Community Experience Distilled

PrimeFaces Blueprints

Create your very own portfolio of customized web applications with PrimeFaces

Sudheer Jonna
Ramkumar Pillai

[PACKT] PUBLISHING open source* community experience distilled

PrimeFaces Blueprints

Create your very own portfolio of customized web applications with PrimeFaces

Sudheer Jonna

Ramkumar Pillai

BIRMINGHAM - MUMBAI

PrimeFaces Blueprints

First published: August 2014

Production reference: 1190814

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78398-322-3

www.packtpub.com

Cover image by Benoit Benedetti (benoit.benedetti@gmail.com)

Credits

Authors

Sudheer Jonna

Ramkumar Pillai

Reviewers

Ramanath Chandramohan Bhongale

Aristides Villarreal Bravo

Vineet Jain

S V Narayana

Enrique Enolva Tan

Commissioning Editor

Akram Hussain

Acquisition Editor

Richard Harvey

Content Development Editor

Sankalp Pawar

Technical Editor

Tanvi Bhatt

Copy Editors

Dipti Kapadia

Gladson Monteiro

Insiya Morbiwala

Aditya Nair

Alfida Paiva

Project Coordinator

Harshal Ved

Proofreaders

Simran Bhogal

Maria Gould

Ameesha Green

Paul Hindle

Indexers

Mariammal Chettiyar

Rekha Nair

Tejal Soni

Graphics

Valentina D'silva

Production Coordinator

Adonia Jones

Cover Work

Adonia Jones

About the Authors

Sudheer Jonna was born in Andhra Pradesh, India, in 1987. Currently, he works as a senior software engineer in Chennai, India. He completed his Master's degree in Computer Applications from JNTU. In the past 3-4 years, he has worked on providing architectural designs and building various web applications based on Struts, JSF, Spring, jQuery, and JPA.

He is a JSF and PrimeFaces expert. He has been working with the PrimeFaces component library since 2011. He is a committer / project member of PrimeFaces and PrimeFaces Extensions open source projects. He has been a well-known, recognized member of the PrimeFaces community for the past few years. He is also the author of *Learning PrimeFaces Extensions Development*, *Packt Publishing*.

Besides working with the mentioned technologies, he also writes technical articles, provides online training, designs and develops web application architecture, writes books and reviews, and provides suggestions through online forums and blogs. He is interested in the research and development of various popular Java EE frameworks and many other latest technologies.

He shares his knowledge through GitHub (`https://github.com/sudheerj`); you can also follow him on Twitter (`@SudheerJonna`) or contact him on Gmail at `sudheer.jonna@gmail.com`.

I would like to thank my friend, Çağatay Çivici, the book reviewers, and the team at Packt Publishing for their support and great teamwork for the past few years.

Also, a very big thanks to my parents, brother, sister, colleagues, roommates, and friends for their support in helping me complete this book very quickly.

Ramkumar Pillai is the leading authority on latest global technical trends and a proficient technical architect of the Java J2EE technology. He has been employed by major software companies across the globe, and he currently works as a senior consultant in advanced web technology.

He can be described as a smart professional with significant IT experience in technical architecture and project management on leading Java technology stacks. He has received acclaim for his contributions to PrimeFaces, Grails and Groovy, and Play Frameworks all through his career in the form of deliverables, documentation, or presentations. He was also a co-speaker at the conference on the latest trends in the web technology stack at Dallas, Texas, in September 2012.

When he is not working, he creates web designs and illustrations for retail marketing portals and blogs about almost anything, be it Big Data or his favorite Bonsai culture.

He has also been a lead consultant for companies such as Triadic Technologies and Smarterscart. He is the kind of person who believes "good is the enemy of great", and he is currently working on a few research and development projects that are not related to his favorite subject, Java.

I would like to thank all who have supported me with the production of this book, and also Packt Publishing for providing me with this opportunity.

About the Reviewers

Ramanath Chandramohan Bhongale completed his engineering degree in Information Science from KVGCE Sullia (affiliated to VTU) in 2005 and is currently working at KPIT Technologies Ltd. as a technical lead on Java, J2EE, and various development projects. He is passionate about sharing knowledge by conducting corporate seminars and writing technical blogs. His major work experience is in creating and managing automated systems for workflow and testing, developing, and deploying flows.

I would like to express my special gratitude to my wife, Raksha, who gave me the support that I needed in carrying out this assignment. Additionally, I would like to thank my friend, Pankaj Patel, who helped me in re-reviewing difficult sections. Finally, I would like to thank the team at Packt Publishing who gave me an opportunity to contribute to the successful publication of this book.

Aristides Villarreal Bravo is a Java developer, a member of the NetBeans Dream Team, and the leader of a Java User Group. He is also the CEO of Javscaz Software Developers.

Aristides has organized and participated in various conferences and seminars related to Java, Java EE, the NetBeans platform, free software, and mobile devices, both nationally and internationally. He writes tutorials and blogs about Java, NetBeans, and web development too.

He has given several interviews on sites such as NetBeans, NetBeans Dzone, and javaHispano and developed various plugins for NetBeans. He specializes in JSE, JEE, JPA, Agile, and Continuous Integration.

You can visit him at `http://avbravo.blogspot.com`.

I would like to thank my grandfather for his lighting over time.

Vineet Jain is currently working as a project lead in a leading software company. He has a total of over 6 years of experience, during which time he has worked for a number of projects in Java and other technologies. He has a rich experience in building applications using PrimeFaces and PrimeFaces extensions.

S V Narayana was born in Andhra Pradesh, India, in 1981. Currently, he is working as a project lead in Chennai, India. He has completed his professional degree (BTech) in Electrical and Electronic Engineering from S.V. University (NBKRIST, Vidyanagar, Nellore, Andhra Pradesh). For the past eight years (since 2006), he has been architecting, designing, and developing software professionally and has been using Java as his primary programming language. He is a Java, Java EE, and PrimeFaces expert. He has been working with the healthcare and BFS domains.

You can contact him at `svnari@gmail.com`.

Enrique Enolva Tan currently works as senior Java web developer at STM Philippines, a division of Duke Manufacturing based in St. Louis, Missouri, United States, with more than 7 years of experience in Java EE and related frameworks, and 16 years of overall experience in both IT infrastructure and application development engineering.

He has developed and deployed various types of application websites, which range from retail and fast food to online gaming casinos. He has also held several managerial positions in information technology in the online gaming casino industry.

He aspires to be a successful technopreneur—he is working on several concepts for patents in Mobile Name System (MNS), Mobile Certificate Authority (MCA), and Subscriber Identity Module Search and Trust Platform (SIM-STP).

He devotes his free time to develop his own website (`www.hanapsim.com`) for SIM Search and Trust Platform using the Java, JSF, and PrimeFaces technologies.

I would like to thank Rechil Lentejas Artizo and my kids, Aerozekiel, Aerikezedek, and Aerika Faith for bearing with me even though I sacrificed some of our family time to review this book.

---

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface

PrimeFaces is a leading lightweight open source user interface component library for JSF-based web applications. It provides a rich set of 100+ UI components with a single JAR, zero configuration, and no prerequisites. PrimeFaces aims to create built-in Ajax components that are based on standard JSF 2.0 Ajax APIs with a rich look and feel, with the help of a theming mechanism.

The initial development of PrimeFaces was started in 2008 by a Turkish JSF expert, Çağatay Çivici. Prior to developing PrimeFaces, he had been working on the YUI4JSF library. This experience of working with the YUI4JSF library allowed him to start developing PrimeFaces, which was initially based on the YUI JavaScript library. A few days later, the PrimeFaces team decided to replace this library with the powerful jQuery framework in order to create the component widgets.

This book will guide you through the process of creating a wide range of rich UI web applications based on successful, real-world business models. Each chapter comes with a custom web project, which you can build with a detailed, step-by-step procedure. This is accompanied by explanations of the key features used. By the end of each chapter, you will learn how to build specific, customized web applications using the PrimeFaces components. The projects inside this book make use of the latest versions of PrimeFaces (Version 5.0) and JSF (Version 2.2).

What this book covers

Chapter 1, *Creating a "Hello World" Application*, discusses how to create a simple "Hello World" PrimeFaces application in a step-by-step procedure and how to create the recommended PrimeFaces - supported environment (required software, browsers, IDE, tools, and so forth) that is required for real-world application development. A brief introduction to PrimeFaces and its role in UI applications will also be covered at the beginning of the chapter.

Chapter 2, *Creating an Employee Registration Application*, shows you how to create an employee registration application, which is an example of how to form components and their validations. The big set of PrimeFaces form components will be divided into two main categories, which are input components and select components, to create the registration type of the form filling application. These components and their validations that are used will also be explained in detail.

Chapter 3, *Creating a Simple Restaurant Point of Sale Application*, explains how to create a restaurant POS application with the help of the PrimeFaces layout and grouping components. These topics provide the templating mechanism needed for a fancy application, and each concept will be explained in detail.

Chapter 4, *Global Mutual Funds Tracking*, directs you on how to create a global mutual funds tracking application with the help of a data container and dialog components. This project emphasizes the usage of the data container and dialog components in order to maintain the big datasets in investment/financial schemes. The master-client and hierarchical data relationships (which are linked to these components) will also be well explained.

Chapter 5, *Investor Information Analysis and Reporting*, shows you how to create an investor information analysis application, which is useful because it is an example of data visualization and reporting components. Big datasets for reporting data will be analyzed in the form of charts and various formats of export features. The data visualization and reporting components used within this application will also be explained.

Chapter 6, *Creating a Simple Online Shopping Cart Application*, shows you how to create an online shopping cart application using the major menu variations and drag-and-drop components in PrimeFaces. This shopping cart application can be used to cover various products, such as electrical goods or household products.

Chapter 7, *Creating an Online Video Portal Application*, directs you on how to create an online movie portal application, which makes use of PrimeFaces multimedia, maps, and schedule components. This application can be used in the entertainment world and to book events that vary seasonally, such as special events in the summer or winter. Customers can pick the events that interest them and book them accordingly.

Chapter 8, *Creating an Online Printing Station Application*, guides you through how to build an online printing station application. This application handles all types of files using the PrimeFaces file upload and download components. This application will showcase a new concept that provides a platform to track, submit, and process printing jobs from multiple users across different locations.

Chapter 9, *Creating an Online Chat Application*, shows you how to create an online chat room application using the PrimePUSH. The PrimePUSH API deals with the asynchronous communication between the server and client using this chat room application. This chapter uses the simple social network application as our sample project.

Chapter 10, *Creating a Healthcare Products Application*, shows you how to create an online healthcare products application using common utility components and theming concepts (such as using built-in themes, customizing these, and creating new ones).

What you need for this book

As a reader of this book, you will need Java 5 or above and Maven installed on your machine, along with the JSF and PrimeFaces libraries. Optionally, you can use software and tools such as the Eclipse IDE, MySQL DB, and any browser tools to debug the application. You should also have a basic knowledge of JSF, PrimeFaces, and jQuery.

In order to run the customized web projects in this book, you need to store them on GitHub. This means that you can pull the source code at any time in order to have an efficient practical experience.

The software libraries or tools used in all of the customized projects of this book are listed as follows:

- JDK 1.5+ from Oracle's official site. You can download this at `http://www.oracle.com/technetwork/java/javase/downloads/index.html`.
- The Mojarra Java Server Faces implementation's latest version. This can be downloaded at `https://javaserverfaces.java.net/download.html`.
- The latest version of PrimeFaces from the PrimeFaces official site, which can be downloaded at `http://www.PrimeFaces.org/downloads.html`.
- The latest version of PrimeFaces Extensions, which can be downloaded at `http://PrimeFaces-extensions.github.io/`.
- MySQL from its official site (`http://www.mysql.com/`), or any other relational database based on the JDBC driver.
- Eclipse from the official site (`http://www.eclipse.org`), or any other Java IDE.
- The Apache Maven build tool from the official site (`http://maven.apache.org/`) to work with customized projects.

- PrimeFaces components result in a rich UI application with lots of CSS, JS, and HTML markup resources; it would be very helpful to use browser tools such as the Firebug plugin for Firefox, FirebugLite for the Chrome browser, Developer Tools (*F12*) for Internet Explorer, and Developers Tools (which you can access using *Ctrl + Shift + I)* for the Chrome browser.
- To work with the Blueprint projects, you need to check out the source code using a Git client or using IDE Git repositories. The step-by-step instructions to run the customized web projects are covered in GitHub (`https://github.com/sudheerj/PrimeFaces-blueprints`).

Who this book is for

This book is for anyone who wants to learn how to create customized PrimeFaces web applications. If you want to create a different set of categories for customized applications using PrimeFaces components, then this book is for you.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "These managed beans interact between XHTML/Facelets and the `POJO` class with the `@ManagedBean` annotation."

A block of code is set as follows:

```
<div id="header">
<ui:insert name="header">
  <ui:includesrc="/templates/common/header.xhtml" />
</ui:insert>
</div>
```

Any command-line input or output is written as follows:

```
mvn clean package
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Select the PrimeFaces component suite from the **Components** tab."

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title through the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on `www.packtpub.com` or e-mail `suggest@packtpub.com`.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website, or added to any list of existing errata, under the Errata section of that title.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Creating a "Hello World" Application

This chapter will show you how to create your own "Hello World" application in order to give you a head start with the application development for PrimeFaces. This chapter will provide you with a brief introduction to its features and its role in web applications, with a step-by-step setup and configuration. Most importantly, it will give you an insight into how Ajaxified components change the old development trend. We will also learn how to make application development easier with code completion, IDE support, and the code generator tool used for creating the **CRUD** web application. The specific topics that will be covered in this chapter are as follows:

- An introduction to **PrimeFaces**, its features, and its role in customized application development
- PrimeFaces setup and configuration for development
- How to quickly develop a project: a "Hello World" application, in this case
- How to change the old trend of development using Ajaxified components
- How to use code completion, NetBeans bundles PrimeFaces, and the code generator tool

An introduction to JavaServer Faces and PrimeFaces

JavaServer Faces (**JSF**) is a component-based MVC framework used for building rich **User Interface** (**UI**) Java web applications. JSF is a powerful framework with a six-phase lifecycle, and it will automate the common web application tasks such as decoding the user input, processing the input validations and conversions, and rendering or updating the output in the form of generated HTML. Page authors can easily build a customized UI by just dragging-and-dropping the reusable components on the page that provide a rich look and feel to modern UI applications. JSF has built-in support for input conversions and validations, and Ajax support for the components.

Going by the growing popularity of JSF technology, many open source and proprietary UI component frameworks were created to have user interfaces with a fancier look and feel. These component suites were created by introducing their own new components and extending the standard JSF components with additional features. Among all these component suites, PrimeFaces is the best and most popular component suite considering its features, quick releases with more new components and bug fixes, ease of development, extensive documentation, and support from its community.

PrimeFaces is a leading, lightweight, open source user interface component library for JSF-based web applications. In the JSF world, it is miles ahead of the other existing component sets because of the many features it has at its disposal:

- Over 100 sets of components
- Built-in Ajax-supported components
- Ease of development, as there are no configurations required
- A single jar install without the need for any mandatory third-party libraries
- More than 30 predefined themes and custom themes by using the ThemeRoller support
- Multibrowser support

It is so well designed that it is important to consider its importance when developing web applications. Page authors and application developers can easily develop web pages by simply dragging-and-dropping the components of the webpage and then adding the required features in a step-by-step fashion: customizing the CSS style classes, extending the component widgets, and rendering according to the custom requirements.

Setting up and configuring PrimeFaces

PrimeFaces is a lightweight single library with minimal external libraries. The only external libraries required are those with component-specific features. Apart from these component-specific features, projects only require JSF runtime implementations such as **Oracle Mojarra** or **Apache MyFaces**.

The setup and configuration for Maven and non-Maven users is explained in the following two sections.

Setting up and configuring using Maven

In this section, we will define the various Maven configuration steps required to run a PrimeFaces-based application. Perform the following steps:

1. Configure the PrimeFaces library dependency or Maven coordinates in your project `pom.xml` file as shown here:

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

2. Add the PrimeFaces repository to the repositories list of your project `pom.xml` file as follows:

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

Note that this step is not required for releases after PrimeFaces 4.0. The team started adding its library in the Maven central repository.

3. Configure either of the JSF runtime implementations, Oracle Mojarra or Apache MyFaces. Choose either of the following two blocks of code:

    - This is the runtime implementation for Oracle Mojarra:

```
<dependency>
    <groupId>com.sun.faces</groupId>
    <artifactId>jsf-impl</artifactId>
    <version>2.2.6</version>6
</dependency>
```

- This is the runtime implementation for Apache MyFaces:

```
<dependency>
    <groupId>org.apache.myfaces.core</groupId>
    <artifactId>myfaces-impl</artifactId>
    <version>2.2</version>
</dependency>
```

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

Depending on the component-specific features, you can use the following mandatory and optional dependencies. Here is a list of dependencies categorized into mandatory and optional. The following are the mandatory dependencies:

Dependencies	Version	Description
JSF runtime	2.0, 2.1, and 2.2	Oracle's Mojarra or Apache MyFaces implementation
PrimeFaces	5.0	The PrimeFaces UI component library

The following are the optional dependencies:

Dependencies	Version	Description
iText	2.7	To use the DataExporter component for PDF format
POI	3.7	To use the DataExporter component for Excel format
Rome	1.0	To use the Feed reader component
commons-fileupload	1.3	To use the fileupload component (when web server / application server doesn't support servlet 3.0)
commons-io	2.2	To use the fileupload component

Setting up and configuring for non-Maven (or Ant) users

In this section, we will define the various non-Maven (or Ant) configurations required to run a PrimeFaces-based application. Perform the following steps:

1. Download the PrimeFaces library from the official download section of PrimeFaces at `http://www.primefaces.org/downloads.html`.
2. Following this, add the PrimeFaces JAR library to the classpath.
3. You should then download either the JSF library runtimes from Oracle's Mojarra or those from Apache MyFaces from their official sites and add them to the classpath. You can access the JSF library at Oracle by going to `https://javaserverfaces.java.net/2.2/download.html` or alternatively access it at Apache by going to `http://myfaces.apache.org/download.html`.
4. After this, you should download the component-specific third-party libraries from their official site and add them to the classpath.

Application-level configuration

As you know, PrimeFaces is a JSF-based component suite. Therefore, the first thing you have to do is configure the JSF Faces Servlet in your project deployment descriptor file (`web.xml`). The following is a mandatory configuration for any JSF-based application:

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet
  </servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.jsf</url-pattern>
</servlet-mapping>
```

```
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.faces</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>*.xhtml</url-pattern>
</servlet-mapping>
```

It is not mandatory to use all of the JSF extensions or servlet mappings. Any of the preceding servlet mappings is enough to configure Faces Servlet to your project.

There are other configurations that can be made to your project. These are shown in the following table:

Context parameter name	Default value	Description
`THEME`	Aristo	Used to apply a specific theme to your application. All theme names are valid values.
`SUBMIT`	Full	Enables the Ajax submit mode. The valid values are `full` and `partial`.
`DIR`	Ltr	Defines the component content orientation. The valid values are `ltr` and `rtl`.
`RESET_VALUES`	False	When this is enabled, any Ajax-updated inputs are reset first. The valid values are `true` and `false`.
`SECRET`	PrimeFaces	Defines the secret key to encrypt-decrypt the value of the expressions that are exposed in rendering `StreamedContents`.
`CLIENT_SIDE_VALIDATION`	False	Controls client-side validations to the form components.
`UPLOADER`	Auto	Defines the `fileuploader` mode. The valid values are `auto`, `native`, and `commons`.

As an example, the following code snippet configures a theme with `context-param`:

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

Checking the JSF runtime compatibility

PrimeFaces 5.0 supports all the JSF runtime versions: 2.0, 2.1, and 2.2 at the same time using feature detection without having to compile a dependency to any specific version. In other words, some of the features that are available are based on the runtime version used. The newly released JSF 2.2 version supports more popular HTML5.

The runtime detection policy for PrimeFaces is quite useful for the newly added features in JSF library. The **JSF 2.2** `passthrough` attribute's feature is a good example of the runtime detection policy. That is, the passthrough attribute only gets rendered if the runtime is JSF 2.2.

An introduction to the `autofocus` and `pattern` HTML5 attributes' integration with PrimeFaces can be seen in the following example:

```
<!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>
        <h:form>
           <p:inputText value="#{bean.value}" pt:autofocus="autofocus"
             pt:pattern= "[A-Za-z]"/>
        </h:form>
      </h:body>
</html>
```

Developing your first PrimeFaces application

In the previous section, you have learned how to set up and configure PrimeFaces for JSF-based web applications. To start using its components in your web project, you have to add the following namespace at the top of the namespace section:

```
xmlns:p="http://primefaces.org/ui"
```

Once you have successfully completed the setup and configuration for PrimeFaces, you will be shown how to develop a simple "Hello World" application by simply following these steps:

1. The first step is to create a simple `helloworld.xhml` page. This will display the "Hello World" message from the PrimeFaces Panel component. You will be able to display this message using the following section of code:

```
<!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">
<f:view contentType="text/html" >
  <h:head>
    <title>Primefaces Hello World page</title>
  </h:head>
  <h:body>
     <h:form>
         <p:panel  header="Hello" footer="Blueprints world"
           style="width:300px;margin-left:40%;margin-top:15%">
           Welcome to Primefaces
         </p:panel>
     </h:form>
  </h:body>
</f:view>
</html>
```

2. Following this, package the application somewhere in the target directory (by default) and then run the application with the help of the following Maven commands:

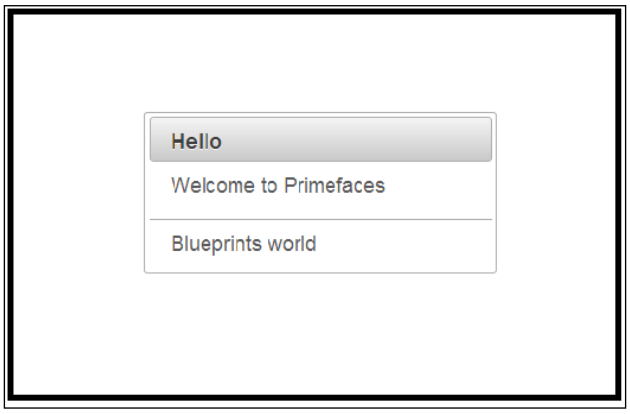
```
mvn clean package
```

```
mvn jetty:run
```

or:

```
mvn tomcat:run
```

3. After this, go to the browser address bar and access the "Hello World" application by navigating to `http://localhost:8080/chapter01/views/helloworld.jsf`.
4. You should now be able to see the **Hello World** message with the help of the panel component in the web page. This can be seen in the following screenshot:



Change the old trend of development with Ajaxified components

One of the design goals of the PrimeFaces component suite is to simplify the web development by using Ajaxified components. The Partial Processing and Partial Page Rendering features play a major role in creating a powerful AJAX framework.

Learning Partial Processing

PrimeFaces provides the **Partial Processing** feature in order to execute the JSF lifecycle phases required for the specified components with the help of a process attribute. Here, you can only process the required components instead of the entire web page, which are called *lightweight* requests. This occurs quite commonly, such as when creating the PrimeFaces web pages. This is done with the help of registration form fields, which include a certain group of validation results that validate the different components depending on the executed action. However, you can avoid unnecessary validations for other components and prevent validation errors on the submission of the form. The process attribute is not only used to process the specific components but also to process the specific regions with the help of the `@this`, `@form`, `@parent`, `@none`, and `@all` expressions. You can also combine the components with these simple expressions at the same time using a comma separator (,).

The most common scenario is the dependent drop-down input values along with the other required components of the same page. This feature can be explained with an example.

In the case of submitting the order details, you can see one or more dependent relationships. In this example, a list of products depends on the selected category and the list of orders depends on the selected product. Firstly, you have to select the category and then based on this category, select the product. After choosing the product, you then select one order from the list of populated orders. You should also assume that this registration form contains other required form fields such as a `calendar` and the `inputText` components. The preceding use case scenario is represented in the following code snippet:

```
<h:outputText value="Category: " />
<p:selectOneMenu id="categories" value="#{ppController.category}">
<f:selectItems value="#{ppController.categories}" />
<p:ajax listener="#{ppController.updateProducts}" event="change"
  update="products" process="@this"/>
</p:selectOneMenu>

<h:outputText value="Product: " />
<p:selectOneMenu id="products" value="#{ppController.product}">
<f:selectItems value="#{ppController.products}" />
<p:ajax listener="#{ppController.updateOrders}" event="change"
  update="orders" process="@this"/>
</p:selectOneMenu>

<h:outputText value="Order: " />
<p:selectOneMenu id="orders" value="#{ppController.order}">
<f:selectItems value="#{ppController.orders}" />
</p:selectOneMenu>

<h:outputText value="Number of Orders: " />
<p:inputText value="#{ppController.ordersCount}" id="ordercount" />
  required="true" />

<h:outputText value="Date of Order: " />
<p:calendar value="#{ppController.selectdate}" id="selectdate" />
  required="true" />
```

You can clearly see that without using the Partial Processing feature, validation errors might occur on the number of orders of the input field and the date of the order calendar components.