PrimeFaces: Date Input

(And also PrimeFaces Versions of Standard JSF Elements)

Originals of slides and source code for examples: http://www.coreservlets.com/JSF-Tutorial/primefaces/
Also see the JSF 2 tutorial – http://www.coreservlets.com/JSF-Tutorial/jsf2/
and customized JSF2 and PrimeFaces training courses – http://courses.coreservlets.com/jsf-training.html

Customized Java EE Training: http://courses.coreservlets.com/
Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

more SERVLETS and JAVASERVER PAGES
MARTY HALL
2ND EDITION
core SERVLETS and JAVASERVER PAGES
Volume 1: Core Technologies
MARTY HALL · LARRY BROWN

For live training on JSF 2, PrimeFaces, or other Java EE topics, email hall@coreservlets.com
Marty is also available for consulting and development support

Taught by the author of *Core Servlets and JSP*, this tutorial, and JSF 2.2 version of *Core JSF*. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
  - JSF 2, PrimeFaces, Ajax, jQuery, Spring MVC, JSP, Android, general Java, Java 8 lambdas/streams, GWT, custom topic mix
  - Courses available in any location worldwide. Maryland/DC area companies can also choose afternoon/evening courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
  - Hadoop, Spring, Hibernate/JPA, RESTful Web Services

Contact hall@coreservlets.com for details

Topics in This Section

- **Basic Date input with p:calendar**
- **Inline calendars**
- **PrimeFaces versions of standard elements**
  - p:commandButton, p:message, p:messages, p:fieldset, p:ajax, standard HTML
- **Ajax updates**
- **Controlling look of the calendar**
- **Animation effects on open/close**
- **Collecting times as well as dates**

Big Ideas in Using PrimeFaces

- **Normal JSF approaches**
  - You use the normal JSF way of writing managed beans, doing validation, looping, applying the expression language, and *everything* else in the previous sections of the coreservlets.com JSF tutorial. You just have more choices for GUI elements.
- **Consistent and extensive Ajax support**
  - PrimeFaces builds in GUI-element-specific Ajax events, and has *many* Ajax enabled components.
- **To plan for themes, switch h: to p:**
  - Even when you are not making use of any exttra PrimeFaces functionality, you should change h:commandButton to p:commandButton, h:inputText to p:inputText, and so forth.
  - Relatively early on, you should also learn some of the most basic PrimeFaces CSS names, so your regular HTML will also pick up the PrimeFaces theme.

p:calendar – Overview

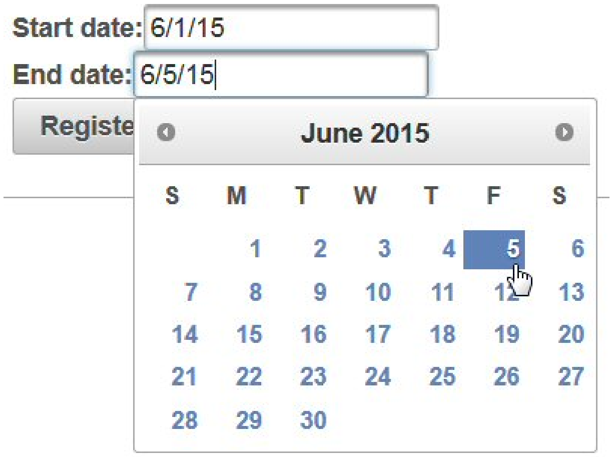
Customized Java EE Training: http://courses.coreservlets.com/

Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop.

Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

p:calendar: Overview

- **Appearance and behavior**
  - Textfield with associated popup calendar.
- **Purpose: for collecting dates from user**
  - Value is automatically converted to java.util.Date. UI prevents user from entering an illegal value.
- **Ajax behavior**
  - PrimeFaces defines “dateSelect” event. Can respond by using p:ajax.



Start date: 6/1/15

End date: 6/5/15

Minor nit: for parallelism with standard JSF elements, I would have preferred that the component be called p:inputDate instead of p:calendar.

Summary of Most Important Attributes

- **<p:calendar value="…" …/>**
  - value
    - Should point to bean property of type Date.
  - mode (popup [default] or inline)
    - Should calendar pop up on click or always be shown?
  - showButtonPanel (true or false [default])
    - Should there be “Current Date” and “Close” buttons at bottom?
  - showOn (focus [default], button, both)
    - When to display the calendar.
  - navigator (true or false [default])
    - Should it show menu that lets users navigate to month or year?
  - pages (*integer* [default is 1])
    - Number of months to show at once.
  - effect (slide, explode, fold, fadeIn, etc. [default: none])
    - Animation effect for showing and hiding calendar

Summary of Usage

- **Input form**

```
<h:form>
Choose arrival date:
<p:calendar value="#{dateBean.startDate}" …/> …
</h:form>
```

- **Java code**

```
@ManagedBean
public class DateBean {
  private Date startDate;

  public Date getStartDate() { return(startDate); }

  public void setStartDate(Date startDate) {
    this.startDate = startDate;
  }
}
```

- **Output page**

```
Your arrival date: #{dateBean.startDate}
```

Example: Basics

Example Overview: Basics (JSF Resort Checkin)

- **Input page collects**
  - Start and end dates for resort checkin
    - Calendar pops up when user clicks in textfield
  - Validation: date required, checkout date after checkin
    - Example uses required and requiredMessage for validation. Also custom validation is performed in action controller, with message set unless the end date is later than the start date.
      - See lecture on validation in the general JSF2 tutorial at http://www.coreservlets.com/JSF-Tutorial/jsf2/
    - Replaced h:messages with p:messages to get look and feel of current PrimeFaces theme
      - But would have behaved properly with h:messages
- **Results page shows**
  - Confirmation of dates

Bean (Bean Properties)

```
@ManagedBean
public class DateBean {
  private Date startDate, endDate;

  public Date getStartDate() {
    return(startDate);
  }

  public void setStartDate(Date startDate) {
    this.startDate = startDate;
  }

  public Date getEndDate() {
    return(endDate);
  }

  public void setEndDate(Date endDate) {
    this.endDate = endDate;
  }
```

Bean (Formatted Date for Results Page)

```
/** Returns a String representing the starting date, in a form similar to
 * "Wednesday, June 10, 2015". For use in results page.
 */
 public String getStartDay() {
   return(DateUtils.formatDay(startDate));
 }

/** Returns a String representing the ending date, in a form similar to
 * "Thursday, June 11, 2015". For use in results page.
 */
 public String getEndDay() {
   return(DateUtils.formatDay(endDate));
 }
```

Bean (Action Controller)

```
public String register() {
  FacesContext context = FacesContext.getCurrentInstance();
  if (!startDate.before(endDate)) {
    endDate = null;
    FacesMessage errorMessage =
      new FacesMessage("End date must be after start date");
    errorMessage.setSeverity(FacesMessage.SEVERITY_ERROR);
    context.addMessage(null, errorMessage);
    return(null);
  } else {
    return("show-dates");
  }
}
```

If I use <h:messages/> or <h:message for="some-id"/>, then there is no particular reason to set the severity; it displays the same thing regardless of the severity. But, if I use <p:messages/> or <p:message for="some-id"/>, it is useful to set the severity so that the background color is red (in default theme) and so the icon corresponds to errors, not informational notices. The errors generated by the builtin validation (required/requiredMessage, validationMessage, f:validateLength, etc.) have this severity, so it is useful for the custom messages to match.

Helper Class (Date Formatting)

```
public class DateUtils {

  /** Given a Date, returns a String of the form "Day, Month Number, Year",
   *  e.g., "Wednesday, June 10, 2015". Used by getStartDay and getEndDay.
   */
  public static String formatDay(Date date) {
    if (date == null) {
      return("");
    } else {
      return(String.format("%tA, %tB %te, %tY",
                           date, date, date, date));
    }
  }
  ...
  private DateUtils() {} // Uninstantiatable class
}
```

You could also omit this method and use h:outputText with f:convertDateTime. But, using this method is simpler.

Input Page

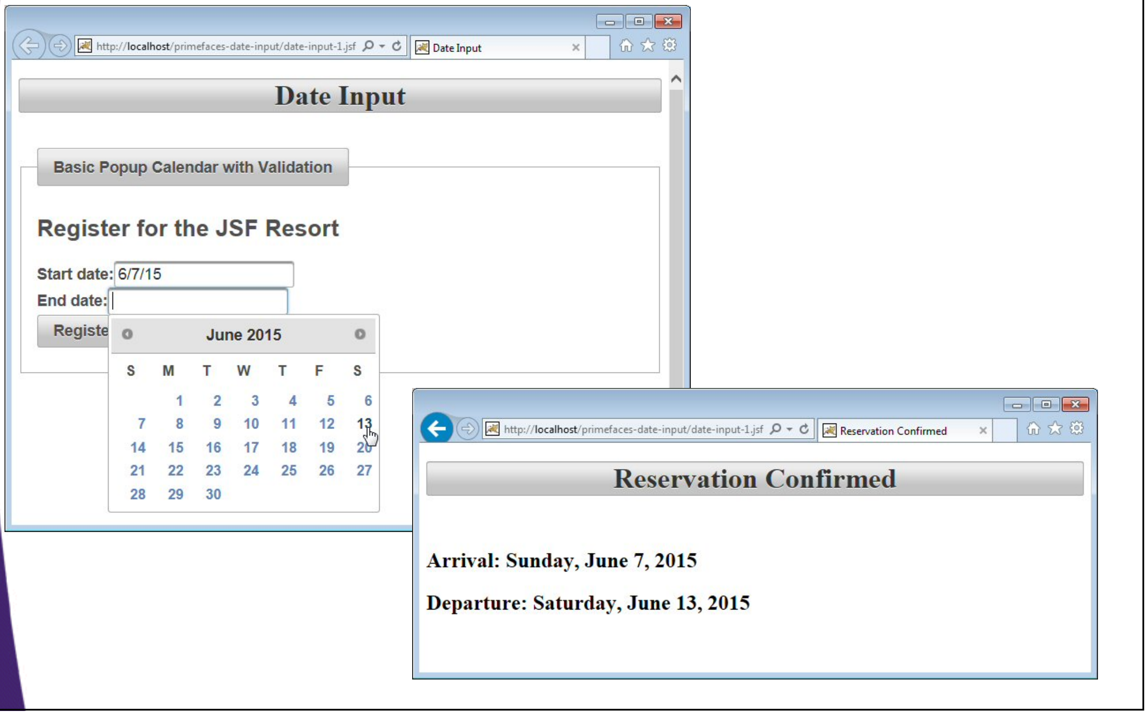
```
<!DOCTYPE html ...>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://xmlns.jcp.org/jsf/html"
      xmlns:p="http://primefaces.org/ui">
...
<h:form>
   <h2>Register for the JSF Resort</h2>
   <p:messages/>
   <b>Start date:</b>
   <p:calendar value="#{dateBean.startDate}"
               required="true"
               requiredMessage="Start date required"/><br/>
   <b>End date:</b>
   <p:calendar value="#{dateBean.endDate}"
               required="true"
               requiredMessage="End date required"/><br/>
   <p:commandButton action="#{dateBean.register}"
                    value="Register" ajax="false"/><br/>
</h:form>
...
```

It works fine to use h:commandButton, but the advantage of p:commandButton is that the button picks up the fonts, sizes, and colors of the current theme. But, if you use p:commandButton, you have to use ajax="false" if you want it to behave just like h:commandButton.

Results Page

```
...
<h2>Arrival: #{dateBean.startDay}</h2>
<h2>Departure: #{dateBean.endDay}</h2>
...
```

Results (Good Dates)



http://localhost/primefaces-date-input/date-input-1.jsf

Date Input

Basic Popup Calendar with Validation

Register for the JSF Resort

Start date: 6/7/15

End date:

Register

June 2015

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

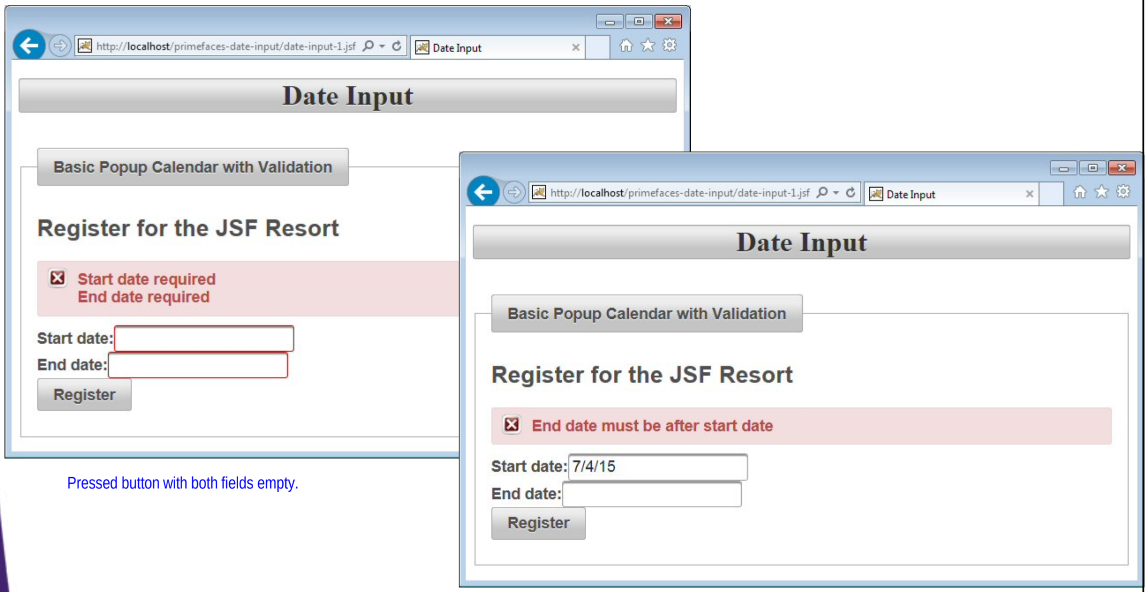
http://localhost/primefaces-date-input/date-input-1.jsf

Reservation Confirmed

Arrival: Sunday, June 7, 2015

Departure: Saturday, June 13, 2015

Results (Bad Dates)



http://localhost/primefaces-date-input/date-input-1.jsf

Date Input

Basic Popup Calendar with Validation

Register for the JSF Resort

☒ Start date required
End date required

Start date:

End date:

Register

Pressed button with both fields empty.

http://localhost/primefaces-date-input/date-input-1.jsf

Date Input

Basic Popup Calendar with Validation

Register for the JSF Resort

☒ End date must be after start date

Start date: 7/4/15

End date:

Register

Pressed button with same date in both fields.

Inline Calendars

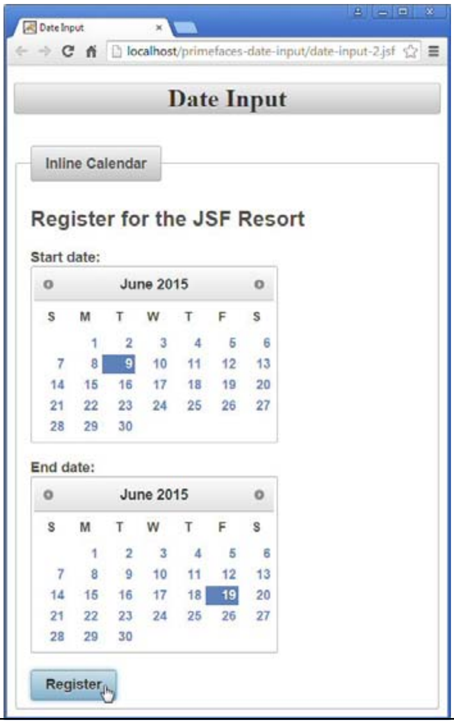
Example Overview: Inline Calendar

- **Input page collects**
  - Start and end dates for resort checkin
    - Calendar is displayed inline, and no textfield is shown
    - <p:calendar value="…" mode="inline" …/>
      - Default is popup, so omitting mode is same as mode="popup"
- **Results page shows**
  - Confirmation of dates
- **Bean and results page**
  - Same as in previous example, so code not repeated here

Input Page

```
...
<h:form>
   <h2>Register for the JSF Resort</h2>
   <p:messages/>
   <b>Start date:</b>
   <p:calendar value="#{dateBean.startDate}"
               mode="inline"
               required="true"
               requiredMessage="Start date required"/><br/>
   <b>End date:</b>
   <p:calendar value="#{dateBean.endDate}"
               mode="inline"
               required="true"
               requiredMessage="End date required"/><br/>
   <p:commandButton action="#{dateBean.register}"
                    value="Register" ajax="false"/><br/>
</h:form>
...
```

Results



Date Input

Inline Calendar

Register for the JSF Resort

Start date:

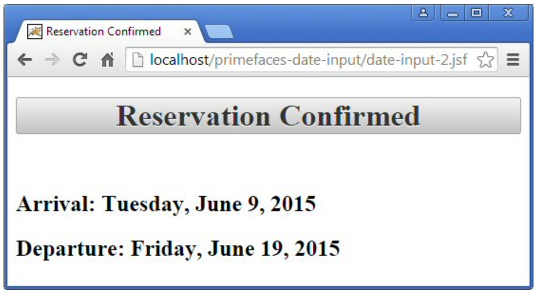
June 2015

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

End date:

June 2015

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				



Reservation Confirmed

Arrival: Tuesday, June 9, 2015

Departure: Friday, June 19, 2015

Aside: PrimeFaces Versions of Standard JSF Elements

PrimeFaces Versions of Standard Elements

- **General idea**
  - Every h:blah input element has a corresponding p:blah element. For example, h:inputText and h:commandButton can be replaced by p:inputText and p:commandButton.
- **Advantages**
  - Theme-aware.
    - The elements automatically pick up the fonts, colors, and sizes of the current theme. If you change the theme, the p:blah elements change automatically.
  - Extra features
    - Many of the p:blah elements add extra functionality in addition to being theme-aware
- **Disadvantages**
  - Requires more changes to existing apps. Harder to undo if you leave PrimeFaces.

Summary of Simple Elements Used in this Section

- **p:commandButton**
  - Change <h:commandButton action="…" value="…"/> to <p:commandButton action="…" value="…" ajax="false"/>
    - Additional features: “icon” attribute, Ajax functionality. For icon, refer to a CSS name that has a background-image property.
- **p:messages**
  - Change <h:messages styleClass="…"/> to <p:messages/>
    - Additional feature: automatic Ajax updates. By setting autoUpdate="true", messages update for every Ajax request. More details in section on PrimeFaces overlays and popup windows.
- **p:fieldset**
  - Change <fieldset><legend>Title<legend>…</fieldset> to <p:fieldset legend="Title">…</p:fieldset>
    - Additional feature: toggling. By setting toggleable="true", the end user can interactively hide/collapse and redisplay the content.
    - Note that this replaces a standard HTML element, not a JSF one.

Other Theme-Aware Replacements for Standard Elements

- **p:inputText and p:inputTextArea**
  - For h:inputText and h:inputTextArea.
  - You can also use p:watermark to set placeholder text
- **p:password**
  - For h:inputSecret (notice the inconsistent name).
  - Also supports feedback regarding password strength. For examples, see the second of the sections on string input
- **p:selectOneMenu, p:selectManyMenu, p:selectOneRadio, etc.**
  - For h:selectOneMenu, h:selectManyMenu, h:selectOneRadio, etc.
  - Can also display custom content and animation effects

p:ajax

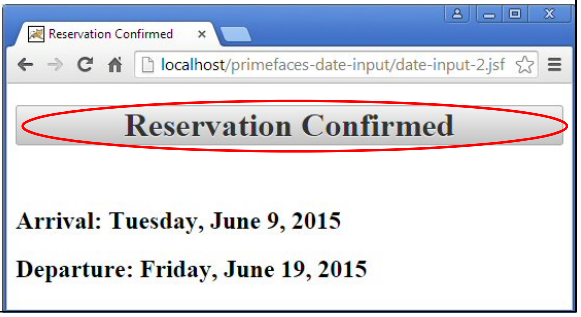
- **Designed for PrimeFaces elements**
  - Use p:ajax instead of f:ajax when used in the body of PrimeFaces elements.
    - Uses “update” instead of “render”
    - Uses “process” instead of “execute”
- **More events**
  - Many of the PrimeFaces elements define new events to respond to. For example, p:calendar defines a “dateSelect” event.
    - However, p:ajax does not usually define reasonable defaults like f:ajax does. For example, with p:calendar, you <u>must</u> explicitly say <p:ajax event="dateSelect" …/>
- **Simpler listeners**
  - When you use <p:ajax listener="#{someBean.someMethod}" …/>, the method can take zero arguments, unlike with f:ajax.
- **Documentation note**
  - In the “Component Suite” section of the official PrimeFaces User’s Guide, p:ajax is listed under AjaxBehavior, not under Ajax

Using PrimeFaces Themes in Standard HTML

- **Idea**
  - PrimeFaces documents the CSS class names that it uses (most come from jQuery UI). If you use those names in your normal HTML, then your HTML will also change when the PrimeFaces theme changes.
- **Example**

```
<h1 class="ui-widget-header ui-corner-all" align="center">
Reservation Confirmed</h1>
```

- **More details**
  - See lecture on Themes



Reservation Confirmed

Arrival: Tuesday, June 9, 2015

Departure: Friday, June 19, 2015

Ajax Updates

Example Overview: Instant Ajax Updates

- **Input page**
  - Shows popup calendar and message showing current selection. Message updates whenever user clicks on a date

Note that attribute is “update”, rather than “render” as with f:ajax.

```
<p:calendar value="#{dateBean2.sampleDate}" mode="inline">
   <p:ajax event="dateSelect" update="selection"/>
</p:calendar>
<h:outputText value="#{dateBean2.sampleDay}" id="selection"/>
```

- **Bean**
  - Has getter/setter for Date and getter for formatted Date
    - No special methods to support Ajax
- **No separate results page**
  - Date shown on main page

Reminder: Ajax is covered in detail in the general JSF2 tutorial at http://www.coreservlets.com/JSF-Tutorial/jsf2/.
This is just a small variation with events specific to p:calendar.

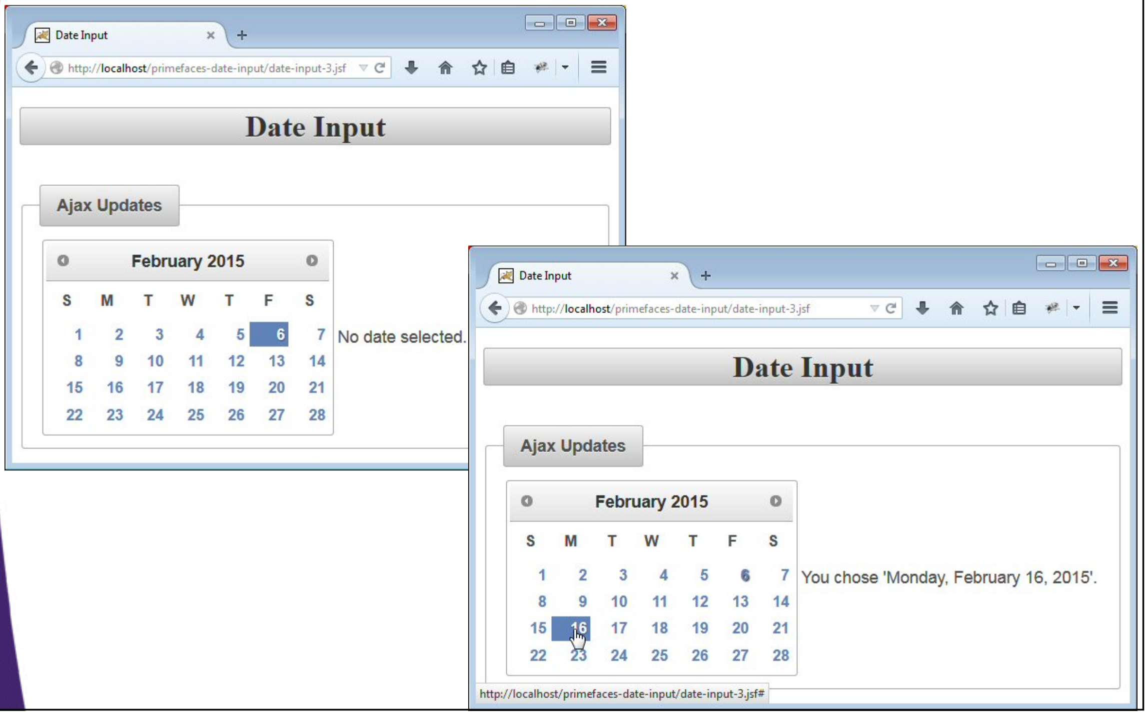
Bean

```
@ManagedBean
public class DateBean2 {
  private Date sampleDate;

  public Date getSampleDate() {
    return (sampleDate);
  }
  public void setSampleDate(Date sampleDate) {
    this.sampleDate = sampleDate;
  }

  public String getSampleDay() {
    if (sampleDate == null) {
      return("No date selected.");
    } else {
      String message =
        String.format("You chose '%s'.",
                      DateUtils.formatDay(sampleDate));
      return(message);
    }
  }
}
```

Input Page

```
...
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}" mode="inline">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleDay}"
                 id="selection"/>
</h:panelGrid>
</h:form>
...
```

Results



Date Input

http://localhost/primefaces-date-input/date-input-3.jsf

Date Input

Ajax Updates

February 2015

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

No date selected.

Date Input

http://localhost/primefaces-date-input/date-input-3.jsf

Date Input

Ajax Updates

February 2015

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

You chose 'Monday, February 16, 2015'.

http://localhost/primefaces-date-input/date-input-3.jsf#

Controlling Look of Calendar

Attributes that Control Calendar Appearance

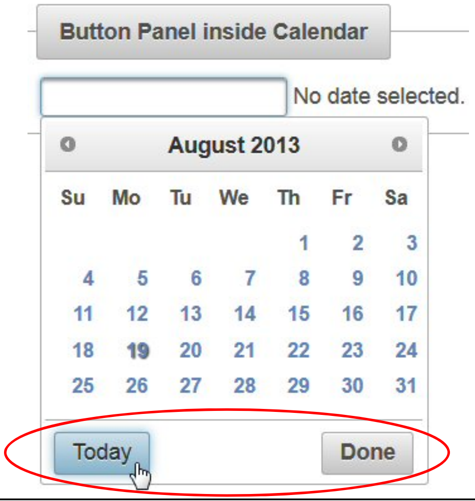
- **showButtonPanel (true or false [default])**
  - Should there be “Today” and “Done” buttons at bottom?
- **showOn (focus [default], button, both)**
  - When to display the calendar.
- **navigator (true or false [default])**
  - Should it show menu that lets users navigate to month or year?
- **showOtherMonths (true or false [default])**
  - For first and last week, should dates in previous or next month also be shown, rather than empty spaces?
- **pages (*integer* [default is 1])**
  - Number of months to show at once.

Button Panel

- **Code**

```
<p:calendar value="#{dateBean2.sampleDate}"
            showButtonPanel="true"/>
```

- **Result**



Button Panel inside Calendar

No date selected.

August 2013

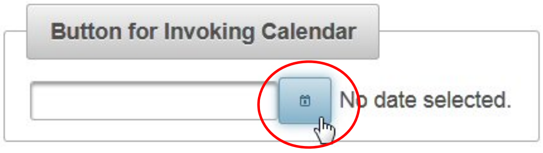
Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Button for Invoking Calendar

- **Code**

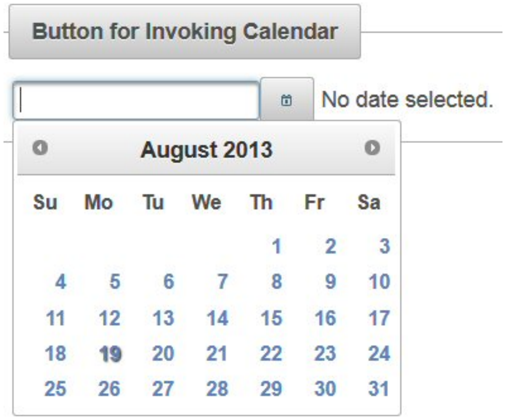
```
<p:calendar value="#{dateBean2.sampleDate}"
            showOn="both"/>
```

- **Result**



Button for Invoking Calendar

No date selected.



Button for Invoking Calendar

No date selected.

August 2013

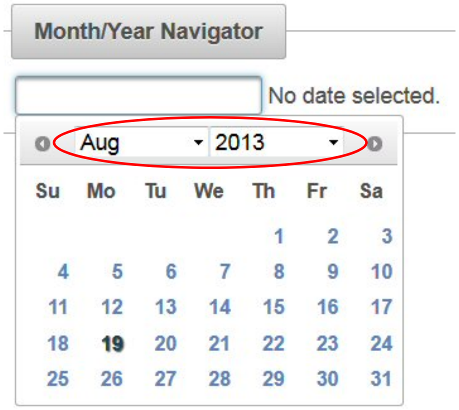
Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Month/Year Navigator Menu

- **Code**

```
<p:calendar value="#{dateBean2.sampleDate}"
            navigator="true"/>
```

- **Result**



Month/Year Navigator

No date selected.

Aug 2013

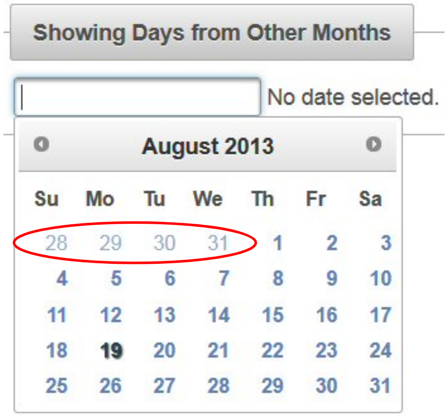
Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Showing Days from Other Months

- **Code**

```
<p:calendar value="#{dateBean2.sampleDate}"
            showOtherMonths="true"/>
```

- **Result**



Showing Days from Other Months

No date selected.

August 2013

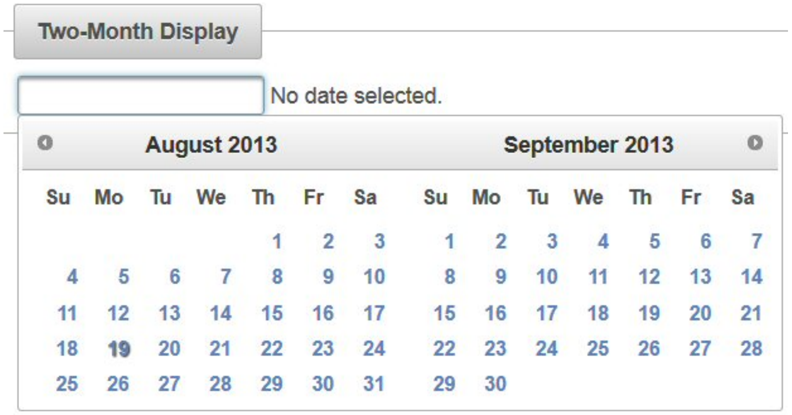
Su	Mo	Tu	We	Th	Fr	Sa
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Multi-Month Display

- **Code**

```
<p:calendar value="#{dateBean2.sampleDate}"
            pages="2"/>
```

- **Result**



Two-Month Display

No date selected.

August 2013

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

September 2013

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Animation Effects

Idea

- **Display/hide calendar in various ways**
  - Default is to make it appear/disappear instantly
  - But can use any of the jQueryUI animation effects so that calendar slides in from left, explodes after selection, etc.
    - The best way to get a feel for what each effect does is to run them interactively. The example in this section uses *every* available effect, so go to http://www.coreservlets.com/JSF-Tutorial/primefaces/, navigate to section on date input, then visit online app.
- **Basic syntax**
  - <p:calendar … effect="slide"/>
- **Components that support effects**
  - p:dialog, p:imageSwitch, p:orderList, p:overlayPanel, p:pickList, p:selectOneMenu, p:tooltip
  - But also see documentation for p:effect, which lets you add effects to other elements.

Available Effects

- **blind**
  - Gradually shows more/less of image.
    - For the difference between “blind” and “clip”, think of a window with a face showing in it. With “blind”, the face stays in the same position but the blind being raised/lowered shows more/less of the face. With “clip”, the blind is totally open, but the face moves up or down.
- **bounce**
  - Shows the image, then bounces it up and down.
- **clip**
  - Gradually shows more/less of image. See “blind”.
- **drop**
  - Drops image in from left. Looks identical to “slide”.

Available Effects (Continued)

- **explode**
  - Image gets bigger and breaks into pieces
- **fade (or fadeIn)**
  - Increases opacity until visible
- **fold**
  - Appearance like unfolding a piece of paper
- **highlight**
  - Image briefly highlighted in yellow
- **puff**
  - To disappear, image gets bigger and bigger

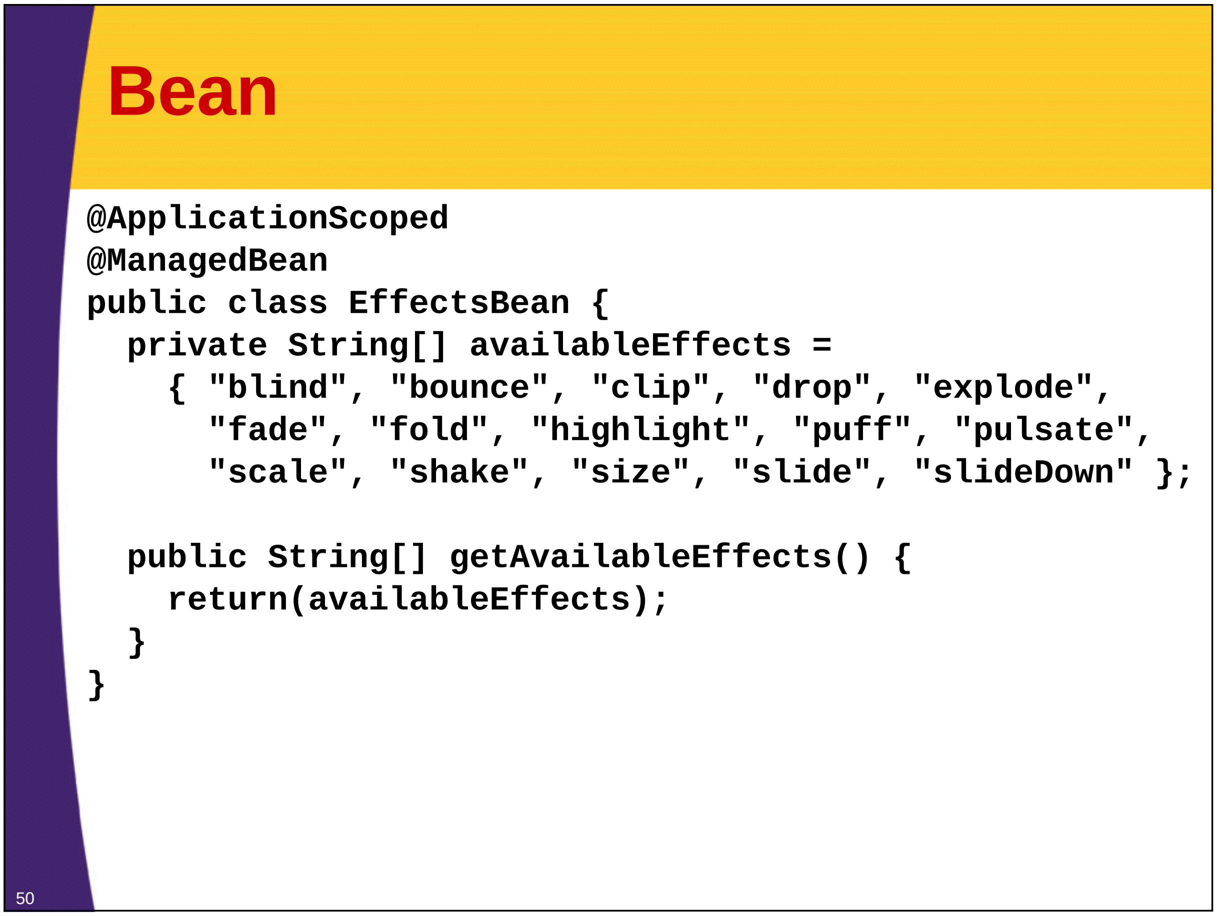
Available Effects (Continued)

- **pulsate**
  - Image flashes on/off
- **scale**
  - To disappear, gets smaller and smaller
- **shake**
  - Like “pulsate”, but left/right instead of up/down
- **size**
  - Supposedly like “scale”, but effect hard to see
- **slide**
  - Slides in from the left.
    - To me, looks identical to “drop”

- **slideDown**
  - Slides down from top instead of left

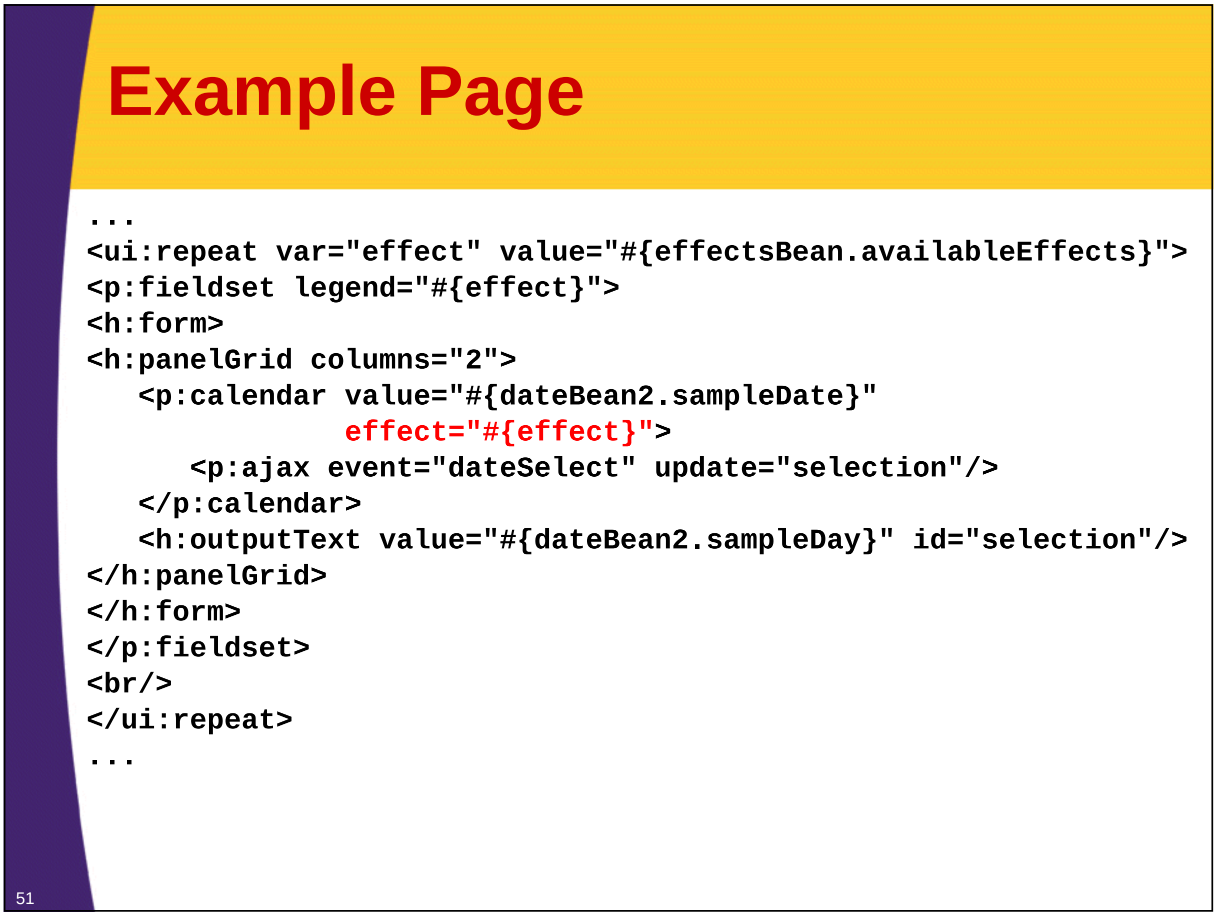
Notes on Effects

- **Documentation on effects**
  - Details on each effect at http://docs.jquery.com/UI/Effects
    - Again, to see effects in action, visit http://www.coreservlets.com/JSF-Tutorial/primefaces/, navigate to section on date input, then visit online app.
      - » Or jump directly to http://www.apps.jsf2.com/primefaces-date-input/
- **Errors in User’s Guide re effects for p:calendar**
  - The official PrimeFaces User’s Guide gives only a partial list of the supported effects. It omits explode, fade, highlight, puff, pulsate, scale, and size, even though all those effects are supported by p:calendar. Also, the User’s Guide mis-labels “slide” as “slideIn”.



Bean

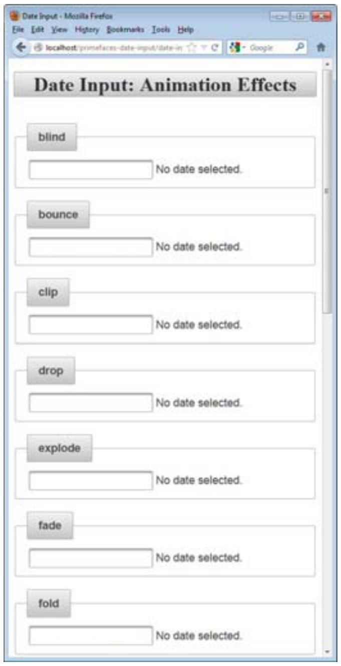
```
@ApplicationScoped
@ManagedBean
public class EffectsBean {
  private String[] availableEffects =
    { "blind", "bounce", "clip", "drop", "explode",
      "fade", "fold", "highlight", "puff", "pulsate",
      "scale", "shake", "size", "slide", "slideDown" };

  public String[] getAvailableEffects() {
    return(availableEffects);
  }
}
```



Example Page

```
...
<ui:repeat var="effect" value="#{effectsBean.availableEffects}">
<p:fieldset legend="#{effect}">
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               effect="#{effect}">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleDay}" id="selection"/>
</h:panelGrid>
</h:form>
</p:fieldset>
<br/>
</ui:repeat>
...
```

Results


Date Input - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-date-input/date-in
Google

Date Input: Animation Effects

blind

No date selected.

bounce

No date selected.

clip

No date selected.

drop

No date selected.

explode

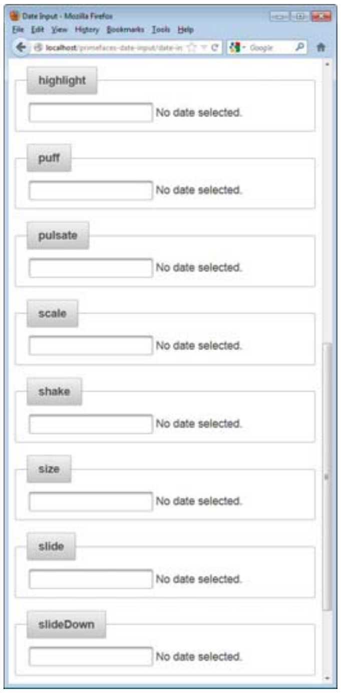
No date selected.

fade

No date selected.

fold

No date selected.


Date Input - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-date-input/date-in
Google

highlight

No date selected.

puff

No date selected.

pulsate

No date selected.

scale

No date selected.

shake

No date selected.

size

No date selected.

slide

No date selected.

slideDown

No date selected.

To see effects in action, visit
http://www.coreservlets.com/JSF-Tutorial/primefaces
navigate to section on date input, then visit online ap
Or, jump directly to
http://www.apps.jsf2.com/primefaces-date-input/.

Choosing Times

Idea

- **p:calendar can collect times as well as dates**
  - Use “pattern” to tell p:calendar what info to collect
  - Use timeOnly="true" if you want to *only* select a time
    - Day will default to Jan 1, 1970, but there will be no calendar for choosing or changing the day
  - The bean property is still of type java.util.Date, since the Date class already stores times
- **Examples**
  - Collect a date and a time in 24-hour format
    - <p:calendar value="…" pattern="MM/dd/yyyy HH:mm"/>
  - Collect a date and time in 12-hour format with am/pm
    - <p:calendar value="…" pattern="MM/dd/yyyy hh:mm a"/>
  - Collect only a time (no user choice of day)
    - <p:calendar value="…" pattern="HH:mm" timeOnly="true"/>

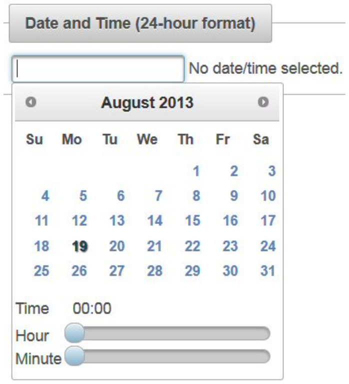
Bean

```
@ManagedBean
public class DateBean2 {
  private Date sampleDate;

  public Date getSampleDate() {
    return (sampleDate);
  }
  public void setSampleDate(Date sampleDate) {
    this.sampleDate = sampleDate;
  }

  public String getSampleTime() {
    if (sampleDate == null) {
      return("No date/time selected.");
    } else {
      String message =
        String.format("You chose '%s'.",
                      DateUtils.formatTime(sampleDate));
      return(message);
    }
  }
}
```

This is the same bean shown in earlier example. Bean property remains unchanged (i.e., of type Date), but added a new method for formatting the date with time included.

Date and Time (24-hour Format)

```
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               pattern="MM/dd/yyyy HH:mm">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleTime}" id="selection"/>
</h:panelGrid>
</h:form>
```



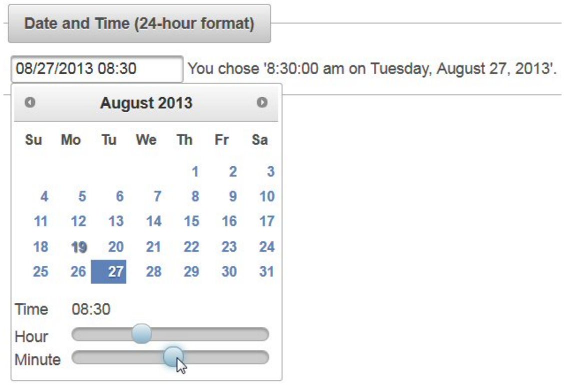
Date and Time (24-hour format)

No date/time selected.

August 2013

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Time 00:00



Date and Time (24-hour format)

08/27/2013 08:30 You chose '8:30:00 am on Tuesday, August 27, 2013'.

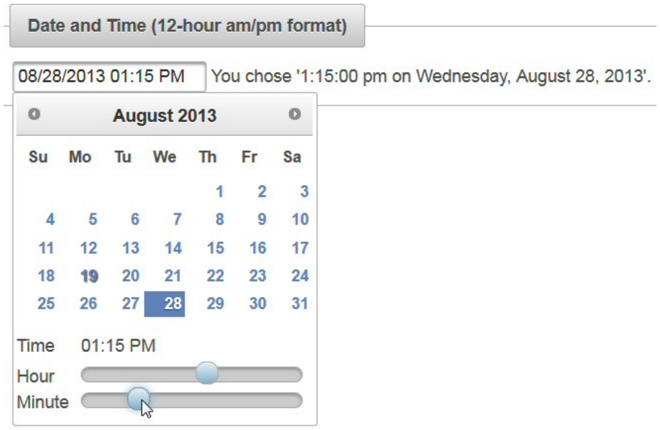
August 2013

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Time 08:30

Date and Time (12-hour Format with AM/PM)

```
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               pattern="MM/dd/yyyy hh:mm a">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleTime}" id="selection"/>
</h:panelGrid>
</h:form>
```



Date and Time (12-hour am/pm format)

08/28/2013 01:15 PM You chose '1:15:00 pm on Wednesday, August 28, 2013'.

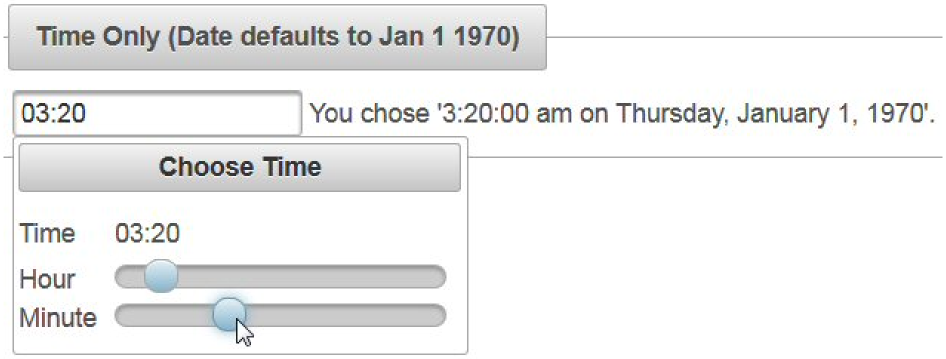
August 2013

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Time 01:15 PM

Time Only

```
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               pattern="HH:mm" timeOnly="true">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleTime}" id="selection"/>
</h:panelGrid>
</h:form>
```

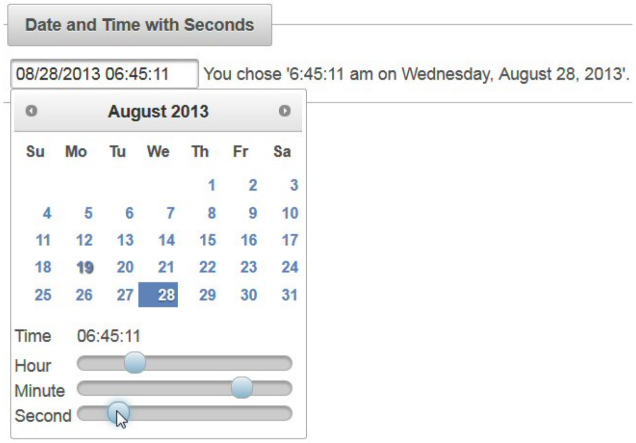


Time Only (Date defaults to Jan 1 1970)

03:20 You chose '3:20:00 am on Thursday, January 1, 1970'.

Date and Time with Seconds

```
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               pattern="MM/dd/yyyy HH:mm:ss">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleTime}" id="selection"/>
</h:panelGrid>
</h:form>
```



Date and Time with Seconds

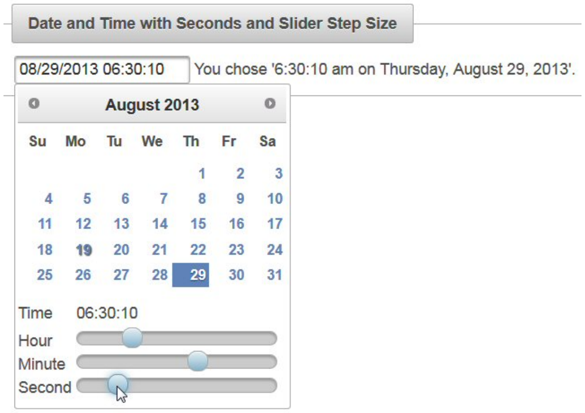
08/28/2013 06:45:11 You chose '6:45:11 am on Wednesday, August 28, 2013'.

August 2013

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Slider Step Size

```
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               pattern="MM/dd/yyyy HH:mm:ss"
               stepHour="2" stepMinute="10" stepSecond="10">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleTime}" id="selection"/>
</h:panelGrid>
</h:form>
```

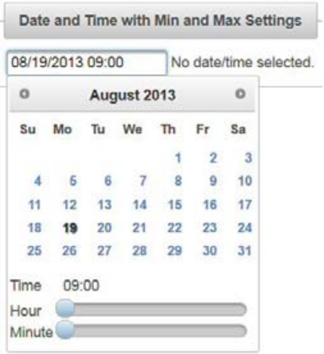


Date and Time with Seconds and Slider Step Size

08/29/2013 06:30:10 You chose '6:30:10 am on Thursday, August 29, 2013'.

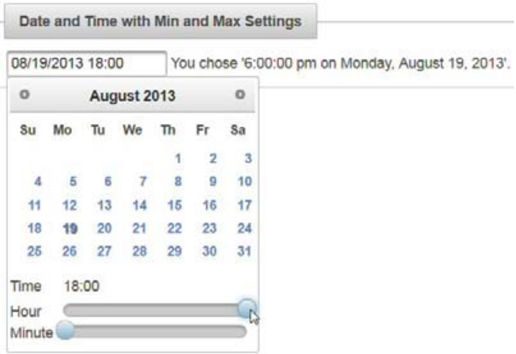
Min and Max Settings

```
<h:form>
<h:panelGrid columns="2">
   <p:calendar value="#{dateBean2.sampleDate}"
               pattern="MM/dd/yyyy HH:mm"
               minHour="9" maxHour="18">
      <p:ajax event="dateSelect" update="selection"/>
   </p:calendar>
   <h:outputText value="#{dateBean2.sampleTime}" id="selection"/>
</h:panelGrid>
</h:form>
```



Date and Time with Min and Max Settings

08/19/2013 09:00 No date/time selected.



Date and Time with Min and Max Settings

08/19/2013 18:00 You chose '6:00:00 pm on Monday, August 19, 2013'.

Wrap-Up

Summary

- **Basics**

```
<p:calendar value="#{someBean.dateProperty}"/>
```

  - With p:commandButton, use ajax="false" if you want non-Ajax page navigation

- **Ajax updates**

```
<p:calendar value="#{someBean.dateProperty}">
   <p:ajax event="dateSelect" update="…"/>
</p:calendar>
```

- **Options**

```
<p:calendar value="#{someBean.dateProperty}"
                     mode="inline" showButtonPanel="true"
                     showOn="both" showOtherMonths="true"
                     pages="2" navigator="true"
                     pattern="…" timeOnly="true"/>
```

Questions?

More info:
http://www.coreservlets.com/JSF-Tutorial/jsf2/ – JSF 2.2 tutorial
http://www.coreservlets.com/JSF-Tutorial/primefaces/ – PrimeFaces tutorial
http://courses.coreservlets.com/jsf-training.html – Customized JSF and PrimeFaces training courses
http://coreservlets.com/ – JSF 2, PrimeFaces, Java 7 or 8, Ajax, jQuery, Hadoop, RESTful Web Services, Android, HTML5, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training