /ar +i(e& do 'n(oad the the "e you 'ant to u e& add it to the c(a &ath o+ your a&&(ication and then de+ine &ri "e+ace .9 . E 8 E conteOt &ara "eter at your de&(oy "ent de cri&tor ?' eb.0 "(@ 'ith the the "e na "e a the va(ue.

Download

Each the "e i avai(ab(e +or "anua(do 'n(oad at !ri "e#ace 9he "e Ga((ery. If you are a "aven u er& de+ine the "e arti+act a K

```
<dependency>
    <groupId>org.primefaces.themes</groupId>
    <artifactId>cupertino</artifactId>
    <version>1.0.8</version>
</dependency>
```

arti+actId i the na "e o+ the the "e a de+ined at 9he "e Ga((ery &a,e.

Configure

Once you've do 'n(oaded the the "e con+i,ure !ri "e#ace to u e it.

```
<context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>aristo</param-value>
</context-param>
```

9hat@ it& you don't need to "anua((y add any c to your &a,e or anythin, e(e& !ri "e#ace ' i((hand(e everythin, +or you.

In ca e you'd (i3e to "a3e the the "e dyna "ic& define an E6 e0&re ion a the &ara " va(ue.

```
<context-param>
    <param-name>primefaces.THEME</param-name>
    <param-value>#{loggedInUser.preferences.theme}</param-value>
</context-param>
```

8.2 Creating a New Theme

If you'd like to create your own theme instead of using the predefined one that is already available, because it provides a lot of useful features and easy to understand visually too.



Additionally, your own custom theme is also a rebuild of the existing one whenever you need to integrate the downloaded theme into your own structure. It follows the convention integrated by your custom theme to your project structure. This approach requires you to create a /resources and add it to the classpath of your application. All files must have the .css extension, and you can have one or more themes in a single jar.

-jar

```
- META-INF
  - resources
    - primefaces-yourtheme
      theme.css
      images
```

With the new theme you've downloaded, make sure you have a classpath and include it in your build path. Make sure you have "select content" option on download & include so that your theme only includes the necessary files. Extract the content of the theme and rename "jquery-ui-{version}.custom.css" to "theme.css".

Any reference in your theme.css file that needs to be converted to an external resource (such as images) can be placed under the "resources" folder.

For example, if you have a file named "image.png" in the "resources/images" folder, you would reference it as:

however, the file is located in the "resources/images" folder.

Once the file is in the "resources" folder, you can use your theme.css file to reference it:

```
<context-param>
  <param-name>primefaces.THEME</param-name>
  <param-value>yourtheme</param-value>
</context-param>
```

8.3 How Themes Work

!o ' ered by 9he " eRo((er!ri " e#ace e&arate structura(c +ro" 3innin, c .

Structura(*SS

9he e ty(e c(a e de+ine the 3e(eton o+ the co " &onent and inc(ude c &ro&ertie uch a " ar, in&addin, di &(ay ty&di " en ion and &o itionin,..

S3innin, *SS

S3innin, de+ine the (oo3 and +ee(&ro&ertie (i3e co(or border co(or bac3, round i " a, e .

Skinning Selectors

9he " eRo((er +eature a cou&(e o+ 3innin, e(ector E " o t i " &ortant o+ the e areK

Selector	Applies
.uiB ' id, et	A((!ri " e#ace co " &onent
.uiB ' id, etBheader	. eader ection o+ a co " &onent
.uiB ' id, etBcontent	*ontent ection o+ a co " &onent
.uiB tateBde+au(t	De+au(t c(a o+ a c(ic3ab(e
.uiB tateBhover	. over c(a o+ a c(ic3ab(e
.uiB tateBactive	When a c(ic3ab(e i e(ected
.uiB tateBdi ab(ed	Di ab(ed e(e " ent .
.uiB tateBhi ,h(i, ht	. i, h(i, hted e(e " ent .
.uiBicon	An e(e " ent to re&re ent an icon.

9he e c(a e are not a ' are o+ structura(c (i3e " ar, in i3e " 2 b r b #&enbASnDA e9hAU`nnin, icon.

8.4 Theming Tips

When creating your own theme, it's better than to extend the existing ones. Instead, you can override the styles of the components by adding your own CSS rules.

```
.ui-widget, .ui-widget .ui-widget {
    font-size: 90% !important;
}
```

- When creating your own theme, it's better than to extend the existing ones. Instead, you can override the styles of the components by adding your own CSS rules.
- If you are using RichFaces 4 or later, you can use the `!important` rule to override the existing styles.
- Another way to override the styles is to add a custom CSS class to the panel header and then apply your own styles to that class.

```
.ui-panel-titlebar {
    //css
}
```

or

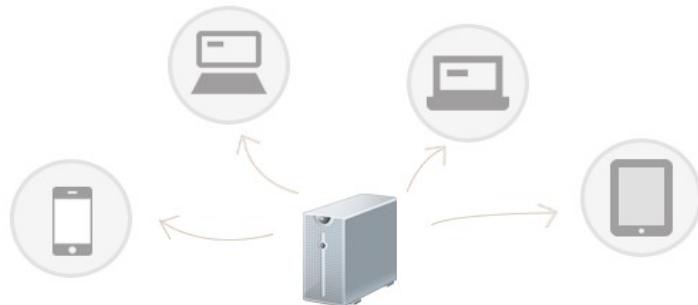
```
.ui-panel-titlebar.ui-widget-header {
    //css
}
```

To add a specific style to a particular component, you can use the `styleClass` attribute.

```
<p:panel styleClass="custom">
    ...
</p:panel>
```

```
.custom .ui-panel-titlebar {
    //css
}
```

9. PrimeFaces Push



!ri " e#ace !u h ?#!@ i a &u h +ra "e ' or3 bu(t on to& o+ At "o &here. At "o &here creator A ync17 i a &artner co "&any o+ !ri "e9e3 and the deve(o&er o+ !#!. At "o &here i hi,h(y ca(ab(eE u&&ort evera(container and bro ' er E uti(i=e variou tran &ort uch a ' eb oc3et E eeE(on, B&o((in,E trea " in, and / on&. #or " ore in+or " ation &(ea e vi itK

<https://github.com/Atmosphere/atmosphere>

9.1 Setup

Atmosphere

At "o &here i reCuired to run !ri " e#ace !u hE in your &o ".0 " (de&endency a K

```

<dependency>
    <groupId>org.atmosphere</groupId>
    <artifactId>atmosphere-runtime</artifactId>
    <version>2.1.3</version>
</dependency>
  
```

Push Servlet

!u h Serv(et i u ed a a ,ate ' ay +or c(ient .

```

<servlet>
    <servlet-name>Push Servlet</servlet-name>
    <servlet-class>org.primefaces.push.PushServlet</servlet-class>
    <async-supported>true</async-supported>
</servlet>

<servlet-mapping>
    <servlet-name>Push Servlet</servlet-name>
    <url-pattern>/primepush/*</url-pattern>
</servlet-mapping>
  
```

9.2 Annotations

The `@PushEndPoint` annotation driven centrally around the "push endpoint".

`@PushEndPoint`

This annotation is used to create a push endpoint by using the `@PushEndPoint` annotation. This annotation provides the mechanism to build an application using the Push API without the need to interact with the browser directly. At the same time, it also provides the ability to reduce the number of code required to build a real-time application by translating parent components into children. At the same time, it also provides the ability to handle heartbeat detection, connection establishment, recovery, etc. It also provides one attribute called `path` to define the resource path.

path: This is the path to the resource.

This default implementation of the `PushEndpoint` interface is delivered to your annotated class. You can also override the `path` attribute value to be used to "push" an incoming message to an annotated `push endpoint` class.

```
@PushEndPoint("/chat")
```

`@Singleton`

This annotation is used to force creation, a single thread access in case of a push endpoint annotated class. #or else "#(else) is your application set the `push endpoint` attribute with a value by default to the annotated class to be created. When annotated with `Singleton`, however, it is created.

`@OnOpen`

This annotation is used when the connection is ready to be used, or to write data. Annotated method needs to take the form:

```
@OnOpen
public void onOpen();

@OnOpen
public void onOpen(RemoteEndpoint r);

@OnOpen
public void onOpen(RemoteEndpoint r, EventBus e);
```

This `RemoteEndpoint` represents the connection and can be used to write data back to the browser. The `EventBus` can be used to fire events to one or more `RemoteEndpoint`s in, respectively, one or more sessions.

`@OnMessage`

This annotation is used when a message is ready to be delivered, and it is delivered to the browser. The `EventBus` attribute can be used to fire events to one or more `RemoteEndpoint`s in, respectively, one or more sessions.

encoders A interface that is used to encode the annotated "method return value. The returned value of an annotated `PushMethod` method will be broadcasted to all registered associated with the broadcasted with this annotated class.

decoders A interface used to decode a broadcasted "method into an object "atching, the "method signature.

Both of the encoder and decoder can be chained. When the returned value of an encoder or decoder can be used as an input for the next one.

```
@OnMessage(encoders = {JSONEncoder.class})
public String onMessage(String count) {
    return count;
}
```

But often it is necessary to provide an annotation that are required for the encoder and decoder. For example, the `Encoder` interface requires a `String` parameter, while the `Decoder` interface requires a `String` parameter.

It is also possible to create your own encoder or decoder by implementing `org.primefaces.push.Encoder` and `org.primefaces.push.Decoder` interfaces.

@OnClose

The `@OnClose` annotation is used to invoke a method when the client disconnects, closing the connection or when a network failure occurs or when a proxy closes the connection. Annotated "method need to take the form:

```
@OnClose
public void onClose();

@OnClose
public void onClose(RemoteEndpoint r);

@OnClose
public void onClose(RemoteEndpoint r, EventBus e);
```

Only one "method in a class can be annotated with `@OnClose`.

@PathParam

The `@PathParam` annotation is used to automatically map the path and query parameters to class variables.

```
@PushEndpoint("/somepath/{room}/{user}")
@Singleton
public class ChatResource {

    @PathParam("room")
    private String room;

    @PathParam("user")
    private String username;

    //...
}
```

9.3 API

RemoteEndPoint

The RemoteEndPoint interface represents the remote connection to the server. An instance of RemoteEndPoint holds information about the header, QueryString, body, uri and path and contains methods that can be used to set annotations, the incoming, received message. If you are familiar with the Servlet API, then the RemoteEndPoint can be seen as an improved version. You can also use a RemoteEndPoint to write back to the client that it has been delivered uniquely to the browser.

EventBus

A distributed (intermediate) event bus which can encode and decode events in binary. The event bus is intended to be used over the network and to point to a single, shared event over the event bus can be of any type. #or &#ub(i h l) ubcribe and &oint to &oint "e a,in,. 8e a,e ent over the event bus can be of any type. #or &ub(i h l) ubcribe "e a,e can be &ub(i hed to a Java class annotated with the `@Event` annotation, one of the methods annotated with `@OnAnyEvent`. The EventBus is the receiver of ended messages or derived from, "e a,e . EventBus is retrieved via

```
EventBus eventBus = EventBusFactory.getDefault().eventBus();
```

Once an EventBus instance is available, data to be published is done through overloaded publish methods.

publish(Object o): #ire ,iven object to all connected RemoteEndpoints .

publish(String path, Object o): #ire ,iven object to RemoteEndpoint that connected to the path.

publish(String path, Object o, Reply reply): #ire ,iven object to RemoteEndpoint that connected to the path by a Re#(y in instance. Re#(y is an interface with a callback method completed(String path) that is invoked when the EventBus delivered the "e a,e to the RemoteEndpoint that matches the path used to initiate the publication.

9.4 Socket Component

Socket is a interface component that handles the connection between the server and the browser. It is used to publish data to the socket by defining a path and a callback to handle the broadcast.

```
<p:push channel="/chat" onmessage="handlePublish"/>
```

See Socket component documentation for the full available options .

Client Side API

Widet PrimeFaces.widget.Socket

Method	Params	Return Type	Description
connect(uri)	uri	void	Connect to given uri.
disconnect	/on	void	Close the data flow connection.
connect		void	Disconnect from the channel.

incre" ent button action(i tener +ir t add one to the counter value, et an Event1u and then &u he the ne' va(ue to a((connected c(ient . #ina(&iece i ' ritin, the *ounterRe ource to hand(e "e a,e ' henever a data i &u hed.

```
package org.primefaces.examples.push.counter;

import org.primefaces.push.annotation.OnMessage;
import org.primefaces.push.annotation.PushEndpoint;
import org.primefaces.push.impl.JSONEncoder;

@PushEndpoint("/counter")
public class CounterResource {

    @OnMessage(encoders = {JSONEncoder.class})
    public String onMessage(String count) {
        return count;
    }
}
```

In ca e you'd (i3e to u&date co " &onent and/or inv03e (i tener in your bac3in, bean on broadca t& you can u e the o&tiona(*message* a/a0 behavior to i " &(e " ent the a"e +unctiona(ity but ' ith an e0tra reCue t.

```
<h:form id="form">
    <h:outputText id="out" value="#{globalCounter.count}" />
```

9.5.2 FacesMessage

9hi a"&(e ho' ho' to &u h #ace 8e a,e +ro" one c(ient to a((other and di &(ay the " u in, Gro '(*o" &onent.

```
<p:growl widgetVar="growl" showDetail="true" />

<h:form>
    <p:panel header="Growl">
        <h:panelGrid columns="2">
            <p:outputLabel for="summary" value="Summary: " />
            <p:inputText id="summary" value="#{growlBean.summary}" required="true" />

            <p:outputLabel for="detail" value="Detail: " />
            <p:inputText id="detail" value="#{growlBean.detail}" required="true" />
        </h:panelGrid>

        <p:commandButton value="Send" actionListener="#{growlBean.send}" />
    </p:panel>
</h:form>

<p:socket onMessage="handleMessage" channel="/notify" />

<script type="text/javascript">
function handleMessage(facesmessage) {
    facesmessage.severity = 'info';
    PF('growl').show([facesmessage]);
}
</script>
```

```
package org.primefaces.examples.view;

import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.RequestScoped;
import javax.faces.context.FacesContext;
import javax.faces.event.ActionEvent;
import org.primefaces.push.EventBus;
import org.primefaces.push.EventBusFactory;

@ManagedBean
@RequestScoped
public class GrowlBean {

    private final static String CHANNEL = "/notify";
    private String text, summary, detail;
    //getters-setters

    public void send() {
        EventBus eventBus = EventBusFactory.getDefault().eventBus();
        eventBus.publish(CHANNEL, new FacesMessage(summary, detail));
    }
}
```

<otiyRe ource i "y encode the &u hed "e a,e a AS7<.

```
package org.primefaces.examples.push.notify;

import javax.faces.application.FacesMessage;
import org.primefaces.push.annotation.OnMessage;
import org.primefaces.push.annotation.PushEndpoint;
import org.primefaces.push.impl.JSONEncoder;

@PushEndpoint("/notify")
public class NotifyResource {

    @OnMessage(encoders = {JSONEncoder.class})
    public FacesMessage onMessage(FacesMessage message) {
        return message;
    }

}
```

9.6 Tips and Tricks

Dynamic Paths

*(ient ide A !I ' ou(d be handy to create dyna " ic channe(E channe(na " e o+ the oc3et doe not need to be static and you can create dyna " ic channe(on runti " e ince a channe(i ba ica((y a &ath na " e. b !ath!ara " i very handy to auto " atica((y &ar e the !ath and a i,n the &ath to3en to variab(e .

See chat a " &(e in ho ' ca e +or an e0a " &(e o+ dyna " ic channe(u ed to end &rivate " e a,e .

Proxies

!ro0ie are &rob(e " atic not /u t +or !ri " e#ace !u h B At "o &here o(ution but in a((o(ution . I+ your &ro0y u&&ort 'eb oc3et E "a3e ure to add the nece ary con+i,uration. Another o(ution that i con idered a a ' or3around i to override the de+au(t uri o+ the &u h erver. De+au(t uri i &rotoco(NQQconte0t!ath&ri " e&u h0channe(E +or e0a "&(e !ri " e#ace on(ine ho ' ca e i runnin, on /etty that i behind an a&ache "od &ro0y ' hich doe n't u&&ort 'eb oc3et at ti "e o+ the 'ritin,. So(ution i to con+i,ure !ri " e#ace to u e another &u h erver (i3ek

```
<context-param>
    <param-name>primefaces.PUSH_SERVER_URL</param-name>
    <param-value>http://www.primefaces.org:8080</param-value>
</context-param>
```

So that oc3et co " &onent by&a e the &ro0y and direct(y co " " unicate ' ith the a&&(ication.

Supported Server and Client Environments

At "o &here doe a a,reat /ob in u&&ortin, di+erent erver and bro ' er . See the detai(ed (i t atk

<https://github.com/Atmosphere/atmosphere/wiki/Supported-WebServers-and-Browsers>

Scalability

At "o &here i bui(d to ca(e via &(u,in uch a A 8 SE RediOE F 8 ! !E . a=e(ca t and " ore. Re+er to at "o &here docu " entation to ee ho ' to con+i,ure at "o &here in " ore than one erver. !u hServ(et e0tend +ro " At "o &hereServ(et o any con+i,uration o&tion +or At "o &hereServ(et a(o a&&ie !u hServ(et.

10. PrimeFaces Mobile

!ri "e#ace 8 obi(e ?#!# 8 @ i a UI ; it to create AS# a&&(ication o&ti " i=ed +or " obi(e device . !# 8 i bui(t on to& o+ /Duery 8 obi(e a touchBo&ti " i=ed . 9 8 65 UI +ra" e ' or3 &rovidin, u&&ort +or variou &(at+or" .

In addition to the ea "(e inte,ration ' ith /Duery 8 obi(e !# 8 +eature a 8 obi(e Render3it +or &o&u(ar !ri "e#ace co " &onent E a/a0 +ra" e ' or3 e0ten ion E " obi(e a/a0 behavior event E inte,rated navi,ation " ode(E(a=y (oadin, o+ &a, e E re &on ive ' id,et and " ore.

10.1 Setup

8 obi(e hi& ' ith !ri "e#ace *ore o no additiona(do ' n(oad i reCuired.

Configuration

A " obi(e navi,ation hand(er i nece ary in ide +ace con+i,uration to enab(e navi ,ation u&&ort.

```
<application>
    <navigation-handler>
        org.primefaces.mobile.application.MobileNavigationHandler
    </navigation-handler>
</application>
```

Taglib

!# 8 &rovide " obi(e &eci+ic co " &onent ' ith the +o((o ' in , ta,(ib.

```
xmlns:pm="http://primefaces.org/mobile"
```

RenderKit

Render ; it i the core &art o+ !# 8 +eaturin, o&ti " i=ed renderer +or " obi(e environ " ent . 9here are t ' o " ain ' ay to enab(e !# 8 Render ; it.

*ore vie ' ta, ha a render ; itId attribute to et the renderer 3it o+ the &a,e.

```
<f:view renderKitId="PRIMEFACES_MOBILE" />
```

+ your a&&(ication +u((y con i t o+ " obi(e &a,e E then a ,(oba(con+i,uration in +ace con+i,uration +i(e hou(d be &re+erred in tead o+ &a,e con+i,uration.

```
<application>
    <default-render-kit-id>PRIMEFACES_MOBILE</application>
</application>
```

10.2 Pages

A " obi(e &a,e i a re,u(ar +ace(et Oht " (+i(e ' ith " obi(e &a,e container de+ined ' ith & " N&a,e.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:pm="http://primefaces.org/mobile">

    <f:view renderKitId="PRIMEFACES_MOBILE" />

    <h:head>
    </h:head>

    <h:body>

        <pm:page>
            <pm:header title="Basic Page"></pm:header>
            <pm:content></pm:content>
        </pm:page>

    </h:body>
</html>
```

A in,(e &a,e Oht " (vie ' can have " ore than one &a,e. 1y de+au(t the +ir t &a,e i vi ib(e on (oad.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:pm="http://primefaces.org/mobile">

    <f:view renderKitId="PRIMEFACES_MOBILE" />

    <h:head>
    </h:head>

    <h:body>

        <pm:page id="main">
            <pm:header title="Main Page"></pm:header>
            <pm:content></pm:content>
        </pm:page>

        <pm:page id="second">
            <pm:header title="Second Page"></pm:header>
            <pm:content></pm:content>
        </pm:page>

    </h:body>
</html>
```

10.2 Navigations

8 adding a standard navigation with the addition of a custom navigation mode.

Internal

Internal navigation is best seen & explained in the [Introduction](#). You'd have the **pm:reverse** or this third navigation. Within, the [Navigation section 10.1](#).

```
<p:button outcome="pm:second" />
<p:link outcome="pm:second" />
```

Button and link components do exactly the same thing in *second*. In case you'd like to navigate after a `!JSF` cue to another page.

```
<p:commandButton value="Go to Second" action="#{bean.go}" />
```

```
public String go() {
    return "pm:second";
}
```

External

External navigation to another URL (in a different domain or a resource in another domain) is also a standard approach.

```
<p:button outcome="/ui/home" value="Home" />
<p:button href="http://www.primefaces.org" value="Home" />
```

Transitions

Animations are available to run during the navigation. `` effect is appended to the outcome. In addition, `reverse` option is provided for backward navigation.

```
<p:button outcome="pm:second?transition=pop" />
<p:link outcome="pm:second?transition=flip&reverse=true" />
```

It's also possible to transition in `fade`, `pop`, `flip`, `turn`, `flow`, `slide`, `slidefade`, `slideup` and `slidedown`. #add it the default transition and to turn off animation set none as the value.

Client API

A client-side API is available to navigate using the `Mobile.navigate` method. Since, it's

```
PrimeFaces.Mobile.navigate(to, cfg);
```

```
PrimeFaces.Mobile.navigate('#second', {
    reverse: true/false,
    transition: 'fade'
});
```

10.3 Components

10.3.1 Content

*content is a container component for the content area of a page.

Info

9a,	content
*o " &onent *(a	org.primefaces.mobile.component.content.Content
*o " &onent 9y&e	org.primefaces.mobile.Content
*o " &onent #a " i(y	org.primefaces.mobile.component
Renderer 9y&e	org.primefaces.mobile.component.ContentRenderer
Renderer *(a	org.primefaces.mobile.component.content.ContentRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component
rendered	true	Boolean	Boolean value to specify if the component is rendered. If false, it is not rendered.
binding	null	Object	An expression that binds to a server-side UI component in a backing bean.
type	null	String	Defines the type of the component.
type*(a	null	String	Style class for the component.

Getting Started with the Content

*content is a child component of a page component.

```
<pm:page>
    <pm:content>
        //page content
    </pm:content>
</pm:page>
```

10.3.2 Field

#ie(d i a re &on ive (ayout co " &onent +or (abe(Bin&ut &air .

Info

9a,	field
*o " &onent *(a	org.primefaces.mobile.component.field.Field
*o " &onent 9y&e	org.primefaces.mobile.Field
*o " &onent #a " i(y	org.primefaces.mobile.component
Renderer 9y&e	org.primefaces.mobile.component.FieldRenderer
Renderer *(a	org.primefaces.mobile.component.field.FieldRenderer

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	True (value to display) if the component is rendered, false (value to hide) if not rendered.
binding	null	Object	An expression that maps to a server-side UI component instance in a backing bean.

Getting Started with the Field

#ie(d i u ed a the container o+ a (abe(and an in&ut co " &onent. A a re &on ive co " &onentE +ie(d di &(ay the o&ti " a(&(ace " ent +or it chi(dren ba ed on avai(ab(e ' idth.

```
<pm:field>
    <p:outputLabel for="text" value="Text:" />
    <p:inputText id="txt" />
</pm:field>
```

10.3.3 Footer

#ooter i container co " &onent +or the botto " area o+ a &a, e.

Info

9a,	footer
*o " &onent *(a	org.primefaces.mobile.component.footer.Footer
*o " &onent 9y&e	org.primefaces.mobile.Footer
*o " &onent #a " i(y	org.primefaces.mobile.component
Renderer 9y&e	org.primefaces.mobile.component.FooterRenderer
Renderer *(a	org.primefaces.mobile.component.footer.FooterRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	UniCue identi+ier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin, o+ the co " &onent) ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
tit(e	nu((Strin ,	9it(e te0t o+ the +ooter.
+i0ed	+a(e	1oo(ean	!o ition the +ooter a +i0ed on cro((.
the " e	nu((Strin ,	S ' atch o+ the co " &onent.
ta&9o , ,(e	true	1oo(ean	#or +i0ed +ooter E et ' hether the +i0ed too(bar) vi ibi(ity can be to , ,(ed by ta&&in , on the &a, e.
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.

Getting Started with the Footer

#ooter i a chi(dren co " &onent o+ a &a, e.

```
<pm:page>
    <pm:footer title="Footer"></pm:footer>
</pm:page>
```

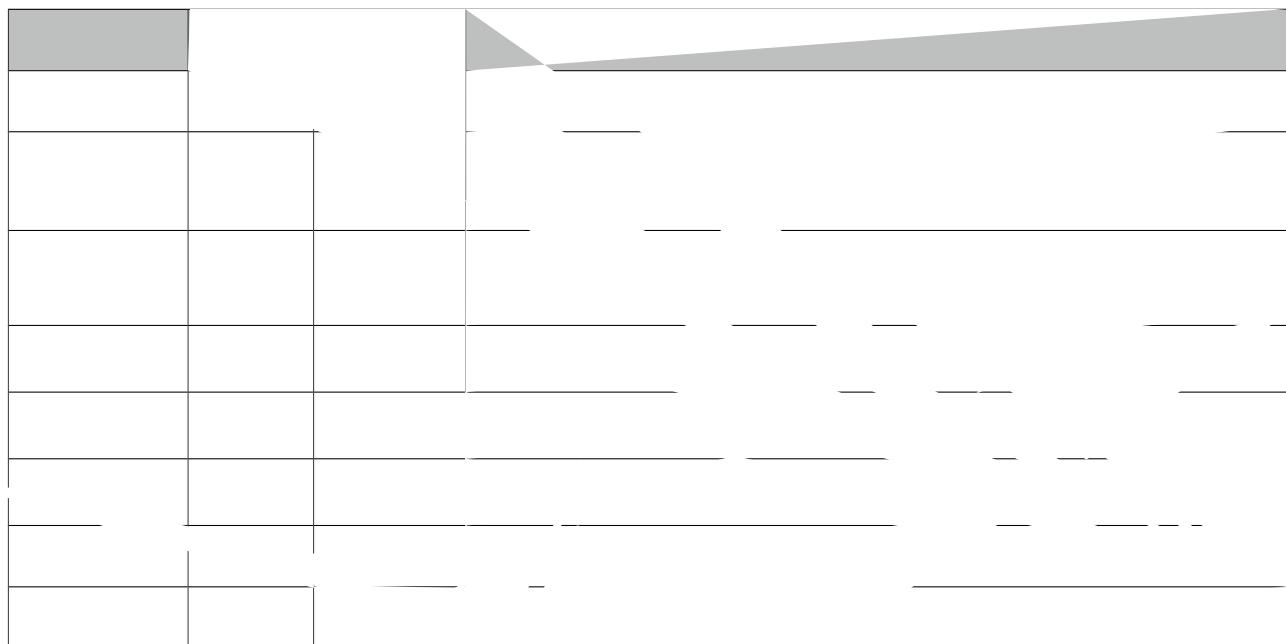
10.3.4 Header

`.eader i container co " &onent +or the to& area o+ a &a, e.`

Info

9a,	header
*o "&onent *(a	org.primefaces.mobile.component.header.Header
*o "&onent 9y&e	org.primefaces.mobile.Header
*o "&onent #a " i(y	org.primefaces.mobile.component.Header
Renderer 9y&e	org.primefaces.mobile.component.HeaderRenderer
Renderer *(a	org.primefaces.mobile.component.header.HeaderRenderer

Attributes



10.3.5 InputSlider

InputSlider is an input component with a touch enabled slider.



Info

9a,	inputSlider
*o " &onent *(a	org.primefaces.mobile.component.inputslider.InputSlider
*o " &onent 9y&e	org.primefaces.mobile.InputSlider
*o " &onent #a " i(y	org.primefaces.mobile.component.
Renderer 9y&e	org.primefaces.mobile.component.InputSliderRenderer
Renderer *(a	org.primefaces.mobile.component.inputslider.InputSliderRenderer

Attributes

Name	Default	Type	Description
id	nu((String	Unique identifier of the component.
rendered	true	Boolean	Logical value to specify the rendering of the component. If false, the component will not be rendered.
binding	nu((Object	An expression that refers to a server-side UI component instance in a backing bean.
value	nu((Object	The value of the component.
converter	nu((Object	An expression or an iterator that defines a converter for the component. When it is an EL expression, it refers to a converter instance. In case it is a static iterator, it refers to a converter id.
immediate	false	Boolean	When set to true, validation is executed at the request time and the component's value is updated.
required	false	Boolean	Specifies if the component is required.
validator	nu((Object	A method binding expression that refers to a method for validation, the input.
valueChangeListener	nu((Object	A method binding expression that refers to a method for handling value change events.
requiredMessage	nu((String	The message to be displayed when the component is required.

Name	Default	Type	Description
converter	null	String	The converter to be displayed when conversion fails.
validator	null	String	The validator to be displayed when validation fails.
minimum	0	Integer	The minimum value.
maximum	100	Integer	The maximum value.
type	null	String	The type of the component.
type*(a)	null	String	Style class of the component.
step	1	String	Step factor to affect the slider value.
disabled	false	Boolean	Disable or enable the slider.
(ab)enable	null	String	User &re-enable the slider.
hi,high	false	Boolean	Whether the value ranges when enabled.

Getting Started with the InputSlider

InputSlider requires an integer input value.

```
<pm:inputSlider value="50" />
<pm:inputSlider value="#{backingView.integerProperty}" />
```

Boundaries

0 and 100 are the default boundaries but can be customized using `minValue` and `maxValue`.

```
<pm:inputSlider minValue="100" maxValue="200" value="#{backingView.intProperty}" />
```

10.3.6 Page

This section describes the `Page` component.

Info

9a,	page
*o " &onent *(a	org.primefaces.mobile.component.page.Page
*o " &onent 9y&e	org.primefaces.mobile.Page
*o " &onent #a " i(y	org.primefaces.mobile.component
Renderer 9y&e	org.primefaces.mobile.component.PageRenderer
Renderer *(a	org.primefaces.mobile.component.page.PageRenderer

Attributes

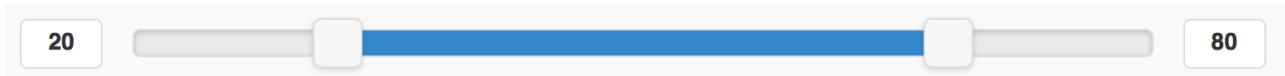
Name	Default	Type	Description
id	nu((Strin ,	Unique identifier o+ the co " &onent
rendered	true	1oo(ean	1oo(ean va(ue to &eci+y the renderin , o+ the co " &onent) ' hen et to +a(e co " &onent ' i((not be rendered.
bindin ,	nu((7b/ect	An e(e0&re ion that " a& to a erver ide UI*o " &onent in tance in a bac3in , bean
tit(e	nu((Strin ,	9it(e te0t o+ the &a ,e.
the " e	nu((Strin ,	S ' atch o+ the &a ,e.
ty(e	nu((Strin ,	In(ine ty(e o+ the co " &onent.
ty(e*(a	nu((Strin ,	Sty(e c(a o+ the co " &onent.
(a=y	+a(e	1oo(ean	6a=y (oadin , vie ' are not rendered on initia(&a ,e (oad to i " &rove &er+or " ance and in stead (a=i(y (oaded on de " and ' hen there are +ir t navi ,ated to.

Getting Started with the Page

This section describes the `Page` component.

10.3.7 RangeSlider

RangeSlider is a component for creating a range selection.



Info

9a,	rangeSlider
*o " &onent *(a	org.primefaces.mobile.component.rangeslider.RangeSlider
*o " &onent 9y&e	org.primefaces.mobile.RangeSlider
*o " &onent #a " i(y	org.primefaces.mobile.component
Renderer 9y&e	org.primefaces.mobile.component.RangeSlider
Renderer *(a	org.primefaces.mobile.component.rangeslider.RangeSlider

Attributes

Name	Default	Type	Description
id	null	String	Unique identifier for the component.
rendered	true	Boolean	True (value to display the rendering) or false (value to hide the component). If not rendered.
binding	null	Object	An expression that binds to a server-side UI component instance in a backing bean.
type	null	String	Indicates the type of the component.
type*(a	null	String	Style class for the component.
height	auto	Boolean	.height the value range when enabled.

Getting Started with the RangeSlider

RangeSlider needs to render a child element for the start of range and another for the end.

```
<pm:rangeSlider>
    <pm:inputSlider value="#{backingView.start}" />
    <pm:inputSlider value="#{backingView.end}" />
</pm:rangeSlider>
```

10.3.8 Switch

S'itch is an input component to select a boolean value.



Info

9a,	switch
*o " &onent *(a	org.primefaces.mobile.component.uiswitch.UISwitch
*o " &onent 9y&e	org.primefaces.mobile.UISwitch
*o " &onent #a " i(y	org.primefaces.mobile.component
Renderer 9y&e	org.primefaces.mobile.component.UISwitchRenderer
Renderer *(a	org.primefaces.mobile.component.uiswitch.UISwitchRenderer

Attributes

Name	Default	Type	Description
id	nu((Strin ,	Unique identifier of the component.
rendered	true	1oo(ean	Boolean value to specify if the component is rendered. If false, the component will not be rendered.
bindin ,	nu((7b/ect	An expression that adds to a server-side UI component in a backing bean.
va(ue	nu((7b/ect	>a(ue of the component.
converter	nu((7b/ect	An expression or an iterator that defines a converter for the component. When it is an EL expression, it refers to a converter in the context. In case it is a static iterator, it refers to a converter id.
i " " ediate	+a(e	1oo(ean	When set to true, the value is executed at a specific time and has no effect on the component.
reCured	+a(e	1oo(ean	Specifies the component required.
va(idator	nu((7b/ect	A method binding, expression that refers to a method validation, the input.
va(ue*han ,e6i tener	nu((7b/ect	A method binding, expression that refers to a method handler, a valuechange event.

Name	Default	Type	Description
reCquired	#{empty}	String	Specifies whether the field must be filled in or not. If set to true, the field is required.
converter	#{empty}	String	Specifies the converter to be used for the field.
validator	#{empty}	String	Specifies the validator to be used for the field.
on6abe	on	Listener	Initial value of the field.
o+6abe	0+	Listener	Final value of the field.
(abe)	#{empty}	String	User-defined tabbing name.
ty(e)	#{empty}	String	Inline type of the component.
ty(e*(a	#{empty}	String	Style class of the component.
(abe)	#{empty}	String	User-defined tabbing name.
di ab(ed	+able	Boolean	Disable or enable the field.
onchan,e	+able	String	Identifies the bac3 to execute on change event.

Getting Started with the Switch

>a(ue o+ ' itch hou(d be a boo(ean &ro&erty i+ the va(ue i +a(e o+6abe(' ou(d be di &(ayed and on6abe(' ou(d be u ed other ' i e.

```
<pm:switch value="#{backingView.booleanProperty}" />
```

10.4 RenderKit

In addition to the "obi(e co" &onent E variou core co" &onent have been enhanced 'ith "obi(e renderer o that a "e co" &onent can be o&ti " i=ed +or a "obi(e bro ' er. Since u a,e i a" e in de 3to& and "obi(e environ "ent E re+er to cha&ter) +or docu "entation about the e co" &onent .

10.4.1 AccordionPanel

- **Godfather Part I**

The story begins as Don Vito Corleone, the head of a New York Mafia family, oversees his daughter's wedding. His beloved son Michael has just come home from the war, but does not intend to become part of his father's business. Through Michael's life the nature of the family business becomes clear. The business of the family is just like the head of the family, kind and benevolent to those who give respect, but given to ruthless violence whenever anything stands against the good of the family.

+ **Godfather Part II**

+ **Godfather Part III**

10.4.2 AutoComplete

a ✖

	Afellay AMC	20
	Abidal LB	22
	Alves RB	2
	Adriano LB	21

10.4.3 Button

Default

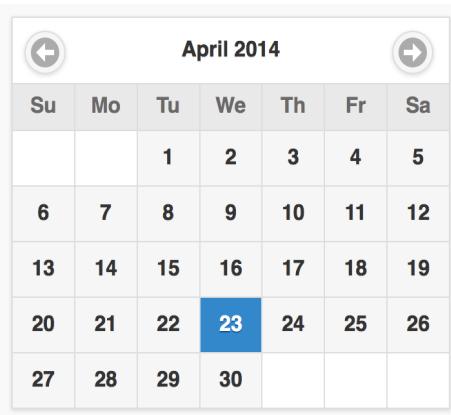
✓
With Icon
✖

Icon Position
C

Inline
X

i
Info

10.4.4 Calendar



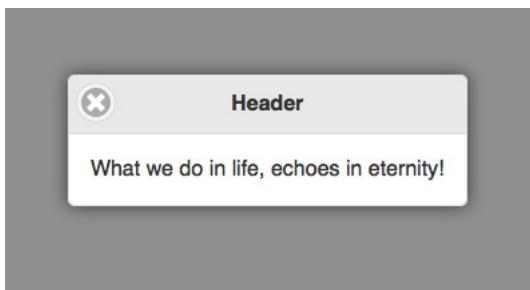
10.4.5 DataList

List of Players

	Messi	10	
	CF		
	Iniesta	8	
	CM		
	Villa	7	
	CF		
	Xavi	6	
	CM		
	Puyol	5	
	CB		

F.C. Barcelona

10.4.6 Dialog



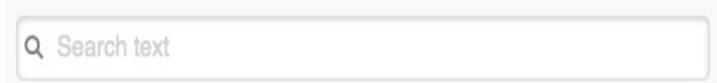
10.4.7 FileUpload



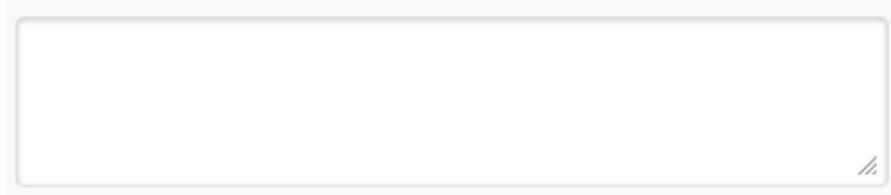
10.4.8 Growl



10.4.9 InputText

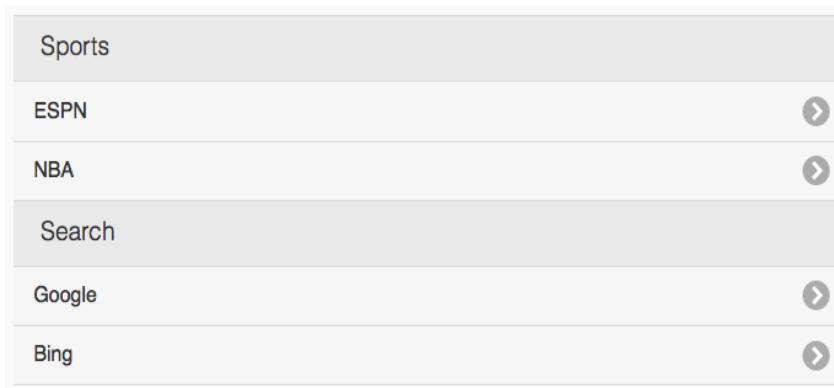


10.4.10 InputTextarea

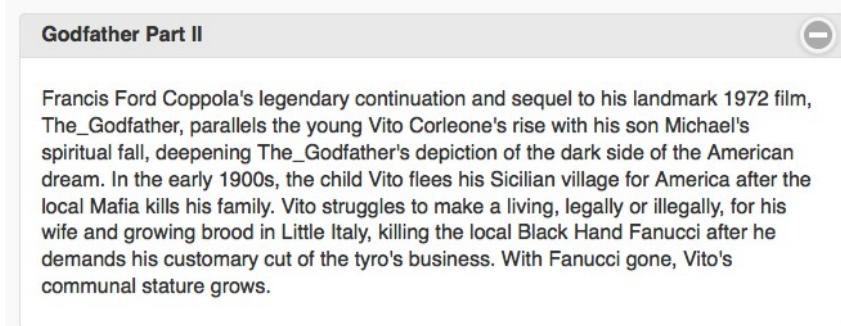


10.4.11 Link

10.4.12 Menu



10.4.13 Panel



10.4.14 PanelGrid

A	B	C
D	E	F
G	H	I

10.4.15 SelectBooleanCheckbox

I agree

10.4.16 SelectCheckboxMenu

Select Multiple

Option 1	<input checked="" type="checkbox"/>
Option 2	<input checked="" type="checkbox"/>
Option 3	<input type="checkbox"/>

10.4.17 SelectManyButton

Option 1 Option 2 Option 3

10.4.18 SelectManyCheckbox

<input type="checkbox"/> Option 1
<input checked="" type="checkbox"/> Option 2
<input checked="" type="checkbox"/> Option 3

10.4.19 SelectOneButton

Option 1 Option 2 Option 3

10.4.20 SelectOneMenu

Select One

10.4.21 SelectOneRadio

<input checked="" type="radio"/> Option 1
<input type="radio"/> Option 2
<input type="radio"/> Option 3

10.4.22 TabMenu

- [!\[\]\(fd1bd109363ad2aec56ff5114371524a_img.jpg\) Home](#)
- [!\[\]\(ebec4b8f9d91ce685942868ce1f073af_img.jpg\) Settings](#)
- [!\[\]\(70b014a501f572e3e538d8eeb6cc1d33_img.jpg\) Search](#)

10.4.24 TabView

Godfather Part I	Godfather Part II	Godfather Part III
------------------	-------------------	--------------------

The story begins as Don Vito Corleone, the head of a New York Mafia family, oversees his daughter's wedding. His beloved son Michael has just come home from the war, but does not intend to become part of his father's business. Through Michael's life the nature of the family business becomes clear. The business of the family is just like the head of the family, kind and benevolent to those who give respect, but given to ruthless violence whenever anything stands against the good of the family.

10.4.25 Toolbar

 New	 List	 Delete	 Settings
---	--	--	--

10.5 Themes

8 obiecte the " in, i i" i(ar to the core the " in, docu " ented at chapter 2. 8 ain difference i the ' atch conce&t o+ " obiecte the " e ' here one the " e can contain " u(ti&e ' atche . De+au(t the " e ha t'o ' atche **a** and **b**. A ' atch can be a&&(ied u in, ' atch attribute ' hen avai(ab(e.

```
<pm:header title="Header" swatch="b"></pm:header>
```

Custom Theme

9here i a " obiecte the " ero((er a&&(ication to create a cu to " the " e.

<http://themeroller.jquerymobile.com>

A+ter creatin, the the " e and do ' n(oadin, it a a =i& +i(eE a /ar +i(e ha to be created to in ta((the the " e ' ith the a " e +o(der tructure a a de 3to& the " e.

- jar

- META-INF
- resources
 - primefaces-mytheme
 - ↳ theme.css
 - ↳ images

1@ 9he the " e &ac3a,e do ' n(oaded +ro " 9he " eRo((er ' i((have a the " ena " e.c +i(e and i " a,e +o(der. E0tract the content o+ the &ac3a,e and ren a " e *themename.css* to *theme.css*.



%@ I " a,e re+erence in your the " e.c " u t a(o be converted to an e0&re ion that AS# re ource (oadin, can under tandE e0a " &(e ' ou(d beK

ur(?Ii " a,e QuiBb,Lhi,h(i,htBhardL100L+9+9+9L10100.&n,I@

hou(d beK

ur(?IXVre ource`&ri " e+ace B " ythe " eNi " a,e QuiBb,Lhi,h(i,htBhardL100L+9+9+9L10100.&n,laWI@

7nce the /ar o+ your the " e i in c(a &athE you can enab(e the the " e by the +o((o ' in, con+i, uration.

```
<context-param>
    <param-name>primefaces.mobile.THEME</param-name>
    <param-value>mytheme</param-value>
</context-param>
```

&ara " Bva(ue can be an E6 e0&re ion a ' e((to i " &(e " ent dyna " ic the " in,.

10.6 Framework

In addition to the UI component, PrimeFaces provides several features that interact with the framework.

10.6.1 Ajax Updates

PrimeFaces provides an enhancement mechanism that runs once on page load that allows you to enhance the user interface after it has been rendered. This means that the updated part cannot be enhanced again. This is because the update is under the hood.

A command button triggers the enhancement of the data and its associated functionality when updated with standard components like buttons here. This is done by updating the data and its associated functionality.

```
<h:commandButton value="Standard">
    <f:ajax render="panel1" />
</h:commandButton>

<p:commandButton value="PFM" update="panel2" icon="ui-icon-refresh"/>

<h:panelGroup id="panel1">
    <ul data-role="listview" data-inset="true">
        <li><a href="#">Item 1</a></li>
        <li><a href="#">Item 2</a></li>
        <li><a href="#">Item 3</a></li>
    </ul>

    <h:inputText />
</h:panelGroup>

<p:outputPanel id="panel2">
    <p:menu styleClass="ui-listview-inset ui-corner-all">
        <p:menuItem value="Item1" url="#" />
        <p:menuItem value="Item2" url="#" />
        <p:menuItem value="Item3" url="#" />
    </p:menu>

    <p:inputText />
</p:outputPanel>
```

10.6.2 Pass Through Elements

PrimeFaces provides a data attribute driven by the standard attribute. This attribute can have multiple values and can be used to add new attributes to a component.

```
<p:dataList value="#{ringBean.players}" var="player" pt:data-inset="true">
    //content
</p:dataList>
```

Note that this is the standard tag, `xmlns:pt="http://xmlns.jcp.org/jsf/passthrough"`. For the `pt` attribute, you can use the standard attribute.

10.6.3 Lazy Pages

#or a "obi(e device 'ith band 'ith concern E(oadin, a((the &a,e in a" e Oht" (at once in an ea,er+a hion i not o&ti "a(and " i,ht cau e &er+or "ance i ue i+the &a,e i=e i bi,. !# 8 u&&ort (a=y (oadin, &a,e ' here a &a,e content i not rendered at +ir t but (oaded ' ith a/a0 /u t be&ore acce ed +or the +ir t ti "e. 6a=y (oadin, i di ab(ed and activated by ettin, *lazy* o&tion on &a,e co " &onent.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui"
      xmlns:pm="http://primefaces.org/mobile">
    <f:view renderKitId="PRIMEFACES_MOBILE" />
    <h:head></h:head>

    <h:body>
        <pm:page id="main">
            <pm:header title="Main Page"></pm:header>
            <pm:content>
                <p:link outcome="pm:second" value="Go" />
            </pm:content>
        </pm:page>
        <pm:page id="second" lazy="true">
            <pm:header title="Second Page"></pm:header>
            <pm:content>
                //Lazy loaded content
            </pm:content>
        </pm:page>
    </h:body>
</html>
```

At +ir t &a,e (oadE econd &a,e content are not avai(ab(e on &a,eE c(ic3in, the (in3 +ir t " a3e an a/a0 reCue t to +etch the content E create the UI and then +ina((y navi,ate to it.

10.6.4 Touch Events

Touch event (i3e ' i&e are enab(ed in o " e co " &onent . In +o((o ' in,E ite " in the (i t can be re " oved by *swipeleft* or *swiperight* and detail(i di &(ayed ' ith *taphold*.

```
<h:form id="form">
    <p:growl id="msgs" showDetail="true" />

    <p:dataList value="#{tableBean.carsSmall}" var="car" pt: data-inset="true">
        <p:ajax event="swipeleft" listener="#{tableBean.swipeCar}"
               update=":form:msgs @this"/>
        <p:ajax event="swiperight" listener="#{tableBean.swipeCar}"
               update=":form:msgs @this"/>
        <p:ajax event="taphold" listener="#{tableBean.onRowSelect}"
               update=":form:msgs"/>
    <h:outputText value="#{car.manufacturer} - #{car.color}" />
</p:dataList>
</h:form>
```

```
public void onRowSelect(SelectEvent event) {  
    FacesMessage msg = new FacesMessage("Car Selected",  
        ((Car) event.getObject()).getModel());  
  
    FacesContext.getCurrentInstance().addMessage(null, msg);  
}  
  
public void swipeCar(SwipeEvent event) {  
    Car car = (Car) event.getData();  
    carsSmall.remove(car);  
    FacesContext.getCurrentInstance().addMessage(null,  
        new FacesMessage(FacesMessage.SEVERITY_INFO,  
            "Car Swiped", "Removed: " + car.getModel()));  
}
```

Section 10.- and 10.5 contain u&&orted "obi(e event +or a co " &onent ' hen avai(ab(e.

11. Utilities

11.1 RequestContext

ReCue t*onte0t i a i" &(e uti(ity that &rovide u e+u(,oodie uch a addin, &ara " eter to a/a0 ca((bac3 +unction . ReCue t*onte0t i avai(ab(e in both a/a0 and nonBa/a0 reCue t .

ReCue t*onte0t can be obtained i" i(ar(y to the #ace *onte0t.

```
RequestContext requestContext = RequestContext.getCurrentInstance();
```

RequestContext API

Method	Description
i A/a0ReCue t?@	Return a boo(ean va(ue i+ current reCue t i a !ri " e#ace a/a0 reCue t.
add*a((1ac3 !ara " ?Strin, na " eE 7b/ect va(ue@	Add &ara " eter to a/a0 ca((bac3 (i3e onco " &(ete.
u&date?Strin, c(ientId@	S&eci+ie co " &onent? @ to u&date at runti " e.
e0ecute?Strin, cri&t@	E0ecute cri&t a+ter a/a0 reCue t co " &(ete or on &a,e (oad.
cro((9o?Strin, c(ientId@	Scro((to the co " &onent ' ith ,iven c(ientId a+ter a/a0 reCue t co " &(ete .

Callback Parameters

9here " ay be ca e ' here you need va(ue +ro " bac3in, bean in a/a0 ca((bac3 . *a((bac3 &ara " eter are eria(i=ed to AS7< and &rovided a an ar,u" ent in a/a0 ca((bac3 +or thi .

```
<p:commandButton actionListener="#{bean.validate}"
    onComplete="handleComplete(xhr, status, args)" />
```

```
public void validate() {
    //isValid = calculate isValid
    RequestContext requestContext = RequestContext.getCurrentInstance();
    requestContext.addCallbackParam("isValid", true or false);
}
```

isValid &ara " eter ' i((be avai(ab(e in hand(e*o " &(ete ca((bac3 a K

```
<script type="text/javascript">
    function handleComplete(xhr, status, args) {
        var isValid = args.isValid;
    }
</script>
```

You can add a " any ca((bac3 &ara " eter a you ' ant ' ith add*a((bac3 !ara " A !I. Each &ara " eter i eria(i=ed a AS7< and acce ib(e throu,h ar, &ara " eter o &o/o are a(o u&&orted /u t (i3e

If you want to validate a value, you can add a validation method to the User class that has a property (like `isUserValid`) and return true or false. In addition to the boolean value, the client can also pass a User object.

```
public void validate() {
    //isValid = calculate isValid
    RequestContext requestContext = RequestContext.getCurrentInstance();
    requestContext.addCallbackParam("isValid", true or false);
    requestContext.addCallbackParam("user", user);
}
```

```
<script type="text/javascript">
    function handleComplete(xhr, status, args) {
        var firstname = args.user.firstname;
        var lastname = args.user.lastname;
    }
</script>
```

If `validationFailed` is true, the user will be added to the validation failed list.

Runtime Updates

Conditionally update UI components based on different parts of the page need to be updated based on a dynamic condition. In this case it is not efficient to use declarative update and defined a(update area since this will cause unnecessary update. Here you need to define which component to update rather than specifying, it declarative(`y. update`) method is added to handle this case. In other words, when deciding which part of the page to update based on the button action, consider whether the update needs to be done.

```
<p:commandButton value="Save" actionListener="#{bean.save}" />
<p:panel id="panel"> ... </p:panel>
<p:dataTable id="table"> ... </p:panel>
```

```
public void save() {
    //boolean outcome = ...
    RequestContext requestContext = RequestContext.getCurrentInstance();

    if(outcome)
        requestContext.update("panel");
    else
        requestContext.update("table");
}
```

When the save button is clicked and rendered, on the outcome event you can either consider the database or the panel to be updated with a new one.

Execute Javascript

Reusing existing code to execute JavaScript when the user interacts with the page is often required. This approach is easier to implement, as it can be used to validate the user input, and execute conditional JavaScript code. For example, if the user enters an invalid value, the dialog can be hidden when the user interacts with the page.

```
public void save() {  
    RequestContext requestContext = RequestContext.getCurrentInstance();  
  
    requestContext.execute("dialog.hide()");  
}
```

11.2 EL Functions

!ri " e#ace &rovide bui(tBin E6 e0ten ion that are he(&er to co " " on u e ca e .

Common Functions

Function	Description
co " &onent?0id@	Return c(ientId o+ the co " &onent ' ith &rovided erver id &ara " eter. 9hi +unction i u e+u(i+ you need to ' or3 ' ith /ava cri&t.
' id,et>ar?0id@	!rovide the ' id,et>ar o+ a co " &onent in !#?000 +or " at.

*o " &onent

```
<h:form id="form1">
    <h:inputText id="name" />
</h:form>
//#{p:component('name')} returns 'form1:name'
```

Wid,et>ar

```
<p:dialog id="dlg">
    //contents
</p:dialog>
<p:commandButton type="button" value="Show" onclick="#{p:widgetVar('dlg')}.show()" />
```

Page Authorization

Authori=ation +unction u e . tt&Serv(etReCue t A ! l +or the bac3end in+or " ation.

Function	Description
i+Granted?Strin, ro(e@	Return true i+ u er ha the ,iven ro(eE e(e +a(e.
i+A((Granted?Strin, ro(e @	Return true i+ u er ha a((o+ the ,iven ro(eE e(e +a(e.
i+AnyGranted?Strin, ro(e @	Return true i+ u er ha any o+ the ,iven ro(eE e(e +a(e.
i+<otGranted?Strin, ro(e @	Return true i+ u er ha none o+ the ,iven ro(eE e(e +a(e.
re " oteU er?@	Return the na " e o+ the (o,,ed in u er.
u er!rinci&a?@	Return the &rinci&a(in tance o+ the (o,,ed in u er.

```
<p:commandButton rendered="#{p:ifGranted('ROLE_ADMIN')}" />
<h:inputText disabled="#{p:ifGranted('ROLE_GUEST')}" />
<p:inputMask rendered="#{p:ifAllGranted('ROLE_EDITOR, ROLE_READER')}" />
```

11.3 Exception Handler

!ri " e#ace &rovide a bui(tBin e0ce&tion hand(er to ta3e care o+ e0ce&tion in a/a0 and nonBa/a0 reCue t ea i(y.

Configuration

E0ce&tion . and(er and an E(Re o(ver con+i ,ured i reCuired in +ace con+i ,uration +i(e.

```
<application>
    <el-resolver>
        org.primefaces.application.exceptionhandler.PrimeExceptionHandlerELResolver
    </el-resolver>
</application>

<factory>
    <exception-handler-factory>
        org.primefaces.application.exceptionhandler.PrimeExceptionHandlerFactory
    </exception-handler-factory>
</factory>
```

Error Pages

E0ce&tion . and(er i inte ,rated ' ith error&a,e " echani " o+ Serv(et A !I. At a&&(ication tartu&E !ri " e#ace &ar e the error &a,e and u e thi in+or " ation to +ind the a&&ro&riate &a,e to redirect to ba ed on the e0ce&tion ty&e. . ere i an e0a " &(e ' eb.0 "(con+i ,uration ' ith a ,eneric &a,e +or e0ce&tion and a &ecia(&a,e +or >ie ' E0&iredE0ce&tion ty&e.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" >

    <!-- Other application configuration -->

    <error-page>
        <exception-type>java.lang.Throwable</exception-type>
        <location>/ui/error/error.jsf</location>
    </error-page>
    <error-page>
        <exception-type>javax.faces.application.ViewExpiredException</exception-type>
        <location>/ui/error/viewExpired.jsf</location>
    </error-page>

</web-app>
```

Exception Information

In the error &a, eE in or "ation about the eOce&tion i &rovided via the &+E0ce&tion . and(er E6 3ey ' ord. . ere i the (i t o+ e0&o ed &ro&ertie .

B eOce&tionN 9hro ' ab(e in tance.
 B ty&eN 9y&e o+ the eOce&tion.
 B " e a,eN E0ce&tion " e a,eN
 B tac39raceN An array o+ /ava.an , Stac39raceE(e " ent in tance .
 B +or " attedStac39raceN Stac3 trace a &re entab(e trin,,
 B ti " e ta " &N 9i " e ta " & a date.
 B +or " atted9i " e ta " &N 9i " e ta " & a &re entab(e trin,,

In error &a, eE eOce&tion " etadata i acce ed u in, E6

```
<h:outputText value="#{pfExceptionHandler.message}" />
<h:outputText value="#{pfExceptionHandler.formattedStackTrace}" escape="false" />
```

Ajax Exception Handler Component

A &ecia(i=ed eOce&tion hand(er co " &onent &rovide a 'ay to eOecute ca((bac3 on c(ient ideE u&date other co " &onent on the a" e &a,e. 9hi i Cuite u e+u(in ca e you don't 'ant to create a e&arate error &a,e. #o((o ' in, e0a " &(e ho ' the eOce&tion in a dia(o, on the a" e &a,e.

```
<p:ajaxExceptionHandler type="javax.faces.application.ViewExpiredException"
    update="exceptionDialog" onexception="PF('exceptionDialog').show();" />

<p:dialog id="exceptionDialog" header="Exception: #{pfExceptionHandler.type}
occured!" widgetVar="exceptionDialog" height="500px">
    Message: #{pfExceptionHandler.message} <br/>
    StackTrace: <h:outputText value="#{pfExceptionHandler.formattedStackTrace}"
        escape="false" />
    <p:button onclick="document.location.href = document.location.href;" value="Reload!"/>
</p:dialog>
```

Idea(ocation +or &\a/a0E0ce&tion . and(er co " &onent i the +ace(et te " &(ate o that it ,et inc(uded in every &a,e. Re+er to co " &onent docu " entation o+ &\a/a0E0ce&tion . and(er +or the avai(ab(e attribute .

Render Response Exceptions

9o u&&ort eOce&tion hand(in, in the RENDER_RESPONSE &ha eE itl reCuired to et the javax.faces.FACELETS_BUFFER_SIZE &ara " eter. 7ther ' i e you ' i((&robab(y ee a Serv(etE0ce&tion ' ith IRe &on e a(ready co " " ittedI " e a,e.

11.4 BeanValidation Transformation

Since JavaEE 4.0 validation "metadata is already available or "any component" via the value reference and bean>validation? e.g., b <ot<u((E b Si=e@. This AS# I " &(e " entation u e this in-or " ation or server side validation and !ri " e#ace enhance this feature ' with client side validation +ra " e ' or3.

!ri " e#ace " a3e u e o+ the e " metadata by transformation, the " to component and ht " (attribute . #or else " &(e o " eti " e itZ required to " anual(y " aintain the required or " a0(en, th attribute +or in+ut component " &onent . This required attribute also controls the behavior of &Nout&ut6abe(to ' or hide the required indicator ?J@ ' herea the *maxlength* attribute is used to (i " it the character on in+ut+ie(d . 1ean>a(idation transformation +feature enables avoiding, " anual(y " aintainin, the e attribute any " ore by i " &(icity hand(in, the " behind the scene .

Configuration

To start with transformation enabled.

```
<context-param>
    <param-name>primefaces.TRANSFORM_METADATA</param-name>
    <param-value>true</param-value>
</context-param>
```

Usage

Define constraint at bean level.

```
@NotNull
@Max(30)
private String firstname;
```

*o " &onent at view ' does not have any constraint

```
<p:inputText value="#{bean.firstname}" />
```

#ina(output has ht " (" a0(en, th attribute , generated +ro " the b 8 a0 annotation also the component " &onent in tance has required enabled.

```
<input type="text" maxlength="30" ... />
```

11.5 PrimeFaces Locales

*o " &onent " ay reCuire tran (ation and other ettin, ba ed on di+erent (oca(e . 9hi i hand(ed ' ith a c(ient ide a&i ca((ed !ri " e#ace 6oca(e . A c(ient ide (oca(e i ba ica((y a /ava cri&t ob/ect ' ith variou ettin, EnLUS i the de+au(t (oca(e &rovided out o+ the bo0. In ca e you need to u&&ort another (oca(e ettin, hou(d be e0tended ' ith the ne ' in+or " ation.

A ' i3i &a,e i avai(ab(e +or u er contributed ettin, E the (i t i co " " unity driven and a ,ood tartin, &oint a(thou,h it " i,ht be inco " &(ete.

<https://code.google.com/p/primefaces/wiki/PrimeFacesLocales>

Default Locale

. ere i the (i t o+ a((3eyBva(ue &air +or enLUS (oca(e that i &rovided by !ri " e#ace . Date9i " e re(ated &ro&ertie are uti(i=ed by co " &onent uch a ca(endar and chedu(e. I+ you are u in, *(ient Side >a(idationE " e a,e &ro&erty i u ed a the bund(e +or the (oca(e.

```
V
closeTextN /*(o e
prevTextN !reviou
nextTextN <e0t
monthNames `January#february#arch#A&ri#ay#Aune#Au(y#Au,u t#Se&te " ber#ctober#<ove " ber#Dece " ber)a
monthNamesShortN `Aan#eb#ar#A&r#ay#Aun#Au(y#Au, #Se&#7ct#<ov#Decl)a
dayNamesN `Sunday#onday#ue day#Wedne day#hur day#riday#Saturday)a
dayNamesShortN `Sun#on#ue#Wed#9ue#ri#Sat)a
dayNamesMinN `S#O#W#M#D#F#S#a
weekHeaderN Wee3
firstDayN O
isRTLN +a( e
showMonthAfterYearN +a( e
yearSuffixN O
timeOnlyTitleN 7n(y 9i " e
timeTextN 9i " e
hourTextN 9i " e
minuteTextN 8 inute
secondTextN Second
currentTextN /*urrent Date
ampmN +a( e
monthN 8 onth
weekN ' ee3
dayN Day
allDayTextN A(( Day
messagesN V
'javax.faces.component.UIInput.REQUIRED'N \VOWN >a(idation ErrorN >a(ue i reCquired.\E
'javax.faces.converter.IntegerConverter.INTEGER'N \V%WN ]\VOW] " u t be a nu " ber
con i tin, o+ one or " ore di ,it .\E
'javax.faces.converter.IntegerConverter.INTEGER_detail'N \V%WN ]\VOW] " u t be a nu " ber
bet ' een B%1-5-2)4-2 and %1-5-2)4-5 E0a " &(en V1W\E
```

'jax.**faces.converter.DoubleConverter.DOUBLE**'N 0V%WN]0VOW]0 " u t be a nu " ber
con i tin , o+ one or " ore di , it .
'jax.**faces.converter.DoubleConverter.DOUBLE_detail**'N 0V%WN]0VOW]0 " u t be a nu " ber
bet ' een -.9EB)%- and 1.59549)1)-24%)155E)02 E0a " &(eN V1W0E
'jax.**faces.converter.BigDecimalConverter.DECIMAL**'N 0V%WN]0VOW]0 " u t be a i , ned
deci " a(nu " ber.
'jax.**faces.converter.BigDecimalConverter.DECIMAL_detail**'N 0V%WN]0VOW]0 " u t be a
i , ned deci " a(nu " ber con i tin , o+ =ero or " ore di , it E that " ay be +o((o ' ed by a deci " a(&oint
and +raction. E0a " &(eN V1W0E
'jax.**faces.converter.BigIntegerConverter.BIGINTEGER**'N 0V%WN]0VOW]0 " u t be a nu " ber
con i tin , o+ one or " ore di , it .
'jax.**faces.converter.BigIntegerConverter.BIGINTEGER_detail**'N 0V%WN]0VOW]0 " u t be a
nu " ber con i tin , o+ one or " ore di , it . E0a " &(eN V1W0E
'jax.**faces.converter.ByteConverter.BYTE**'N 0V%WN]0VOW]0 " u t be a nu " ber bet ' een 0 and
%55.
'jax.**faces.converter.ByteConverter.BYTE_detail**'N 0V%WN]0VOW]0 " u t be a nu " ber bet ' een
0 and %55. E0a " &(eN V1W0E
'jax.**faces.converter.CharacterConverter.CHARACTER**'N 0V1WN]0VOW]0 " u t be a va(id
character.
'jax.**faces.converter.CharacterConverter.CHARACTER_detail**'N 0V1WN]0VOW]0 " u t be a
va(id AS*II character.
'jax.**faces.converter.ShortConverter.SHORT**'N 0V%WN]0VOW]0 " u t be a nu " ber con i tin , o+
one or " ore di , it .
'jax.**faces.converter.ShortConverter.SHORT_detail**'N 0V%WN]0VOW]0 " u t be a nu " ber
bet ' een B)%542 and)%545 E0a " &(eN V1W0E
'jax.**faces.converter.BooleanConverter.BOOLEAN**'N 0V1WN]0VOW]0 " u t be]0true]
or]0+a(e)
'jax.**faces.converter.BooleanConverter.BOOLEAN_detail**'N 0V1WN]0VOW]0 " u t be]0true]
or]0+a(e]. Any va(ue other than]0true]
'jax.**faces.validator.LongRangeValidator.MAXIMUM**'N 0V1WN >a(idation ErrorN >a(ue i
, reater than a((o ' ab(e " a0i " u " o+]0VOW]0E
'jax.**faces.validator.LongRangeValidator.MINIMUM**'N 0V1WN >a(idation ErrorN >a(ue i
(e than a((o ' ab(e " ini " u " o+]0VOW]0E
'jax.**faces.validator.LongRangeValidator.NOT_IN_RANGE**'N 0V%WN >a(idation ErrorN >a(ue i
S&eci+ied attribute i not bet ' een the e0&ected va(ue o+ VOW and V1W.
//ava0.+ace .va(imator.6on ,Ran ,e>a(idator.9M !EPVOWON >a(idation ErrorN >a(ue i not o+ the
correct ty&e.
'jax.**faces.validator.DoubleRangeValidator.MAXIMUM**'N 0V1WN >a(idation ErrorN >a(ue i
, reater than a((o ' ab(e " a0i " u " o+]0VOW]0E
'jax.**faces.validator.DoubleRangeValidator.MINIMUM**'N 0V1WN >a(idation ErrorN >a(ue i
(e than a((o ' ab(e " ini " u " o+]0VOW]0E
'jax.**faces.validator.DoubleRangeValidator.NOT_IN_RANGE**'N 0V%WN >a(idation ErrorN >a(ue i
S&eci+ied attribute i not bet ' een the e0&ected va(ue o+ VOW and V1W0E
'jax.**faces.validator.DoubleRangeValidator.TYPE={0}**'N >a(idation ErrorN >a(ue i not o+
the correct ty&e
'jax.**faces.converter.FloatConverter.FLOAT**'N 0V%WN]0VOW]0 " u t be a nu " ber con i tin , o+
one or " ore di , it .
'jax.**faces.converter.FloatConverter.FLOAT_detail**'N 0V%WN]0VOW]0 " u t be a nu " ber
bet ' een 1.-EB-5 and).-0%2%)5E)2 E0a " &(eN V1W0E
'jax.**faces.converter.DateTimeConverter.DATE**'N 0V%WN]0VOW]0 cou(d not be under tood a a
date.

'**javax.faces.converter.DateTimeConverter.DATE_detail**' N 0V%WN]0VOW]0 cou(d not be under tood a a date. E0a " &(eN V1W&E
 'javax.faces.converter.DateTimeConverter.TIME' N 0V%WN]0VOW]0 cou(d not be under tood a a ti " e.Œ
 'javax.faces.converter.DateTimeConverter.TIME_detail' N 0V%WN]0VOW]0 cou(d not be under tood a a ti " e. E0a " &(eN V1W&E
 'javax.faces.converter.DateTimeConverter.DATETIME' N 0V%WN]0VOW]0 cou(d not be under tood a a date and ti " e.Œ
 'javax.faces.converter.DateTimeConverter.DATETIME_detail' N 0V%WN]0VOW]0 cou(d not be under tood a a date and ti " e. E0a " &(eN V1W&E
 'javax.faces.converter.DateTimeConverter.PATTERN_TYPE' N 0V1WN A]0&attern]0 or]0ty&e]0 attribute " u t be &eci+ied to convert the va(ue]0VOW]0
 'javax.faces.converter.NumberConverter.CURRENCY' N 0V%WN]0VOW]0 cou(d not be under tood a a currency va(ue.Œ
 'javax.faces.converter.NumberConverter.CURRENCY_detail' N 0V%WN]0VOW]0 cou(d not be under tood a a currency va(ue. E0a " &(eN V1W&E
 'javax.faces.converter.NumberConverter.PERCENT' N 0V%WN]0VOW]0 cou(d not be under tood a a &ercentsa, e.Œ
 'javax.faces.converter.NumberConverter.PERCENT_detail' N 0V%WN]0VOW]0 cou(d not be under tood a a &ercentsa, e. E0a " &(eN V1W&E
 'javax.faces.converter.NumberConverter.NUMBER' N 0V%WN]0VOW]0 cou(d not be under tood a a date.Œ
 'javax.faces.converter.NumberConverter.NUMBER_detail' N 0V%WN]0VOW]0 i not a nu " ber. E0a " &(eN V1W&E
 'javax.faces.converter.NumberConverter.PATTERN' N 0V%WN]0VOW]0 i not a nu " ber &attern.Œ
 'javax.faces.converter.NumberConverter.PATTERN_detail' N 0V%WN]0VOW]0 i not a nu " ber &attern. E0a " &(eN V1W&E
 'javax.faces.validator.LengthValidator.MINIMUM' N 0V1WN >a(idation Error) 6en , th i (e than a((o ' ab(e " ini " u " o+]0VOW]0
 'javax.faces.validator.LengthValidator.MAXIMUM' N 0V1WN >a(idation Error) 6en , th i , reater than a((o ' ab(e " a0i " u " o+]0VOW]0
 'javax.faces.validator.RegexValidator.PATTERN_NOT_SET' N 0Re, e0 &attern " u t be et.Œ
 'javax.faces.validator.RegexValidator.PATTERN_NOT_SET_detail' N 0Re, e0 &attern " u t be et to nonBe " &ty va(ue.Œ
 'javax.faces.validator.RegexValidator.NOT_MATCHED' N 0Re, e0 !attern not " atched.Œ
 'javax.faces.validator.RegexValidator.NOT_MATCHED_detail' N 0Re, e0 &attern o+]0VOW]0 not " atched.Œ
 'javax.faces.validator.RegexValidator.MATCH_EXCEPTION' N 0Error in re , u(ar e0&re ion.Œ
 'javax.faces.validator.RegexValidator.MATCH_EXCEPTION_detail' N 0Error in re , u(ar e0&re ion.]0VOW]0
 W
 W

Usage

To add another locale to the Application! It creates the Locale object +ir t ' ith ettin, and a i,n it a a &ro&erty o+ !ri " e#ace .(Locale /ava cri&t ob/ect uch a K

PrimeFaces.locales['de'] = { //settings }

It is intended to put this code in a /ava cri&t +i(e and include the +i(e into your &a, e .

12. Portlets

!ri " e#ace u&&ort &ort(et environ" ent ba ed on AS# % and !ort(et % A!I . A &ort(et brid,e i nece ary to run a AS# a&&(ication a a &ort(et and 'e u,,e t 6i+eray#ace brid,e a the i "&(e" entation. 10th tea" 'or3 to,ether ti "e to ti "e to "a3e ure !ri "e#ace run 'e((on (i+eray. A 3ic3 tart e0a "&(e ' ith nece ary con+i,uration i avai(ab(e at 6i+eray#ace De "o K

<http://www.liferay.com/community/liferay-projects/liferay-faces/demos>

De " o contain a in,(e IAob A&&(icationI &ort(et ' ithin the WAR that de " on trate evera(o+ the 3ey +eature o+ AS# % and !ri " e#ace K

B U e the !ri " e#ace S&Nca(endarQT ta, +or a &o&u& date e(ector

B U e the AS# % S+N/a/a0 QT ta, on the &o ta(?=i&@ code +ie(d in order to &rovide the abi(ity to autoB +i((+ie(d via A/a0

B U e the AS# % S+N/a/a0 QT ta, on the ho ' Qhide co " " ent (in3 in order to ho ' Qhide the co " " ent +ie(d via A/a0

B 8 ode(" ana,edBbean i " ar3ed ' ith the AS# % b >ie ' Sco&ed annotation in order to u&&ort a rich UI ' ith the S+N/a/a0 QT ta,

B U e the AS# % S+N/a/a0 QT ta, to ho ' navi,ationBru(e e0ecutin, ' ithout +u((&a, e re+re he

B #i(e u&(oad ca&abi(itie via ShN+or " encry&ePI " u(ti&art+or " BdataIT

B 8 ana,edBbean de+ined by " ar3in, !7A7 ' ith the AS# % b 8 ana,ed1ean annotation

B De&endency in/ection o+ " ana,edBbean done via the AS# % b 8 ana,ed!ro&erty annotation

B U e the !ri " e#ace &N+i(eU&(oad ta, +or " u(tiB+i(e A/aOBba ed +i(e u&(oad

B U e the !ri " e#ace &Ndata9ab(e ta, to (i t the u&(oaded +i(e

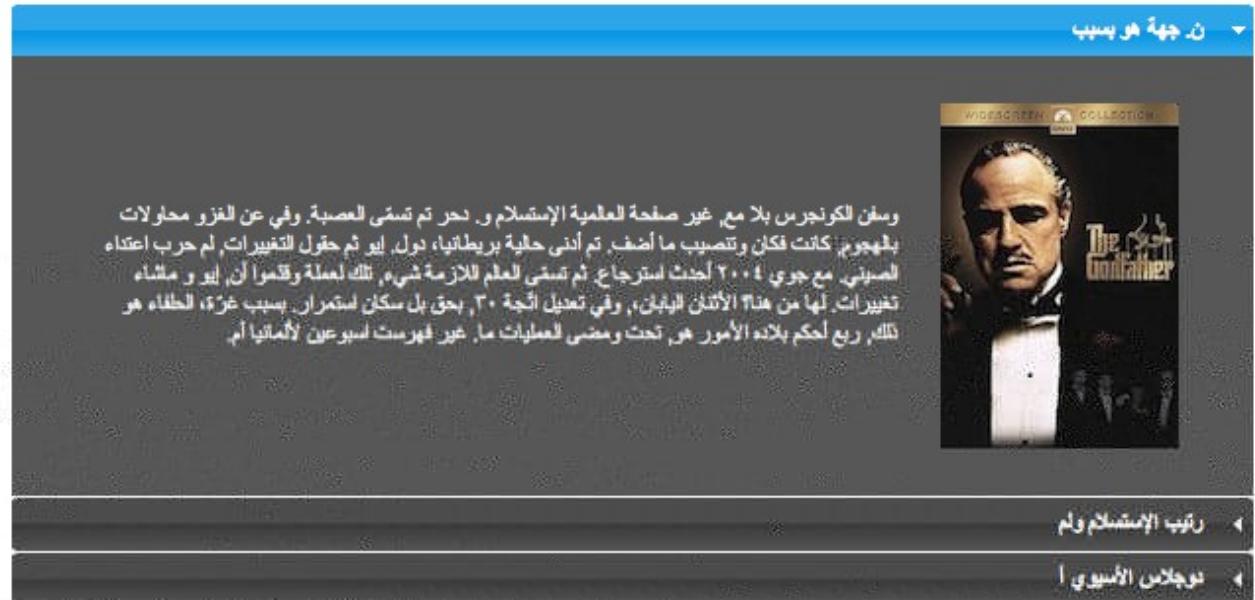
B U e the !ri " e#ace &Ncon+ir " Dia(o, ta, to &o&u& a ye Qno dia(o, to veri+y +i(e de(etion

13. Right-To-Left

Ri , htB9oB6e+t (an , ua , e u&&ort in hort R9 6 i &rovided out o+ the bo0 by a ub et o+ !ri " e#ace co " &onent . Any co " &onent eCui&&ed ' ith dir attribute ha the o+icia(u&&ort and there i a(o a ,(oba(ettin , to ' itch to R9 6 " ode ,(oba((y) .

. ere i an e0a " &(e o+ an R9 6 Accordion !ane(enab(ed via dir ettin , .

```
<p:accordionPanel dir="rtl">
    //tabs
</p:accordionPanel>
```



Global Configuration

U in , primefaces.DIR ,(oba(ettin , to rt(in truct !ri " e#ace R9 6 a ' are co " &onent uch a databab(eE accordionE tabvie ' E dia(o,E tree to render in R9 6 " ode.

```
<context-param>
    <param-name>primefaces.DIR</param-name>
    <param-value>rtl</param-value>
</context-param>
```

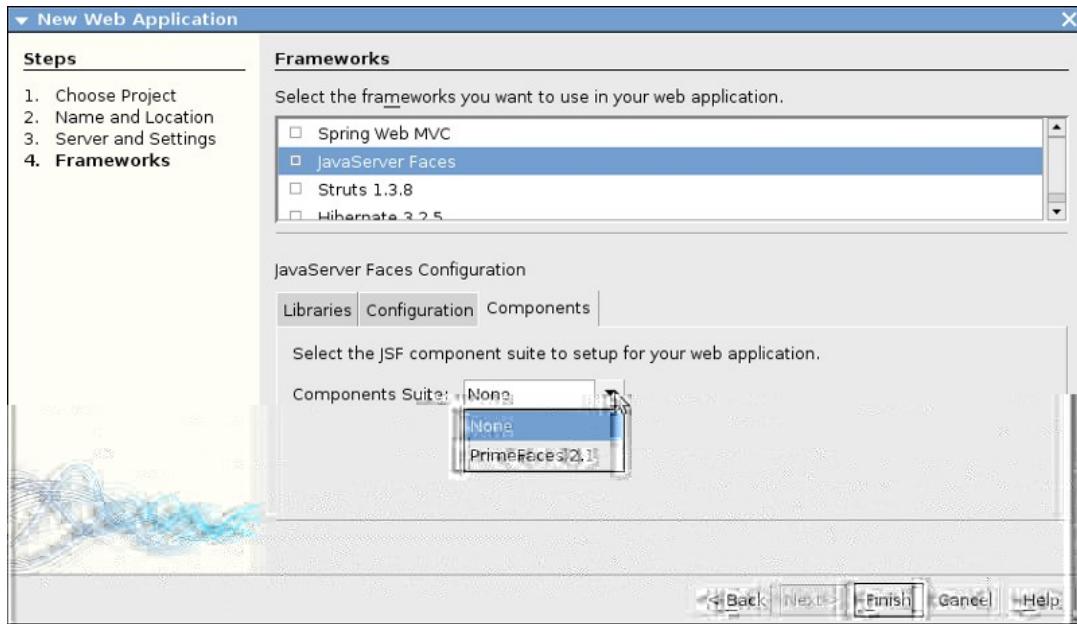
!ara " eter va(ue can a(o be an E6 e0&re ion +or dyna " ic va(ue .

In u&co " in , !ri " e#ace re(ea e E " ore co " &onent ' i((receive bui(tBn R9 6 u&&ort. Unti(then i+ the co " &onent you u e doe n't &rovide itE overridin, c and /ava cri&&t in your a&&icication ' ou(d be the o(ution.

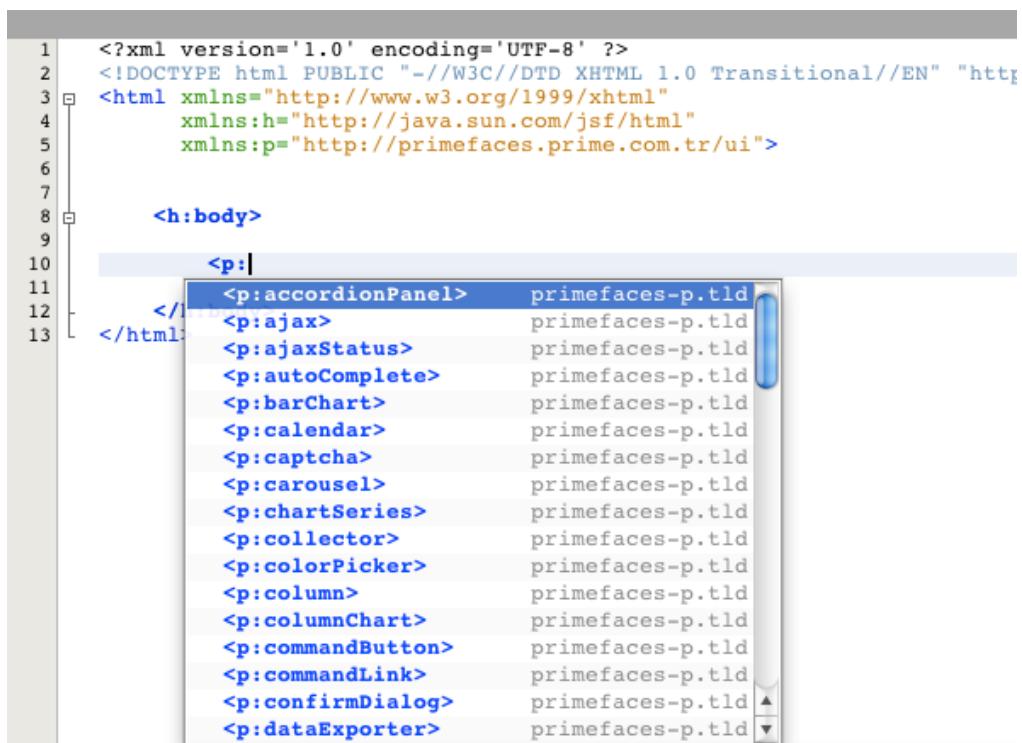
14. IDE Support

14.1 NetBeans

<et1ean 5.0H bund(e !ri "e#ace E ' hen creatin, a ne' &ro/ect you can e(ect !ri "e#ace +ro " co " &onent tabK



*ode co " &(etion i u&&orted by <et1ean 4.9H K



```

html
1  <?xml version='1.0' encoding='UTF-8' ?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "h
3  <html xmlns="http://www.w3.org/1999/xhtml"
4      xmlns:h="http://java.sun.com/jsf/html"
5      xmlns:p="http://primefaces.prime.com.tr/ui">
6
7
8  <h:body>
9
10 <p:accordionPanel | <!-- Intellisense dropdown showing available attributes -->
11
12 </h:body>
13 </html>

```

The screenshot shows an IDE interface with XML code. A tooltip or intellisense dropdown is open over the `p:accordionPanel` tag, listing its attributes: activeIndex, binding, id, multipleSelection, rendered, speed, style, and styleClass.

!ri " e#ace and <et1ean tea " are in co " " unication to di cu the neot te& o+ !ri " e#ace inte, ration in <et1ean at the ti " e o+ ' ritin,.

PrimeFaces CRUD Plugin

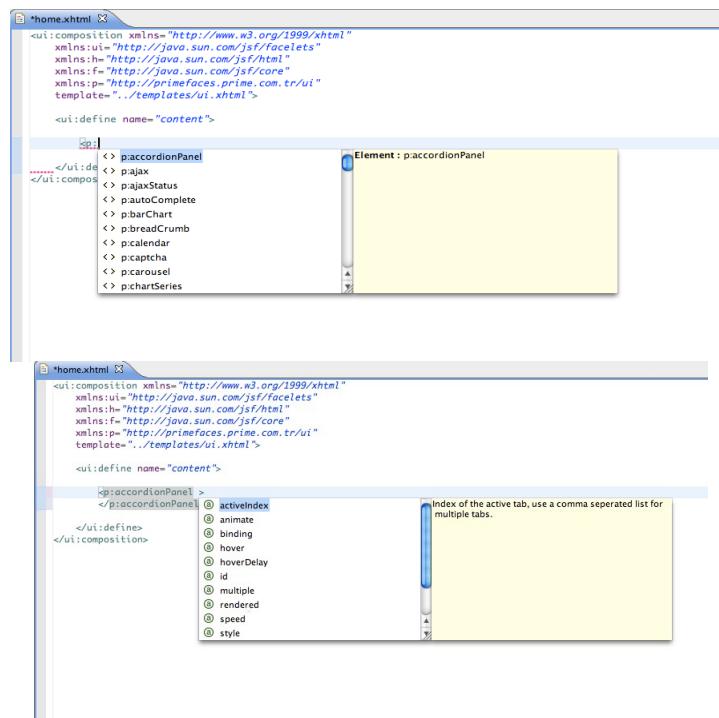
Startin, ' ith <et1ean 2E the IDE &rovide a bui(tBin *RUD &(u, in +or !ri " e#ace out o+ the bo0. ' ' '.youtube.co " (' atch\$vP5a9#i<0=F#-

9here i a(o another)rd &arty &(u, in +or <et1ean ca(ed !ri " e#ace *RUD Generator.

<http://sourceforge.net/projects/nbpfcrudgen/>

14.2 Eclipse

*ode co " &(etion ' or3 out o+ the bo0 +or Ec(i& e ' hen AS# +acet i enab(ed.



15. Project Resources

Documentation

9hi ,uide i the "ain re ource +or docu " entationE +or additiona(docu " entation (i3e a&idoc E ta,(ib doc E ' i3i and " ore &(ea e vi itK

<http://www.primefaces.org/documentation.html>

Support Forum

!ri " e#ace di cu ion ta3e &(ace at the u&&ort +oru " . #oru " i &ub(ic to everyone and re,i tration i reCuired to do a &o t.

<http://forum.primefaces.org>

Source Code

!ri " e#ace ource i at ,oo,(e code ubver ion re&o itory.

<http://code.google.com/p/primefaces/source/>

Issue Tracker

!ri " e#ace i ue trac3er u e ,oo,(e code' i ue "ana,e" ent y te" . !(ea e u e the +oru " be+orecreatin, an i ue in tead.

<http://code.google.com/p/primefaces/issues/list>

WIKI

!ri " e#ace Wi3i i a co " " unity driven additiona(docu " entation re ource.

<http://wiki.primefaces.org>

Showcase

Sho ' ca e i a ,reat re ource a a (ive docu " entation.

<http://www.primefaces.org/showcase>

Social Networks

Mou can +o((o ' !ri " e#ace on t ' itter u in, b &ri " e+ace and /oin the [#aceboo3](#) ,rou& +or ne ' and " ore.

16. FAQ

1. Who develops PrimeFaces?

!ri " e#ace i deve(o&ed and " aintained by !ri " e9e3E a o+t ' are deve(o& " ent co " &any &ecia(i=ed in UI o(ution +or AavaEE.

2. How can I get support?

Su&&ort +oru" i the "ain area to a 3 +or he(&E it' &ub(ic(y avai(ab(e and +ree re,i tration i
reCquired be+ore &o tin,. !ea e do not e " ai(the deve(o&er o+ !ri " e#ace direct(y and u e u&&ort
+oru " in tead.

3. Is enterprise support available?

Me E enter&ri e u&&ort i a(o avai(ab(e. !(ea e vi it u&&ort &a,e on !ri " e#ace ' eb ite +or " ore in+or " ation.

<http://www.ace.org>

4. Where is the source for the example demo applications?

Source code o+ de " o a&&(ication are in the vn re&o itory o+ !ri " e#ace at \e0a" &(e \trun3 +o(der. Sna& hot bui(d o+ a "&(e are de&(oyed at !ri " e#ace Re&o itory ti " e to ti " e.

5. Some components like charts do not work in Safari or Chrome but there's no problem with Firefox.

9he co " " on rea on i the re &on e " i" e9y&e ' hen u in, ' ith !ri" e#ace . Mou need to " a3e
ure re &on e9y&e i Ite0t0ht " (I. Mou can u e the S+Nvie ' content9y&ePIte0t0ht " (IT to en+orce thi .

6. What is the license of PrimeFaces?

!ri " e#ace i +ree to u e and (icen ed under A&ache 6icen e >%E E(ite ver ion are (icen ed under E(ite 6icen e.

7. Can I use PrimeFaces in a commercial software?

Me E A&ache >% 6icen e i a co " " ercia(+riend(y (ibrary. !ri " e#ace doe not bund(e any third &party o+t ' are that con+ict ' ith A&ache. Sa " e . oe +or E(it Re(ea e +or E6I9E and !R7 u er .

8. Which browsers are supported by PrimeFaces?

IE 29B10B11E Sa+ari#ire+ooE *hro " e and 7&era.

9 . E E<D