



PRIMEFACES MOBILE USER'S GUIDE

Author

Çağatay Çivici

Covers 0.9.0

*This guide is dedicated to my wife Nurcan,
without her support PrimeFaces wouldn't exist.*

Çağatay Çivici

About the Author	5
Introduction	6
1.1 What is PrimeFaces Mobile?	6
1.2 Prime Teknoloji	6
2. Setup	7
2.1 Download	7
2.2 Dependencies	7
2.3 Configuration	8
2.4 Taglib	8
2.5 RenderKit	8
3. Pages	9
4. Navigations	11
4.1 Local Navigations	11
4.2 External Navigations	13
5. Ajax	14
6. Mobile Components	15
6.1 ButtonGroup	15
6.2 Content	17
6.3 Field	18
6.4 Footer	19
6.5 Header	20
6.6 InputRange	22
6.7 NavBar	24
6.8 Page	26

6.9 Switch	27
6.10 View	29
7. RenderKit	30
7.1 h:outputLink	30
7.2 h:panelGrid	30
7.3 h:form	30
7.4 p:button	30
7.5 p:commandButton	31
7.6 p:dataList	31
7.7 p:inputText	33
7.8 p:inputTextarea	33
7.9 p:panel	33
7.10 p:accordionPanel	34
7.11 p:selectBooleanCheckbox	34
8. Themes	35
8.1 Theme Framework	35
8.2 Creating a Custom Theme	36
9. Samples	37
10. Supported Platforms	38

About the Author

Çağatay Çivici is a member of JavaServer Faces Expert Group, the founder and project lead of PrimeFaces and PMC member of open source JSF implementation Apache MyFaces. He's a recognized speaker in international conferences including SpringOne, Jazoon, JAX, W-JAX, JSFSummit, JSFDays, Con-Fess and many local events such as JUGs.

Çağatay is also an author and technical reviewer of a couple books regarding web application development with Java and JSF. As an experienced trainer, he has trained over 300 developers on Java EE technologies mainly JSF, Spring, EJB 3.x and JPA.

Çağatay is currently working as a principal consultant for Prime Teknoloji, the company he co-founded in Turkey.

1. Introduction

1.1 What is PrimeFaces Mobile?

PrimeFaces Mobile is a UI kit to implement JSF pages that are optimized for mobile devices with a native look and feel.

- Mobile RenderKit for standard JSF and core PrimeFaces components.
- Mobile JSF components.
- Same backend model for desktop web and mobile web.
- Wide range of platform support.
- Ajax features to bring native application experience.
- Nothing required to install on device.
- Powered by jQuery Mobile

1.2 Prime Teknoloji

PrimeFaces Mobile is maintained by Prime Teknoloji, a Turkish software development company specialized in Agile and Java EE consulting. PrimeFaces Team members are full time engineers at Prime Teknoloji.

- Çağatay Çivici - Architect and Lead Developer
- Levent Günay - Core Developer / QA&Test
- Yiğit Darçın - Core Developer / QA&Test
- Mustafa Daşgın - Core Developer / QA&Test
- Basri Kahveci - QA&Test
- Deniz Silahçılar - QA&Test
- Cenk Çivici - Mentor

2. Setup

2.1 Download

PrimeFaces Mobile has a single jar called **mobile-{version}.jar**. There are two ways to download this jar, you can either download from PrimeFaces homepage or if you are a maven user you can define it as a dependency.

Download Manually

Three different artifacts are available for each PrimeFaces Mobile version, binary, sources and bundle. Bundle contains binary, sources and javadocs.

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

Download with Maven

Group id of the dependency is *org.primefaces* and artifact id is *primefaces*.

```
<dependency>
  <groupId>org.primefaces</groupId>
  <artifactId>mobile</artifactId>
  <version>0.9.0</version>
</dependency>
```

In addition to the configuration above you also need to add PrimeFaces maven repository to the repository list so that maven can download it.

```
<repository>
  <id>prime-repo</id>
  <name>Prime Repo</name>
  <url>http://repository.primefaces.org</url>
</repository>
```

2.2 Dependencies

PrimeFaces Mobile only requires a JAVA 5+ runtime, a JSF 2.0 implementation and PrimeFaces Core as mandatory dependencies. Following is the compatibility matrix between core and mobile.

Mobile	Core
0.9.0	3.0

2.3 Configuration

PrimeFaces Mobile does not require any mandatory configuration.

2.4 Taglib

PrimeFaces Mobile component library namespace should be added to your mobile pages.

```
xmlns:pm="http://primefaces.org/mobile">
```

2.5 RenderKit

PrimeFaces Mobile provides a Mobile Renderkit for some standard JSF and core PrimeFaces Components, in order to run PrimeFaces Mobile, this render kit should be enabled. There are a couple of ways listed below you can choose. Id of the renderkit is “PRIMEFACES_MOBILE”.

Request Parameter

Access a mobile page with `javax.faces.RenderKitId` parameter. This approach is for quick testing and not suggested as maintaining a request parameter is not easy.

```
http://www.yourapp.com?javax.faces.RenderKitId=PRIMEFACES\_MOBILE
```

(Suggested) Configure Application Wide

Define in `faces-config.xml`. This method is suggested if your mobile pages are in a different application than the non-mobile pages like `http://mobile.yourapp.com`.

```
<application>
  <default-render-kit-id>PRIMEFACES_MOBILE</default-render-kit-id>
</application>
```

Decide Yourself

Write a viewhandler by overriding `calculateRenderKitId` API and decide when to display the page in mobile mode. This approach is suggested if your mobile and non-mobile pages are in same application and you need to switch renderkits on-the-fly.

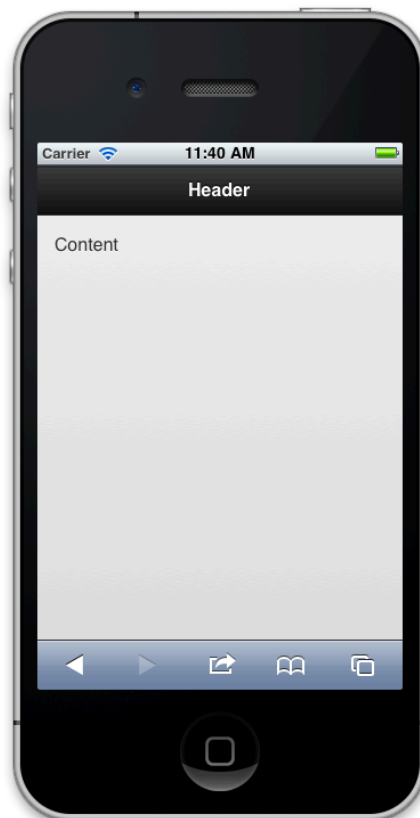
3. Pages

A mobile page is a simple xhtml based JSF view that can have of one or more views. A view is a screen in your layout. Following is a simple example having page, view, header and content components.

```
<f:view xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui"
  xmlns:pm="http://primefaces.org/mobile"
  contentType="text/html">

  <pm:page title="Hello World">
    <pm:view id="main">
      <pm:header title="Header" />

      <pm:content>
        Content
      </pm:content>
    </pm:view>
  </pm:page>
</f:view>
```



You can add more views by using a similar approach using pm:view component. Following is a multi view page.

```

<f:view xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui"
  xmlns:pm="http://primefaces.org/mobile"
  contentType="text/html">

  <pm:page title="Hello World">
    <pm:view id="main">
      <pm:header title="Header" />

      <pm:content>
        Content
      </pm:content>
    </pm:view>

    <pm:view id="second">
      <pm:header title="Second View Header" />

      <pm:content>
        Second View Page
      </pm:content>
    </pm:view>

    <pm:view id="third">
      <pm:header title="Third View Header" />

      <pm:content>
        Third View Page
      </pm:content>
    </pm:view>

  </pm:page>
</f:view>

```

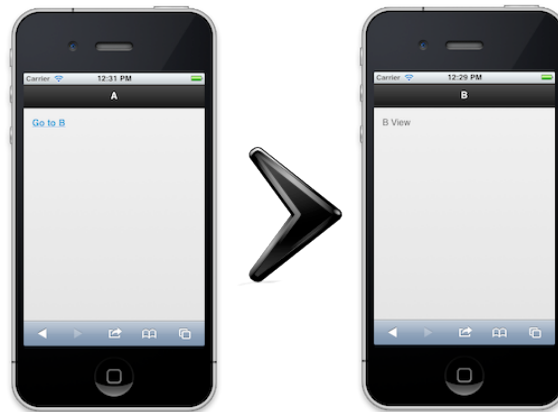
By default, first view on page is displayed initially.

4. Navigations

Navigation in PrimeFaces Model can be between local views with or without ajax and to an external page outside of application.

4.1 Local Navigations

Local navigation means navigation from a view to another view on same page.



GET Style

This type of navigation behaves like a GET navigation although there is no actual HTTP request.

```
<f:view xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui"
  xmlns:pm="http://primefaces.org/mobile"
  contentType="text/html">

  <pm:page title="Mobile">
    <pm:view id="viewA">
      <pm:header title="A" />

      <pm:content>
        <h:outputLink value="#viewB">Go to B</h:outputLink>
      </pm:content>
    </pm:view>

    <pm:view id="viewB">
      <pm:header title="B" />

      <pm:content>
        B View
      </pm:content>
    </pm:view>
  </pm:page>

</f:view>
```

POST Style

This navigation approach integrated with JSF Navigation Model is used when you do an ajax request and need to navigate to another view.

```
<f:view xmlns="http://www.w3.org/1999/xhtml"
  xmlns:f="http://java.sun.com/jsf/core"
  xmlns:p="http://primefaces.org/ui"
  xmlns:pm="http://primefaces.org/mobile"
  contentType="text/html">

  <pm:page title="Mobile">
    <pm:view id="viewA">
      <pm:header title="A" />

      <pm:content>
        <h:form>
          <p:commandButton value="Go" action="#{bean.method}" />
        </h:form>
      </pm:content>
    </pm:view>

    <pm:view id="viewB">
      <pm:header title="B" />

      <pm:content>
        B View
      </pm:content>
    </pm:view>
  </pm:page>

</f:view>
```

```
public String method() {
    return "pm:viewB";
}
```

When the outcome is prefixed by “pm:”, PrimeFaces Mobile navigates to the View B after ajax request is completed. You can also use the outcome directly as;

```
<p:commandButton value="Go" action="pm:viewB" />
```

Reverse Effect

By default, slide forward animation is used to navigate between local views, in case you need to display slide backwards, append reverse=true string to your outcome.

```
<p:commandButton value="Go back to B" action="pm:viewB?reverse=true" />
<h:outputLink value="#viewB?reverse=true">Go back to B</h:outputLink>
```

4.2 External Navigations

Navigation from a mobile application to an external resource is not different then doing it in a regular web page.

```
<h:outputLink value="http://www.primefaces.org">  
    Go Prime  
</h:outputLink>
```

5. Ajax

A mobile page can be enhanced with ajax features using regular PrimeFaces Ajax Framework. Important note is avoid updating whole views but the contents of the views. Following is a simple example;

```
<pm:page title="Mobile">
  <pm:view id="main">
    <pm:header title="Ajax" />

    <pm:content>
      <h:form>
        <p:inputText value="#{bean.text}" />
        <p:commandButton value="Update" update="display" />
        <h:outputText id="display" value="#{bean.text}" />
      </h:form>
    </pm:content>
  </pm:view>
</pm:page>
```

PrimeFaces Ajax extension features a special integration with jQuery Mobile's progressively enhanced UI controls and built-in ajax status dialog whereas core JSF ajax does not. If you update a component with core JSF ajax, you will lose the mobile optimized UI. To avoid this always use PrimeFaces Ajax instead. Updating a component on another view is easy with proper referencing as well, here is an example;

```
<pm:page title="Mobile">
  <pm:view id="viewA">
    <pm:header title="A" />
    <pm:content>
      <h:form>
        <p:inputText value="#{bean.text}" />
        <p:commandButton value="Update" update=":formB:display" />
      </h:form>
    </pm:content>
  </pm:view>

  <pm:view id="viewB">
    <pm:header title="B" />
    <pm:content>
      <h:form id="formB">
        <h:outputText id="display" value="#{bean.text}" />
      </h:form>
    </pm:content>
  </pm:view>
</pm:page>
```

Note that commandButton is referencing to display component using naming container separator as prefix since display component is in another naming container(form). Views themselves are not naming containers.

6. Mobile Components

PrimeFaces Mobile provides a set of mobile components that are specific to mobile platforms.

6.1 ButtonGroup

ButtonGroup component groups a set of buttons.



Info

Tag	buttonGroup
Component Class	org.primefaces.mobile.component.buttongroup.ButtonGroup
Component Type	org.primefaces.mobile.ButtonGroup
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.ButtonGroupRenderer
Renderer Class	org.primefaces.mobile.component.buttongroup.ButtonGroupRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
orientation	vertical	String	Type of the orientation, can be vertical or horizontal.
style	null	String	Inline style of the component.
styleClass	null	String	Style class of the component.

Getting started with ButtonGroup

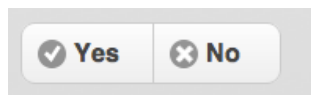
Buttons to be grouped are placed inside the button group.

```
<pm:buttonGroup>  
  <p:commandButton value="Yes" />  
  <p:commandButton value="No" />  
</pm:buttonGroup>
```

Orientation

Default orientation is *vertical* and other option is *horizontal*.

```
<pm:buttonGroup orientation="horizontal">  
  <p:commandButton value="Yes" />  
  <p:commandButton value="No" />  
</pm:buttonGroup>
```



6.2 Content

Content is used the place the contents of a mobile view.

Info

Tag	content
Component Class	org.primefaces.mobile.component.content.Content
Component Type	org.primefaces.mobile.Content
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.ContentRenderer
Renderer Class	org.primefaces.mobile.component.content.ContentRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean

Getting started with Content

See section 3 (Pages) for more information on how content component is used.

6.3 Field

Field component is used to align a label and a related input.

Info

Tag	field
Component Class	org.primefaces.mobile.component.field.Field
Component Type	org.primefaces.mobile.Field
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.FieldRenderer
Renderer Class	org.primefaces.mobile.component.field.FieldRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean

Getting started with Field

Field component wraps the label and the input.

```
<pm:field>
  <h:outputLabel for="input" value="Input: "/>
  <h:inputText id="inputText" value="#{bean.value}"/>
</pm:field>
```

Note that not all platforms due the screen size differences support field component.

Integrated Field

Field alignment is a built-in feature of PrimeFaces Mobile Renderkit, same can be written as;

```
<p:inputText id="inputText" value="#{bean.value}" label="Input: "/>
```

6.4 Footer

Footer is used as a bottom content container of a view.

Info

Tag	footer
Component Class	org.primefaces.mobile.component.footer.Footer
Component Type	org.primefaces.mobile.Footer
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.FooterRenderer
Renderer Class	org.primefaces.mobile.component.footer.FooterRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
fixed	FALSE	Boolean	Whether or not the position of footer is fixed.

Getting started with Footer

To begin with, see section 3 (Pages) for more information on how content component is used. A footer is used inside a mobile view.

```
<pm:view id="main">
  <pm:header title="Header" />

  <pm:content>
    Content
  </pm:content>

  <pm:footer>
    Footer Content
  </pm:footer>
</pm:view>
```

6.5 Header

Header is used as a top content container of a view.

Info

Tag	header
Component Class	org.primefaces.mobile.component.header.Header
Component Type	org.primefaces.mobile.Header
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.HeaderRenderer
Renderer Class	org.primefaces.mobile.component.header.HeaderRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
title	null	String	Title text of the header.
fixed	FALSE	Boolean	Whether or not the position of footer is fixed.

Getting started with Footer

As described in section 3, a header is defined within a view.

```
<pm:view id="main">
  <pm:header title="Header" />

  <pm:content>
    Content
  </pm:content>

  <pm:footer>
    Footer Content
  </pm:footer>
</pm:view>
```

Fixed Position

When fixed position is enabled, header scrolls along with the view.

```
<pm:header title="Header" fixed="true"/>
```

Left and Right Content

Placing buttons at left and right side of a header is common case in a mobile view. Header provides *left* and *right* facets to implement this.

```
<pm:header title="Prime">
  <f:facet name="left">
    <p:button value="Back" icon="back" href="#main"/>
  </f:facet>
  <f:facet name="right">
    <p:button value="Settings" icon="gear" href="#settings"/>
  </f:facet>
</pm:header>
```



6.6 InputRange

InputRange is a slider to provide number input.

Info

Tag	inputRange
Component Class	org.primefaces.mobile.component.inputrange.InputRange
Component Type	org.primefaces.mobile.InputRange
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.InputRangeRenderer
Renderer Class	org.primefaces.mobile.component.inputrange.InputRangeRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
value	null	Object	Value of the component than can be either an EL expression of a literal text
converter	null	Converter /String	An el expression or a literal text 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 text, it must refer to a converter id
immediate	FALSE	Boolean	When set true, process validations logic is executed at apply request values phase for this component.
required	FALSE	Boolean	Marks component as required
validator	null	Method Expr	A method binding expression that refers to a method validationg the input
valueChangeListener	null	Method Expr	A method binding expression that refers to a method for handling a valuchangeevent
requiredMessage	null	String	Message to be displayed when required field validation fails.
converterMessage	null	String	Message to be displayed when conversion fails.
minValue	null	Number	0
maxValue	null	Number	100

Name	Default	Type	Description
style	null	String	Inline style of the component.
styleClass	null	String	Style class of the component.
step	1	Number	Step factor.
disabled	FALSE	Boolean	Disables the component when true.
label	null	string	User presentable name.

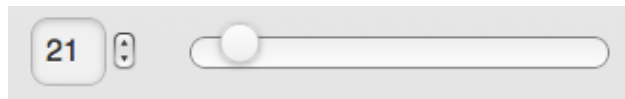
Getting started with InputRange

InputRange is an input component that requires a number value to bind.

```
public class Bean {
    private int value;

    //getter-setter
}
```

```
<pm:inputRange value="#{bean.value}" />
```



6.7 NavBar

NavBar is a grouping component for a set of buttons in a bar.

Info

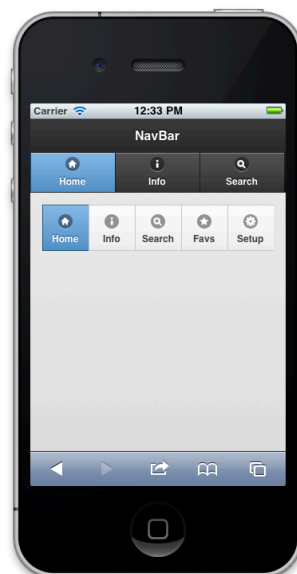
Tag	navBar
Component Class	org.primefaces.mobile.component.navbar.NavBar
Component Type	org.primefaces.mobile.NavBar
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.NavBarRenderer
Renderer Class	org.primefaces.mobile.component.navbar.NavBarRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean

Getting started with NavBar

NavBar can be placed inside a header or directly inside a content.




```

<pm:view id="navbar">
  <pm:header title="NavBar">
    <pm:navBar>
      <p:button value="Home" icon="home" href="#main?reverse=true"
        styleClass="ui-btn-active"/>
      <p:button value="Info" icon="info" href="#main?reverse=true" />
      <p:button value="Search" icon="search" href="#main?reverse=true" />
    </pm:navBar>
  </pm:header>

  <pm:content>
    <pm:navBar>
      <p:button value="Home" icon="home"
        href="#main?reverse=true" styleClass="ui-btn-active"/>
      <p:button value="Info" icon="info" href="#main?reverse=true" />
      <p:button value="Search" icon="search" href="#main?reverse=true" />
      <p:button value="Favs" icon="star" href="#main?reverse=true" />
      <p:button value="Setup" icon="gear" href="#main?reverse=true" />
    </pm:navBar>
  </pm:content>
</pm:view>

```

Adding ui-btn-active style class displays a button in a nav bar as active. Note that up to 5 buttons can be placed inside a navbar.

6.8 Page

Page is the core of a mobile page that contains one or more views and provides page wide options.

Info

Tag	page
Component Class	org.primefaces.mobile.component.page.Page
Component Type	org.primefaces.mobile.Page
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.PageRenderer
Renderer Class	org.primefaces.mobile.component.page.PageRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
title	FALSE	Boolean	Title of the page.
defaultPageTransition	slide	String	Name of animation to use in view navigation.
defaultDialogTransition	pop	String	Name of animation to use when displaying a dialog.
loadingMessage	loading	String	Text to display in built-in ajax status.
viewport	null	String	Meta settings for viewport.

Getting started with Page

See section 3 (Pages) for more information on how page component is basically used.

Event Facets

Page provides two facets name *preinit* and *postinit* that can be used to place content before jQuery mobile is initialized. For example *preinit* event is used to load custom themes, see section 8.2 for more information.

6.9 Switch

Switch is used to choose a binary selection.

Info

Tag	page
Component Class	org.primefaces.mobile.component.uiswitch.UISwitch
Component Type	org.primefaces.mobile.UISwitch
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.UISwitchRenderer
Renderer Class	org.primefaces.mobile.component.uiswitch.UISwitchRenderer

Attributes

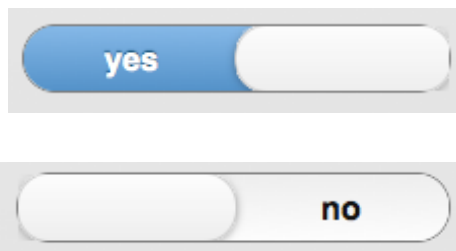
Name	Default	Type	Description
id	null	String	Unique identifier of the component
rendered	TRUE	Boolean	Boolean value to specify the rendering of the component, when set to false component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
value	null	Object	Value of the component than can be either an EL expression of a literal text
converter	null	Converter /String	An el expression or a literal text 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 text, it must refer to a converter id
immediate	FALSE	Boolean	When set true, process validations logic is executed at apply request values phase for this component.
required	FALSE	Boolean	Marks component as required
validator	null	Method Expr	A method binding expression that refers to a method validationg the input
valueChangeListener	null	Method Expr	A method binding expression that refers to a method for handling a valuchangeevent
requiredMessage	null	String	Message to be displayed when required field validation fails.
converterMessage	null	String	Message to be displayed when conversion fails.
onLabel	null	String	Text for true.
offLabel	null	String	Text for false.

Getting started with Switch

Switch is an input component that requires a boolean value to bind.

```
public class Bean {  
    private boolean value;  
  
    //getter-setter  
}
```

```
<pm:switch value="#{bean.value}" onLabel="Yes" offLabel="No"/>
```



6.10 View

View represents a screen in a mobile page.

Info

Tag	view
Component Class	org.primefaces.mobile.component.view.View
Component Type	org.primefaces.mobile.View
Component Family	org.primefaces.mobile.component
Renderer Type	org.primefaces.mobile.component.ViewRenderer
Renderer Class	org.primefaces.mobile.component.view.ViewRenderer

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 component will not be rendered.
binding	null	Object	An el expression that maps to a server side UIComponent instance in a backing bean
title	FALSE	Boolean	Title of the view.
swatch	null	String	Theme swatch.

Getting started with View

See section 3 (Pages) for more information on how view component is used.

7. RenderKit

In addition to the mobile specific components, PrimeFaces Mobile also provides Mobile optimized renderers for some core JSF and PrimeFaces components. Following is the list of components with mobile renderers. Note that mobile renderers may not support all the features of the component available in a desktop web environment.

7.1 h:outputLink

Core output link component is extended to integrate with PrimeFaces Mobile navigation model, see section 4 for more information about Navigations. For example to go from one view to another;

```
<h:outputLink value="#viewIdHere">Navigate</h:outputLink>
```

7.2 h:panelGrid

PanelGrid component is used to do create a grid layout with a table-less approach.

```
<h:panelGrid columns="4">
  <h:outputText value="Cell 1" />
  <h:outputText value="Cell 2" />
  <h:outputText value="Cell 3" />
  <h:outputText value="Cell 4" />

  <h:outputText value="Cell 5" />
  <h:outputText value="Cell 6" />
  <h:outputText value="Cell 7" />
  <h:outputText value="Cell 8" />
</h:panelGrid>
```

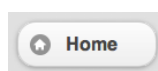
7.3 h:form

In a mobile environment, standard form component always sends `javax.faces.RenderKitId = PRIMEFACES_MOBILE` as a request parameter to set mobile renderkit to keep the mobile renderkit if you are using the request parameter approach defined in section 2.5

7.4 p:button

Button display is optimized for a mobile platform and integrated with mobile navigation model.

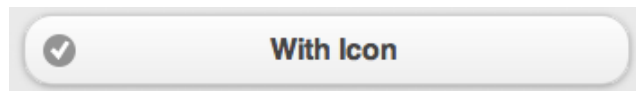
```
<p:button value="Home" icon="home" href="#main" />
```



7.5 p:commandButton

CommandButton display is optimized for a mobile platform and integrated with mobile navigation model. Core ajax features are also available.

```
<p:commandButton value="With Icon" icon="check" actionListener="#{bean.method}"
    update="othercomponent" global="false" />
```



7.6 p:dataList

DataList is an important component for Mobile featuring various display modes for data display. DataList can be used to display children only or iterate them for each data item.

ReadOnly

```
<p:dataList>
    <h:outputText value="Item 1" />
    <h:outputText value="Item 2" />
    <h:outputText value="Item 3" />
</p:dataList>
```

Item 1
Item 2
Item 3

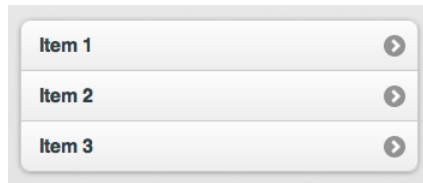
Links

```
<p:dataList>
    <h:outputLink value="#main">Item 1</h:outputLink>
    <h:outputLink value="#main">Item 2</h:outputLink>
    <h:outputLink value="#main">Item 3</h:outputLink>
</p:dataList>
```

Item 1	>
Item 2	>
Item 3	>

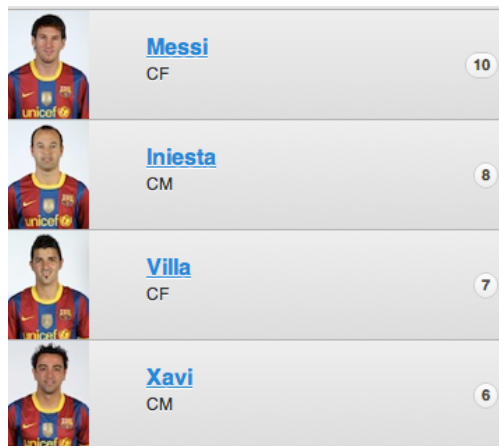
Inset

```
<p:dataList type="inset">
    <h:outputLink value="#main">Item 1</h:outputLink>
    <h:outputLink value="#main">Item 2</h:outputLink>
    <h:outputLink value="#main">Item 3</h:outputLink>
</p:dataList>
```



Custom Content

```
<p:dataList value="#{ringBean.players}" var="player">
    <p:graphicImage value="/images/barca/#{player.photo}" />
    <h3><h:outputLink value="#main">#{player.name}</h:outputLink></h3>
    <p>#{player.position}</p>
    <h:outputText styleClass="ui-li-count" value="#{player.number}" />
</p:dataList>
```



Notes on Mobile DataList

- Ajax pagination is not supported.
- Use PrimeFaces Ajax components or APIs to update the datalist, using core JSF ajax cannot update the datalist properly as explained in section 5 Ajax.
- Wrap the content inside a column component in case you do a selection. This is a requirement in core UIData to process selection properly.

7.7 p:inputText

InputText display is optimized for a mobile platform, in addition using type attribute you can customize the look and feel in supported browsers. For example following virtual display for the input text only displays numbers in iOS5.

```
<p:inputText id="age" type="number" value="#{bean.value}" />
```



7.8 p:inputTextarea

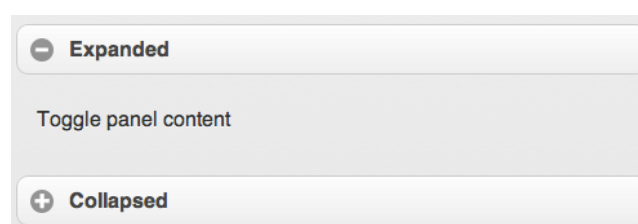
Rendering is optimized for a mobile platforms.

7.9 p:panel

PrimeFaces Panel component is toggleable by default in a mobile environment.

```
<p:panel header="Expanded">
    Toggle panel content
</p:panel>

<p:panel header="Collapsed" collapsed="true">
    Initially collapsed content
</p:panel>
```



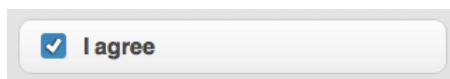
7.10 p:accordionPanel

```
<p:accordionPanel>
  <p:tab title="Title 1">
    Content 1
  </p:tab>
  <p:tab title="Title 2">
    Content 2
  </p:tab>
  <p:tab title="Title 3">
    Content 3
  </p:tab>
</p:accordionPanel>
```



7.11 p:selectBooleanCheckbox

```
<p:selectBooleanCheckbox value="#{true}" itemLabel="I agree" />
```



8. Themes

8.1 Theme Framework

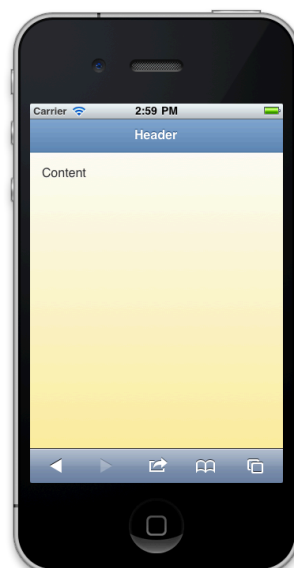
jQuery Mobile provides themes with multi swatches each having a different color scheme. PrimeFaces Mobile currently provides simple integration with this model and advanced integration is in the roadmap. View, Header and Footer components are equipped with swatch attribute to select the swatch to use in a mobile view. Default theme has five swatches; a, b, c, d and e.



Note that default swatch for view content is c and for header-footer it is a. Following is a simple page with b for header and e for content.

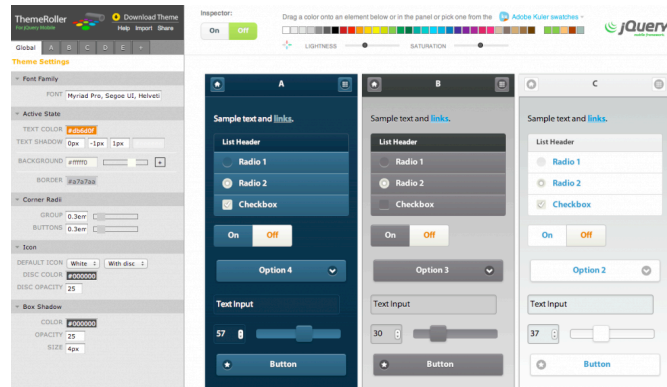
```
<pm:page title="Hello World">
  <pm:view id="main" swatch="e"
    <pm:header title="Header" swatch="b"/>

    <pm:content>
      Content
    </pm:content>
  </pm:view>
</pm:page>
```



8.2 Creating a Custom Theme

jQuery Mobile provides a mobile themeroller tool to create custom themes easily.



<http://jquerymobile.com/themeroller/>

1) Once you have created, give a name (e.g. optimus) and downloaded your theme from themeroller, place the contents of the zip file in your webapp like;

- %webroot%
 - themes
 - images
 - optimus.min.css

2) Configure PrimeFaces Mobile not to include default theme.

```
<context-param>
  <param-name>primefaces.mobile.THEME</param-name>
  <param-value>none</param-value>
</context-param>
```

3) Use the pre-init facet to place your theme.

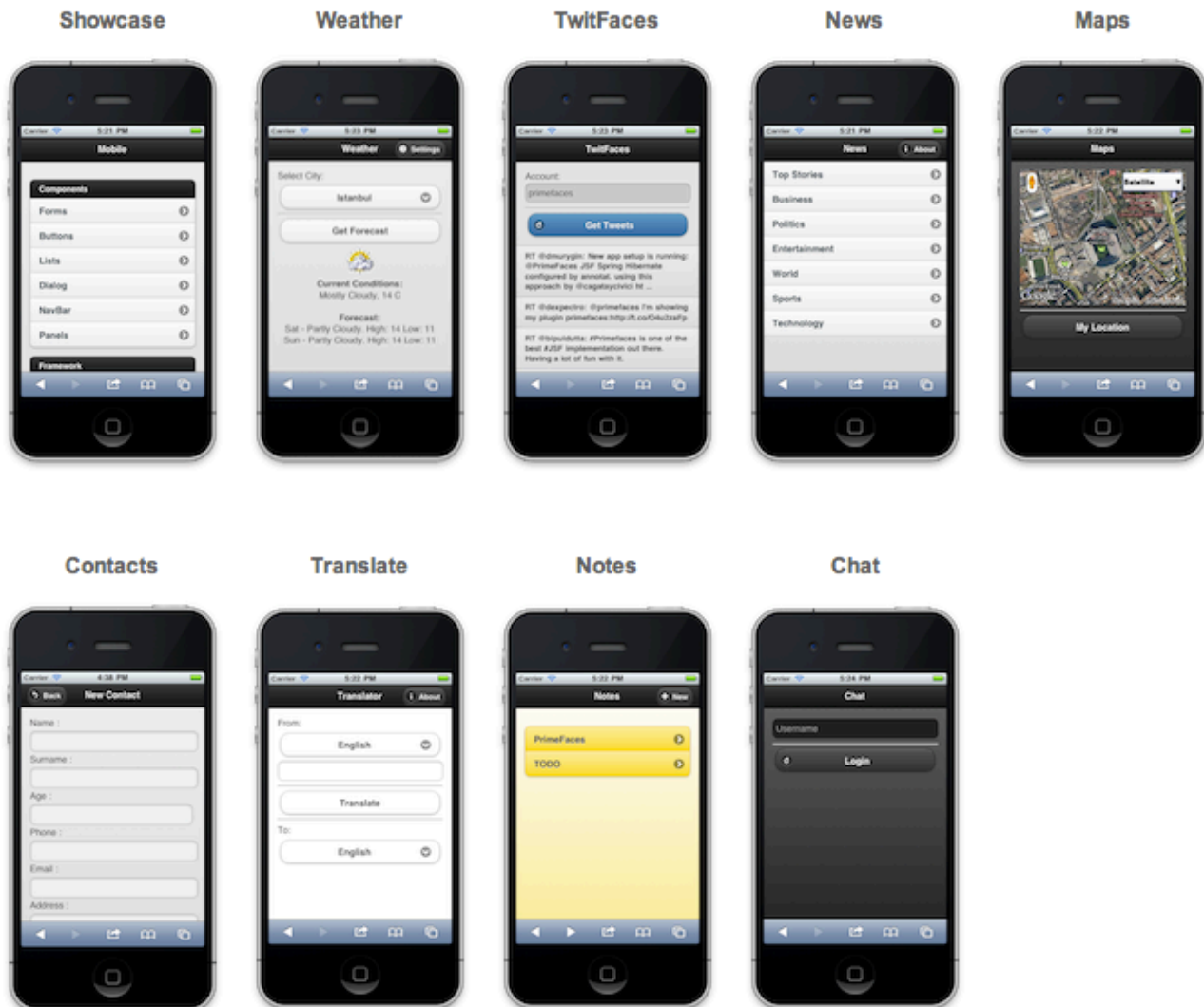
```
<pm:page>
  <f:facet name="preinit">
    <link type="text/css" rel="stylesheet"
          href="#{request.contextPath}/mobile/themes/optimus.css" />
  </f:facet>
</pm:page>
```

Do not use h:outputStylesheet as it can't load icons referenced from css.

4) Enjoy!

9. Samples

PrimeFaces Mobile has various samples on PrimeFaces Showcase demonstrating different features.



10. Supported Platforms

PrimeFaces Mobile is powered by jQuery Mobile which supports various mobile platforms based on a grade level.

See jQuery Mobile documentation for the detailed information;

<http://jquerymobile.com/gbs/>



THE END