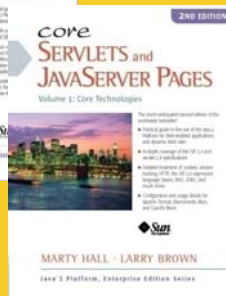
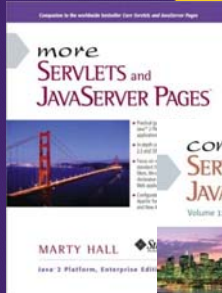




The Google Web Toolkit (GWT): Extended GUI Widgets (GWT 2.5 Version)

Originals of Slides and Source Code for Examples:
<http://courses.coreservlets.com/Course-Materials/gwt.html>

Customized Java EE Training: <http://courses.coreservlets.com/>
GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live Ajax & GWT training, see training courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. 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, servlets/JSP, Ajax, jQuery, Android development, Java 7 or 8 programming, custom mix of topics
 - Courses available in any state or country. Maryland/DC area companies can also choose afternoon/evening courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, GWT, Hadoop, HTML5, RESTful Web Services

Contact hall@coreservlets.com for details



Topics in This Section

- **Popup messages**
 - DialogBox
 - PopupPanel
 - Native dialog boxes
- **Fancy text boxes**
 - Date input boxes
 - Suggest (autocompleter) text boxes
- **Menus**
- **Third-party widget libraries**

5

© 2013 Marty Hall & Yaakov Chaikin



Popup Messages

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

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Overview

- **DialogBox**
 - Div that looks like a popup window
 - Styled caption at the top
 - Under the caption is another Widget (usually a Panel)
 - Size can be changed independently of the contained Widget
 - Can be dragged by the end user (but not outside browser)
- **PopupPanel**
 - Stripped down, simpler-looking popup window
 - No separate caption
 - Size determined by the Widget it contains
- **Native dialog (alert) box**
 - Calls native JavaScript “alert”. A “real” separate window.
 - Plain text only

7

DialogBox: Syntax Summary

- **Constructors**
 - DialogBox(), DialogBox(true)
 - If no args or false, you have to programmatically close it.
 - If true, clicking outside dialog closes it.
- **Methods**
 - setWidget
 - Sets the main content
 - setText, setHTML
 - Sets the caption (as plain text or HTML-styled text)
 - setPopupPosition(x, y),
 - Sets location of top left corner in pixels
 - setSize, setWidth, setHeight – String based
 - Sets size in CSS units (e.g., "300px")
 - setAnimationEnabled(true)
 - Open/closing is animated instead of immediate
 - show(), showRelativeTo(UIObject)
 - Open dialog box, or sets position near to Widget and then open
 - hide()
 - Close dialog box

8

DialogBox: Example (Main App)

```
public void onModuleLoad() {
    addPanelForPopups();
    ...
}

private void addPanelForPopups() {
    HorizontalPanel buttonPanel = new HorizontalPanel();
    buttonPanel.setSpacing(10);
    Button button1 = new Button("DialogBox");
    button1.addClickHandler(new DialogHandler());
    buttonPanel.add(button1);
    ...
    RootPanel.get("popup-buttons").add(buttonPanel);
}
```

Create and pop up DialogBox when button is clicked.

9

DialogBox: Example (Creating DialogBox)

```
public class DialogHandler implements ClickHandler {
    public void onClick(ClickEvent event) {
        DialogBox dialog = new DialogBox(true); // true means close DialogBox when user clicks outside
        String text = "A DialogBox ... ."; // String does not need to contain <br/> tags since you can
        Widget sampleContent = // give explicit size to the DialogBox (and text will wrap).
            new PopupSampleContentPanel(text, dialog); // The sample content is a VerticalPanel
        dialog.setWidget(sampleContent);
        dialog.setText("Dialog Box Caption"); // Text in title bar
        UIObject button = (UIObject)event.getSource();
        int x = button.getAbsoluteLeft() + 100;
        int y = button.getAbsoluteTop() - 100;
        dialog.setPopupPosition(x, y); // Open DialogBox above and to the right of button
        dialog.setAnimationEnabled(true); // Open/close DialogBox slowly, not instantaneously
        dialog.setWidth("350px"); // Use CSS units (a String), not an int
        dialog.show(); // Pop up DialogBox
    }
}
```

10

DialogBox: Example (Widget Inside)

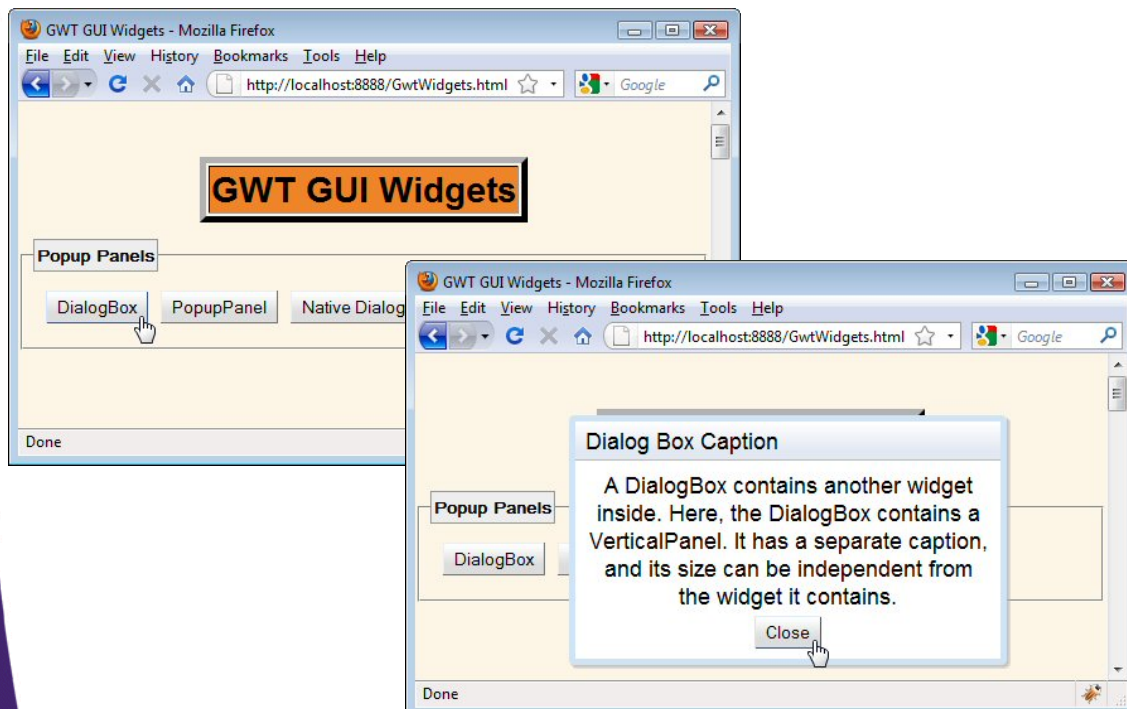
```
public class PopupSampleContentPanel extends VerticalPanel {
    public PopupSampleContentPanel(String text,
                                    final PopupPanel container) {
        setSpacing(5);
        setHorizontalAlignment(HasHorizontalAlignment.ALIGN_CENTER);
        add(new HTML(text));
        Button button = new Button("Close");
        button.addClickHandler(new ClickHandler() {
            public void onClick(ClickEvent event) {
                container.hide();
            }
        });
        add(button);
    }
}
```

In general Java, if an anonymous inner class refers to an outside local variable, that variable must be declared final. (Java 8 introduced the idea of "effectively final", but GWT won't support Java 8 for the foreseeable future.)

Close DialogBox when button clicked.
Clicking outside boundaries of DialogBox
also closes it, since true was passed to the
constructor.

11

DialogBox Example: Results (Production Mode)



12

PopupPanel: Syntax Summary

- **Constructors**

- PopupPanel(), PopupPanel(true)
 - If no args or false, you have to programmatically close it.
 - If true, clicking outside panel closes it.

- **Methods**

- setWidget
 - Sets the main content
- setPopupPosition(x, y)
 - Sets location of top left corner in pixels
- setAnimationEnabled(true)
 - Open/closing is animated instead of immediate
- show(), showRelativeTo(UIObject)
 - Open panel, or sets position near to Widget and then open
- hide()
 - Close panel

Note the lack of meaningful support for setSize, setWidth, and setHeight. The size is determined by the Widget it contains.

13

PopupPanel: Example (Main App)

```
public void onModuleLoad() {
    addPanelForPopups();
    ...
}

private void addPanelForPopups() {
    HorizontalPanel buttonPanel = new HorizontalPanel();
    ...
    Button button2 = new Button("PopupPanel");
    button2.addClickHandler(new PopupHandler());
    buttonPanel.add(button2);
    ...
    RootPanel.get("popup-buttons").add(buttonPanel);
}
```

Pop up alert when button is clicked.

14

PopupPanel: Example (Creating Popup)

```
public class PopupHandler implements ClickHandler {
    public void onClick(ClickEvent event) {
        PopupPanel popup = new PopupPanel(true); true means close DialogBox when user clicks outside
        String text = "A PopupPanel ... ."; String must contain <br/> tags. Panel size comes from widget size.
        Widget sampleContent =
            new PopupSampleContentPanel(text, popup); The sample content is a VerticalPanel
        popup.setWidget(sampleContent);
        UIObject button = (UIObject)event.getSource();
        int x = button.getAbsoluteLeft() + 100;
        int y = button.getAbsoluteTop() - 100;
        popup.setPopupPosition(x, y); Open DialogBox above and to the right of button
        popup.setAnimationEnabled(true); Open/close DialogBox slowly, not instantaneously
        popup.show(); Pop up DialogBox
    }
}
```

15

PopupPanel: Example (Widget Inside)

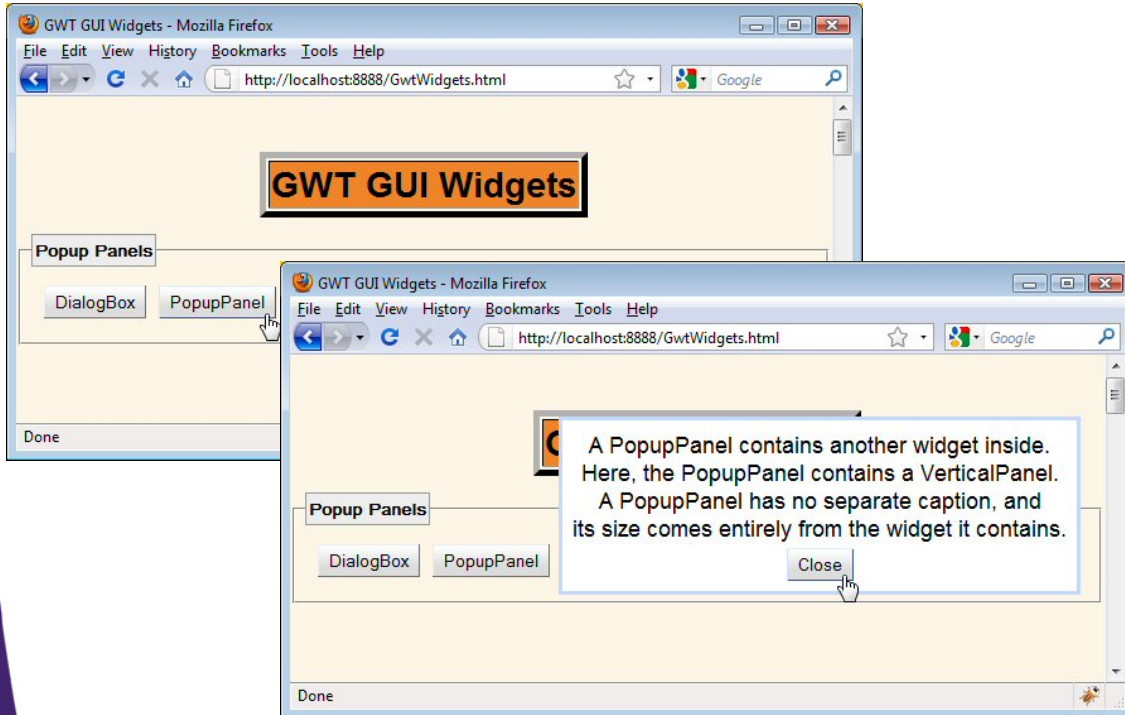
```
public class PopupSampleContentPanel extends VerticalPanel {
    public PopupSampleContentPanel(String text,
        final PopupPanel container) {
        setSpacing(5);
        setHorizontalAlignment(HasHorizontalAlignment.ALIGN_CENTER);
        add(new HTML(text));
        Button button = new Button("Close");
        button.addClickHandler(new ClickHandler() {
            public void onClick(ClickEvent event) {
                container.hide();
            }
        });
        add(button);
    }
}
```

In general Java, if an anonymous inner class refers to an outside local variable, that variable must be declared final.

Close PopupPanel when button clicked.
Clicking outside boundaries of PopupPanel also closes it, since true was passed to the constructor.

16

PopupPanel Example: Results (Production Mode)



17

Native Dialog (Alert Box): Syntax Summary

- **Window.alert("text")**
 - That's it. This is just a pass-through to the native JavaScript "alert" function.
 - This is *much* more limited than DialogBox, and DialogBox is preferred in almost all cases
 - However, this does create a "real" window that can appear outside of the browser window

18

Native Alert Window: Example (Main App)

```
public void onModuleLoad() {
    addPanelForPopups();
    ...
}

private void addPanelForPopups() {
    HorizontalPanel buttonPanel = new HorizontalPanel();
    ...
    Button button3 =
        new Button("Native Dialog (Alert) Box");
    button3.addClickHandler(new AlertHandler());
    buttonPanel.add(button3);
    ...
    RootPanel.get("popup-buttons").add(buttonPanel);
}

```

Create and pop up PopupPanel when button is clicked.

19

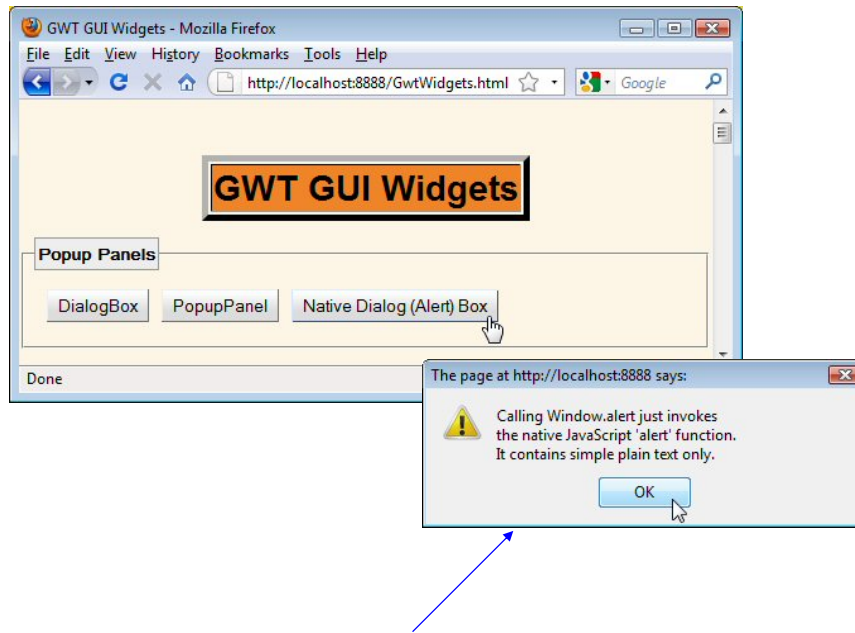
Native Alert Window: Example (Creating Alert Box)

```
public class AlertHandler implements ClickHandler {
    public void onClick(ClickEvent event) {
        String text =
            "Calling Window.alert just invokes\n" +
            "the native JavaScript 'alert' function.\n" +
            "It contains simple plain text only.";
        Window.alert(text);
    }
}

```

20

Native Alert Example: Results (Production Mode)



This is a real separate window, not a div that looks like a window.

21

© 2013 Marty Hall & Yaakov Chaikin



Fancy Text Boxes

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

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Overview

- **DateBox**
 - Textfield that pops up DatePicker when user clicks inside. The `getValue` and `setValue` methods return/accept Date objects, not raw Strings.
- **SuggestBox**
 - An autocompleting textfield. You attach an “Oracle” that contains the list of possible matches.
- **RichTextArea**
 - Aimed at formatted text input, but has *very* poor cross-browser compatibility and should usually be avoided.
 - Not shown in this tutorial.

23

DateBox: Syntax Summary

- **Constructors**
 - `DateBox()`
 - Creates a textfield for entering a date. If you call `setValue` later, that value is displayed. Otherwise initially empty.
- **Methods**
 - `setValue(Date d)`
 - Sets the internal value to the given Date, puts String representing date into textfield
 - Java Calendar is not supported – for date manipulation use GWT’s `CalendarUtil` class
 - `getValue()`
 - Retrieves the internal value as a Date.

24

DateBox: Example (Main App)

```
public void onLoadModuleLoad() {
    addDatePanel(); ...
}

private void addDatePanel() {
    Grid datePanel = new Grid(2, 2);
    datePanel.setText(0, 0, "Departure Date:");
    DateBox departureBox = new DateBox();
    Date departureDate = new Date();
    departureBox.setValue(departureDate);
    datePanel.setWidget(0, 1, departureBox);
    datePanel.setText(1, 0, "Return Date:");
    Date returnDate = new Date(departureDate.getTime());
    CalendarUtil.addDaysToDate(returnDate, 7); // add a week
    DateBox returnBox = new DateBox();
    returnBox.setValue(returnDate);
    datePanel.setWidget(1, 1, returnBox);
    RootPanel.get("date-panel").add(datePanel);
}
```

We can't use the Calendar class because it is not supported by GWT. GWT provides this class instead.

25

DateBox Example: Results (Production Mode)

The screenshot shows two browser windows displaying the application. The left window shows the application with five panels (Panel 2 to Panel 5) and two date input boxes. The right window shows the application with a calendar overlay for January 2010, with the date 4 selected. The date input boxes show the departure date as Jan 4, 2010 2:58:15 PM and the return date as Jan 11, 2010 2:58:15 PM.

	S	M	T	W	T	F	S
Panel 2	27	28	29	30	31	1	2
Panel 3	3	4	5	6	7	8	9
Panel 4	10	11	12	13	14	15	16
Panel 5	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	31	1	2	3	4	5	6

26

SuggestBox: Syntax Summary

- **Constructors**

- SuggestBox(SuggestOracle possibleChoices)
 - Creates an autocompleting textfield that shows matches against the possible choices

- **Methods**

- getText
 - Gets current textfield value. Current value need not be one of (or even part of) the possible choices from the Oracle.
- setText()
 - Changes the textfield value
- getTextBox
 - Gets the underlying TextBox

27

SuggestOracle: Syntax Summary

- **Constructors**

- MultiWordSuggestOracle()
 - A set of possible matches based on simple words (matches will be based on the beginning of the word)
- SuggestOracle
 - A raw SuggestOracle can be extended if you want to get choices from the server instead of from a fixed list

- **Methods**

- add(String possibleMatch)
 - For MultiWordSuggestOracle, adds a possible value
- requestSuggestions, requestDefaultSuggestions
 - For SuggestOracle, can be overridden to determine how to get matches based on custom logic (e.g., from server)

28

SuggestBox: Example (Main App)

```
public void onModuleLoad() {
    addSuggestPanel(); ...
}

private void addSuggestPanel() {
    HorizontalPanel langPanel = new HorizontalPanel();
    langPanel.setSpacing(5);
    langPanel.add(new HTML("Enter language:"));
    MultiWordSuggestOracle langChoices =
        new ProgrammingLanguageOracle();
    SuggestBox langBox =
        new SuggestBox(langChoices);
    langPanel.add(langBox);
    Button searchButton =
        new Button("Send to Google");
    searchButton.addClickHandler(new SearchHandler(langBox));
    langPanel.add(searchButton);
    RootPanel.get("suggest-panel").add(langPanel);
}
```

29

SuggestBox: Example (Oracle for Possible Matches)

```
public class ProgrammingLanguageOracle
    extends MultiWordSuggestOracle {
    // 50 most popular programming languages, listed in order.
    // From http://www.tiobe.com/tpci.htm
    private static final String languages =
        "Java,C,C++,PHP, ...";
    private static final String[] languageNames =
        languages.split(",");

    public ProgrammingLanguageOracle() {
        for(String language: languageNames) {
            add(language);
        }
    }
}
```

30

SuggestBox: Example (Button's ClickHandler)

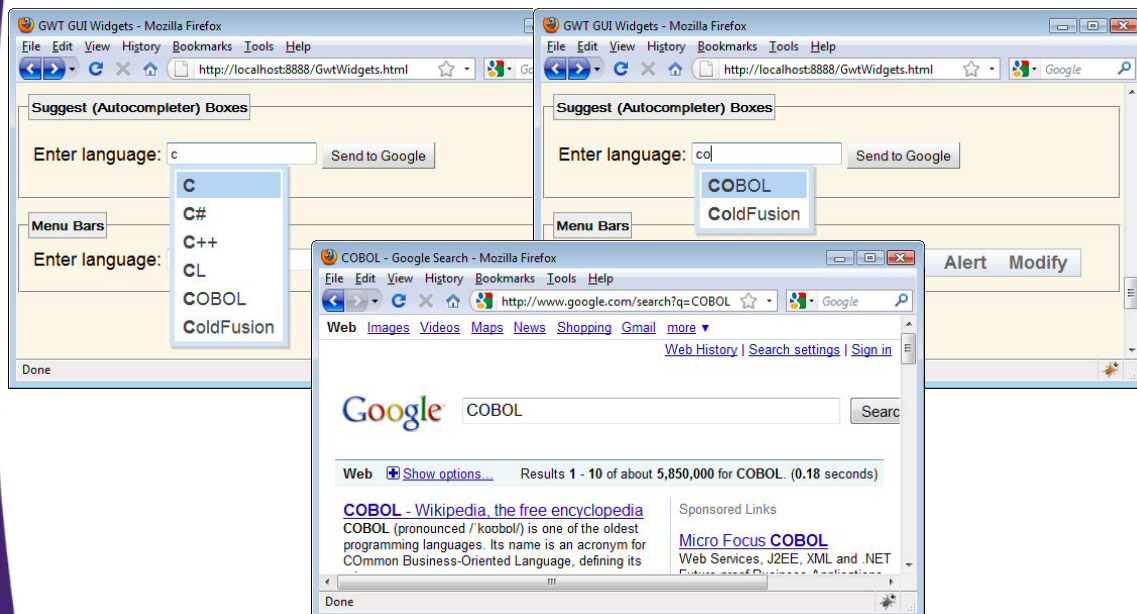
```
public class SearchHandler implements ClickHandler {  
    private SuggestBox box;  
  
    public SearchHandler(SuggestBox box) {  
        this.box = box;  
    }  
  
    public void onClick(ClickEvent event) {  
        String url = "http://www.google.com/search?q=" +  
                    URL.encode(box.getText());  
        Window.open(url, "_blank", "");  
    }  
}
```

Causes browser to leave current page and go to other page.

Use this to open another window. Useful if you have an external link, but don't want the user to forget about your site.

31

SuggestBox Example: Results (Production Mode)



32



Menus

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

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

MenuBar: Syntax Summary

- **Constructors**

- MenuBar()
 - Creates a menu bar for displaying a list of horizontal choices. Usually used for the top-level menu.
- MenuBar(true)
 - Creates a menu bar for displaying a list of vertical choices. Usually used for the submenus that will pop up below the main entries or to the right of nested entries.

- **Methods**

- addItem(String text, Command c)
 - Displays text in menu. When clicked, executes Command
- addItem(String text, MenuBar subMenu)
 - Displays text in menu. When clicked, opens subMenu

Command Interface: Syntax Summary

- **Method to implement**

- public void execute()
 - Action to take when Command invoked.
 - This method takes no arguments, so if execute needs to access data, you should pass the data to the constructor of your class that implements Command, or you should use an inner class.

35

MenuBar: Example (Main App)

```
public void onModuleLoad() {
    addMenuPanel();
    ...
}

private void addMenuPanel() {
    HorizontalPanel searchPanel =
        new HorizontalPanel();
    searchPanel.setSpacing(5);
    searchPanel.add(new HTML("Enter language:"));
    MultiWordSuggestOracle langChoices =
        new ProgrammingLanguageOracle();
    final SuggestBox langBox =
        new SuggestBox(langChoices);
    searchPanel.add(langBox);
}
```

36

MenuBar: Example (Main App, Continued)

```
MenuBar mainMenu = new MenuBar();
MenuBar searchMenu =
    new SearchMenuBar(langBox);
mainMenu.addItem("Search", searchMenu);
Command alertCommand = new Command() {
    public void execute() {
        Window.alert("Selection is " + langBox.getText());
    }
};
mainMenu.addItem("Alert", alertCommand);
MenuBar modifierMenu =
    new ModifierMenuBar(langBox);
mainMenu.addItem("Modify", modifierMenu);
searchPanel.add(mainMenu);
RootPanel.get("menu-panel").add(searchPanel);
}
```

37

MenuBar: Example (Submenu 1)

```
public class SearchMenuBar extends MenuBar {
    public SearchMenuBar(SuggestBox box) {
        super(true); // Vertical (dropdown) menu
        Command googleCommand =
            new GoogleSearchCommand(box);
        addItem("Google", googleCommand);
        Command yahooCommand =
            new YahooSearchCommand(box);
        addItem("Yahoo", yahooCommand);
        Command bingCommand =
            new BingSearchCommand(box);
        addItem("Bing", bingCommand);
    }
}
```

38

MenuBar: Example (SearchCommand Base Class)

```
public abstract class SearchCommand implements Command {
    protected SuggestBox box;

    public abstract String getBaseUrl();

    public void execute() {
        String url = getBaseUrl() +
                    URL.encode(box.getText());
        Window.Location.assign(url);
    }
}
```

39

MenuBar: Example (GoogleSearchCommand)

```
public class GoogleSearchCommand extends SearchCommand {
    public GoogleSearchCommand(SuggestBox box) {
        this.box = box;
    }

    @Override
    public String getBaseUrl() {
        return("http://www.google.com/search?q=");
    }
}
```

YahooSearchCommand and BingSearchCommand differ only in their
baseUrl. Full code can be downloaded from online tutorial.

40

MenuBar: Example (Submenu 2)

```
public class ModifierMenuBar extends MenuBar {
    public ModifierMenuBar(final SuggestBox box) {
        super(true); // Vertical (dropdown) menu
        Command upperCaseCommand = new Command() {
            public void execute() {
                box.setText(box.getText().toUpperCase());
            }
        };
        addItem("Upper Case", upperCaseCommand);
        Command lowerCaseCommand = new Command() {
            public void execute() {
                box.setText(box.getText().toLowerCase());
            }
        };
        addItem("Lower Case", lowerCaseCommand);
    }
}
```

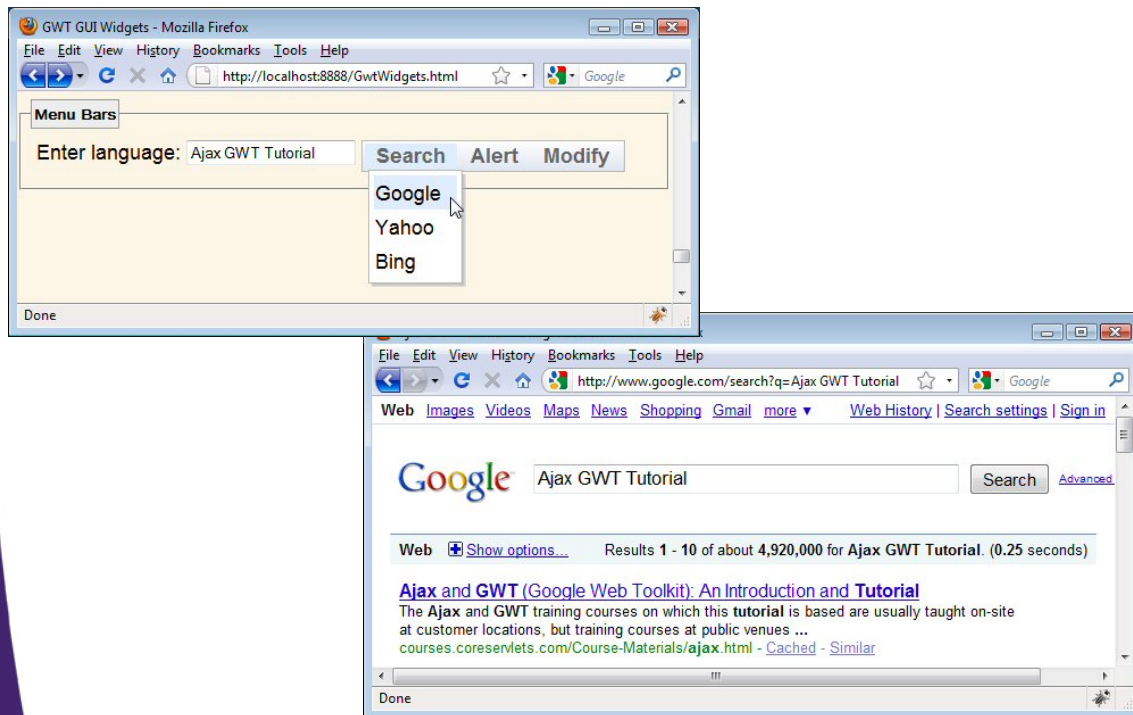
41

MenuBar: Example (Submenu 2, Continued)

```
        Command clearCommand = new Command() {
            public void execute() {
                box.setText("");
            }
        };
        addItem("Clear", clearCommand);
    }
}
```

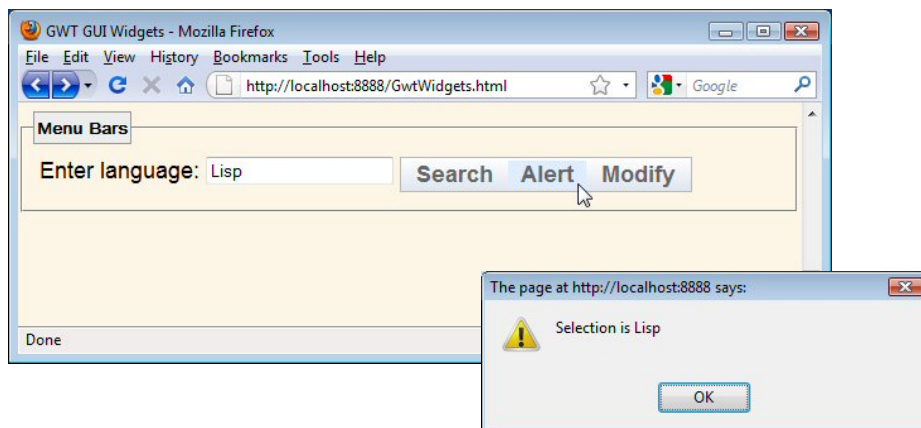
42

MenuBar Example: Results (Production Mode)



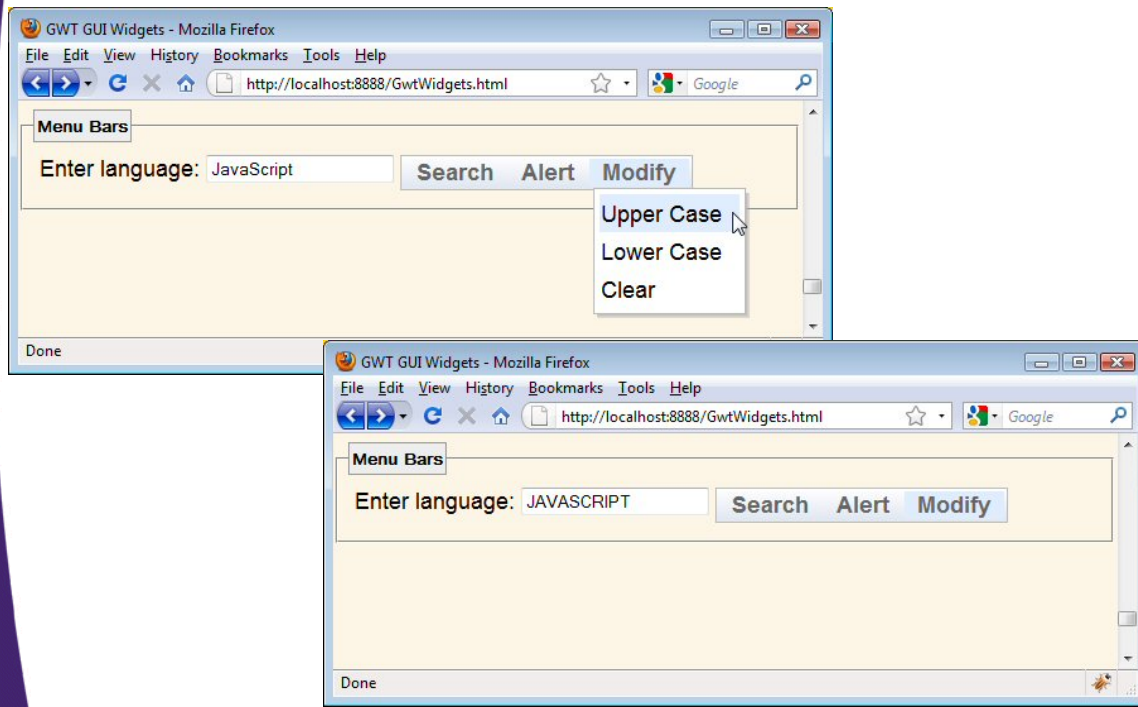
43

MenuBar Example: Results (Production Mode, Continued)



44

MenuBar Example: Results (Production Mode, Continued)



45

© 2013 Marty Hall & Yaakov Chaikin



Wrap-Up

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

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Third-Party GWT Widget Libraries

- **GWT Widget Incubator**
 - Trial for widgets that may become standard.
 - <http://code.google.com/p/google-web-toolkit-incubator/w/list>
- **Ext-GWT (GXT)**
 - Large set of rich widgets from the makers of Ext-JS. Native implementations, not just JSNI wrappers.
 - Not free for commercial use (or free with source code release)
 - <http://www.sencha.com/products/extgwt/>
- **Tatami**
 - Complete JSNI wrapper for Dojo. (only compatible with GWT 1.7 so far)
 - <http://code.google.com/p/tatami/>
- **Smart GWT**
 - GWT version of the extensive SmartClient library.
 - <http://code.google.com/p/smartgwt/>
- **GWT Component Library**
 - Small set of widgets plus complete Scriptaculous wrapper
 - <http://gwt.components.googlepages.com/>

47

Summary

- **DialogBox, PopupPanel**
 - `setWidget(content), show()`
 - `setText(caption), setHTML(caption)` [*DialogBox only*]
- **DecoratedTabPanel, DecoratedStackPanel**
 - Rounded corners without you writing CSS
- **DateBox**
 - `setValue(date), getValue()`
- **SuggestBox**
 - `new SuggestBox(multiwordOracle)`
- **MenuBar**
 - `new MenuBar(), new MenuBar(true)`
 - `addItem(text, command), addItem(text, subMenu)`

48



Questions?

[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.](#)

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

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.