



Rich Interfaces with jQuery UI: Part II: Autocompleter, Effects, Themes (jQuery UI 1.8, jQuery 1.4)

Originals of Slides and Source Code for Examples:
<http://coursescoreservlets.com/Course-Materials/jquery.html>

Customized Java EE Training: <http://coursescoreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
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://coursescoreservlets.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
 - Java 6, servlets/JSP (intermediate and advanced), Struts, JSF 1.x, JSF 2.0, Ajax, GWT 2.0 (with GXT), custom mix of topics
 - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, Google Closure) or survey several
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, Web Services, Ruby/Rails

Contact hall@coreservlets.com for details

Topics in This Section

- **Autocompleter**
 - Local data (source = array)
 - Remote data (source = URL)
 - Custom matcher (source = function)
 - Custom menu styles
 - Display values vs. values to be inserted
 - Performing side effects upon selection
- **Animation effects**
- **Changing themes**

5

© 2010 Marty Hall



Setup for Examples

Overview

- **General setup**
 - jQuery and jQuery UI loaded as in previous lecture
 - ui-lightness theme is the default, as in previous lecture
 - See section on themes for the exact HTML used
- **Specific layout**
 - Each section is in its own accordion panel
 - All buttons are styled so they take on look of theme
 - Custom HTML headings and sections use predefined CSS class names like “ui-widget-header” and “ui-widget-content” so that they take on the look of the theme

7

JavaScript for General Setup

- **Purpose**
 - Styles all the buttons
 - Sets up the accordion panel
 - Both concepts covered in previous lecture
- **Code**

```
$(function() {  
    ...  
    $("button").button();  
    $("#accordion-panel").accordion({  
        collapsible: true, active: false, autoHeight: false  
    });  
});
```

8



Autocompleter Overview

Customized Java EE Training: <http://coursescoreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Overview

- **Idea**
 - Textfield with dropdown list of matching choices
 - Like the Google and Bing home pages
- **HTML**
 - Make a textfield with an id
 - `<input type="text" id="field-id"/>`
- **Basic JavaScript**
 - Call “autocomplete()” on the textfield.
Supply a “source” option.
 - `$("#field-id").autocomplete({ source: ... });`
 - There are three variations on the type of value for “source”
 - There are many options in addition to the required “source”

Options for source

- **Local data**
 - var choices = ["Choice 1", "Choice 2", ...];
 - \$("#field-id").autocomplete({ source: choices });
- **Remote data**
 - var url = "some relative URL that returns JSON array";
 - \$("#field-id").autocomplete({ source: url });
- **Custom matching function**
 - function customMatcher(request, callback) {
 var arrayOfMatches = someLogic(request.term);
 callback(arrayOfMatches);
}
– \$("#field-id").autocomplete({ source: customMatcher });

11

Customizing Display of Values

- **Array of choices**
 - The source option can directly take an array, can refer to a URL that returns an array, or can refer to a function that creates an array and passes it to callback handler
- **Types of arrays**
 - Strings
 - The value in the dropdown and the value that gets inserted upon selection are the same
 - Objects with “label” and “value” properties
 - The “label” property is shown in the dropdown, and the “value” property gets inserted upon selection
- **Difficult to use HTML in displayed values**
 - In jQuery UI 8.1 and 8.2, values could contain HTML
 - This was disabled in 8.3. The fix is very obscure.
 - See <http://bugs.jqueryui.com/ticket/5918> for details

12

Most Important Other Options

- **select**
 - Function that operates when an item is selected.
 - See upcoming example.
- **minLength**
 - The number of characters that should be entered before suggestions are shown. Used when there are too many suggestions for short text.
 - Default is 1
- **delay**
 - The number of milliseconds after a keystroke to wait before showing suggestions
 - Default is 300

13

© 2010 Marty Hall



Autocompleter with Local Data

Overview

- **Idea**

- Textfield with dropdown list of matching choices
 - Choices come from a JavaScript array
 - Case insensitive match anywhere in the choices (not just beginning of the word)

- **HTML**

- Make a textfield with an id
 - <input type="text" id="field-id"/>

- **Basic JavaScript**

- Make an array of choices (strings)
 - var choices = ["Choice 1", "Choice 2" ...];
- Call “autocomplete()” on the textfield.
Supply the array as the “source” option.
 - \$("#field-id").autocomplete({ source: choices });

15

Example: HTML

Programming language:

```
<input type="text" id="langField1"/>
<input type="button" value="Search on Language"
      id="langSearchButton1"/>
```

16

Example: Main JavaScript

```
var uiDemo = {} ; // To avoid name conflicts

uiDemo.langString = "Java,C,C++,PHP,C#,Python..." ;

These are 100 most popular programming languages, based on data from tiobe.com.
I put into string and then split the string into an array to make it simpler to type them in.

uiDemo.langArray = uiDemo.langString.split(",") ;

$(function() {
    $("#langField1").autocomplete({
        source: uiDemo.langArray
    });
    ...
}) ;
```

Case insensitive match anywhere in the choices. So, entering "V" in the textfield will match "Java", "JavaScript", "Visual Basic", etc. See the upcoming custom matcher example for how to change the way that the match is performed.

17

Example: Auxiliary JavaScript

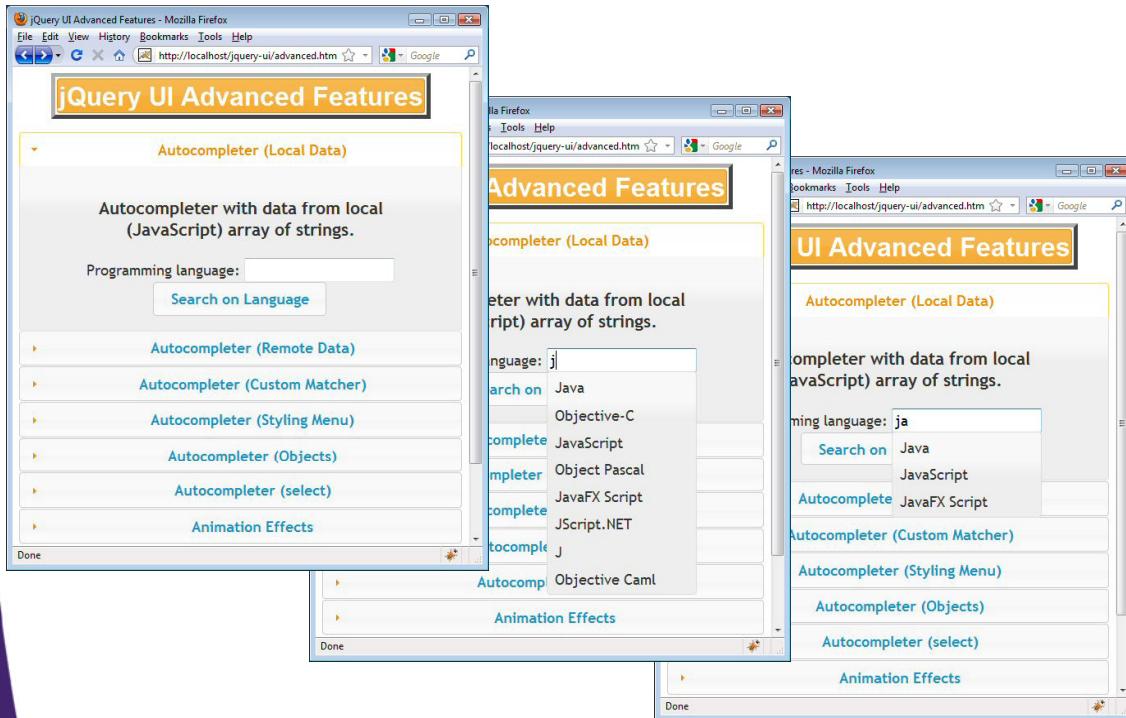
```
uiDemo.googleSearch = function(selector) {
    var language = escape($(selector).val());
    window.location.href =
        "http://www.google.com/search?q=" + language;
};

$(function() {
    $("#langSearchButton1").click(function() {
        uiDemo.googleSearch("#langField1");
    });
    ...
});
```

Just to make the example do something, the button sends the entry from the textfield to Google. However, this functionality is really independent of the autocomplete. I won't repeat the code for the Google button in the later autocomplete examples.

18

Example: Results



19

© 2010 Marty Hall



Autocompleter with Remote Data

Customized Java EE Training: <http://coursescoreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at your location.

Overview

- **Idea**
 - Textfield with dropdown list of matching choices
 - Choices come from server (sent as “term” parameter)
 - Server can match any way it wants
- **HTML**
 - <input type="text" id="field-id"/>
- **Basic JavaScript**
 - Call “autocomplete()” on the textfield.
Supply a relative URL as the “source” option.
 - \$("#field-id").autocomplete({ source: "some-address" });
 - The remote address should take a “term” request parameter and return JSON (array of strings, or, as we will see later, object array)

21

Example: HTML

Programming language:

```
<input type="text" id="langField2"/>
<input type="button" value="Search on Language"
      id="langSearchButton2"/>
```

22

Example: JavaScript

```
$ (function() {  
    $("#langField2").autocomplete({  
        source: "language-completer"  
    });  
    ...  
});
```

URL of servlet that takes "term" request parameter
and returns JSON array of strings.

23

Example: Server Code

```
public class LanguageCompleter extends HttpServlet {  
    private static final String languageString =  
        "Java,C,C++,PHP,C#,Python,Visual Basic,...";  
    private static final String[] languageArray =  
        languageString.split(",");  
  
    public void doGet(HttpServletRequest request,  
                      HttpServletResponse response)  
        throws ServletException, IOException {  
        String languagePrefix = request.getParameter("term");  
        List<String> languages = findLanguages(languagePrefix);  
        response.setContentType("application/json");  
        PrintWriter out = response.getWriter();  
        out.print(new JSONArray(languages));  
    }  
}
```

Relative URL of servlet is language-completer. This can be set either in web.xml (any servlet version) or via the @WebServlet annotation (servlets 3.0).

Uses the json.org utilities to turn Java List or array into JSON array. See separate lecture on automatic JSON generation. If you generate JSON by hand, remember that jQuery 1.4 enforces strict JSON where property names must be in quotes and only double (not single) quotes are used. See first lecture on jQuery support for Ajax.

24

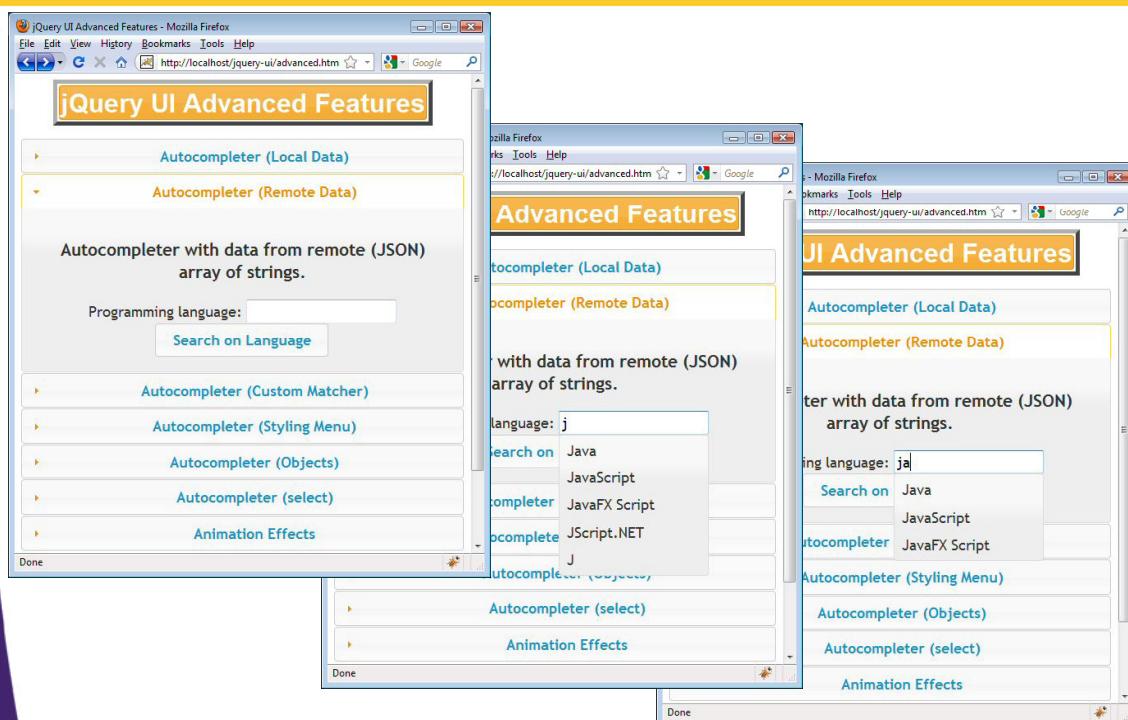
Example: Server Code (Continued)

```
private List<String> findLanguages(String languagePrefix) {  
    languagePrefix = languagePrefix.toUpperCase();  
    List<String> languages = new ArrayList<String>();  
    for(String language: languageArray) {  
        if(language.toUpperCase().startsWith(languagePrefix)) {  
            languages.add(language);  
        }  
    }  
    return(languages);  
}
```

Case insensitive match at the start of the choice. Server code can do any type of matching it wants; all that matters is that a JSON array is returned to the client.

25

Example: Results



26



Autocompleter with Custom Matcher

Customized Java EE Training: <http://coursescoreservlets.com/>
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Overview

- **Idea**
 - Textfield with dropdown list of matching choices
 - Choices come from a JavaScript function.
 - The function can use local array or remote array (e.g., via \$.ajax with dataType "json"), and it can match however it wants.
- **HTML**
 - <input type="text" id="field-id"/>
- **Basic JavaScript**
 - Make function that takes two args: request and callback. Use request.term to produce an array, then call callback(array)
 - ```
function customMatcher(request, callback) {
 var arrayOfMatches = someLogic(request.term);
 callback(arrayOfMatches);
}
```
  - Call “autocomplete()” on the textfield.  
**Supply the function as the “source” option.**
    - `$("#field-id").autocomplete({ source: customMatcher });`

## Example: HTML

Programming language:

```
<input type="text" id="langField3"/>
<input type="button" value="Search on Language"
 id="langSearchButton3"/>
```

29

## Example: JavaScript

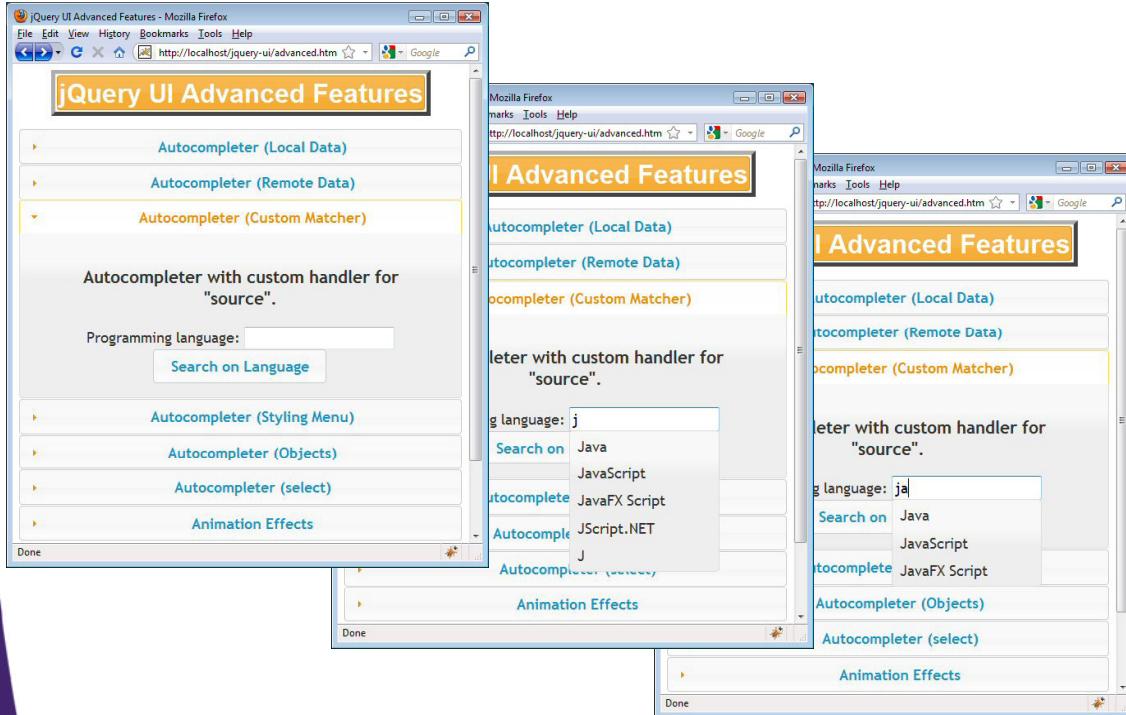
```
uiDemo.langMatcher = function(request, callback) {
 // Case-insensitive match at beginning of word.
 var regex = new RegExp("^" + request.term, "i");
 var matches = $.grep(uiDemo.langArray,
 function(item) {
 return(regex.test(item));
 });
 callback(matches);
};

$(function() {
 $("#langField3").autocomplete({
 source: uiDemo.langMatcher
 });
 ...
});
```

Note that `$.grep` is a jQuery utility function that returns an array of all elements that match `test`.  
Also note that `someRegEx.test` is a builtin method in normal JavaScript. Finally, if you want to worry  
about the user entering characters like `.` or `*` in the textfield, you can escape them using  
`$.ui.autocomplete.escapeRegex(request.term)`.

30

# Example: Results



31

© 2010 Marty Hall



## Autocompleter with CSS Styles Applied

Customized Java EE Training: <http://coursescoreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.  
Developed and taught by well-known author and developer. At public venues or onsite at your location.

# Overview

- **Idea**

- You can apply your own CSS styles to the dropdown box
  - Use ".ui-autocomplete" or ".ui-autocomplete.ui-menu"
- Caution
  - Avoid fg and bg colors that would clash with the theme
    - Reasonable options: setting max-height (to support scrolling), setting opacity for semi-transparent menus, etc.

- **CSS**

- .ui-autocomplete { ... }

- **JavaScript**

- If you apply styles directly as above, no extra JavaScript needed. You can also set a custom style, then apply it at runtime to the drop down
  - \$(".ui-autocomplete").toggleClass("some-custom-name");

33

## Example: HTML

```
<input type="text" id="langField4"/>
<input type="button" value="Search on Language"
 id="langSearchButton4"/>

<input type="button" value="Toggle Menu Style"
 id="toggle-style-button"/>
```

34

## Example: CSS

```
.fancy-menu {
 opacity: 0.8;
 max-height: 140px;
 overflow-y: auto;
}
```

If applied, results in a semi-transparent menu that has scrollbars if there are more than about 4 entries.

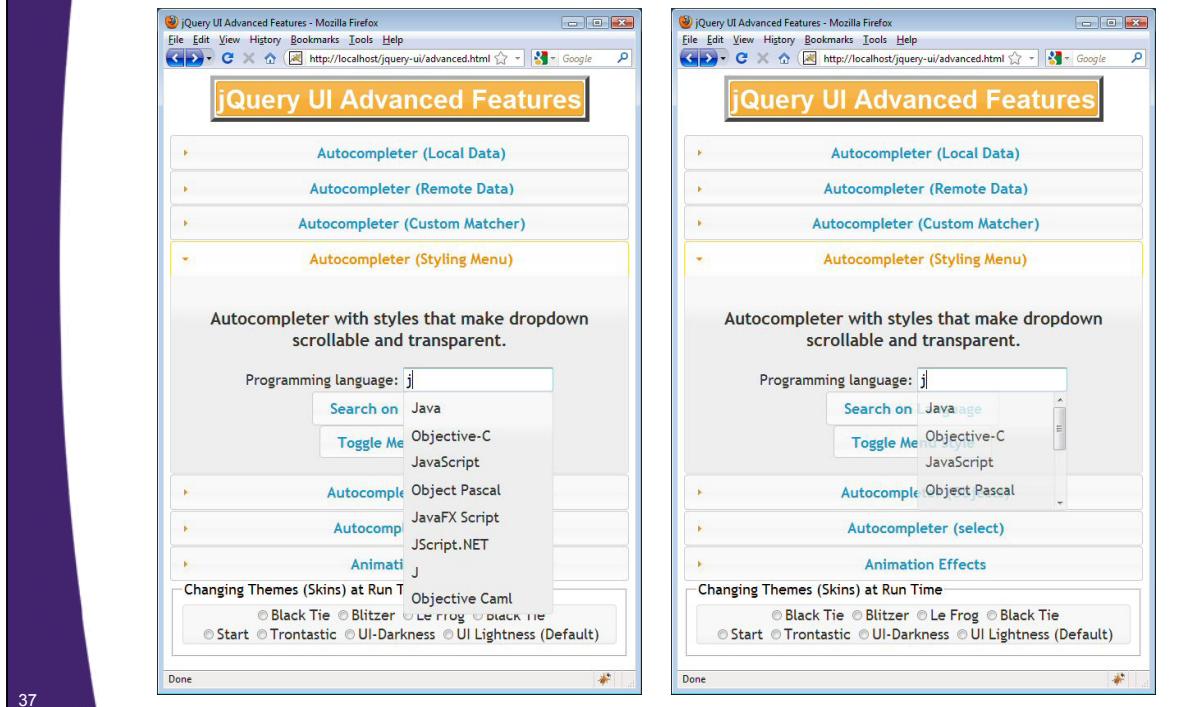
35

## Example: JavaScript

```
$(function() {
 $("#langField4").autocomplete({
 source: uiDemo.langArray
 });
 $("#langSearchButton4").click(function() {
 uiDemo.googleSearch("#langField4");
 });
 $("#toggle-style-button").click(function() {
 $(".ui-autocomplete").toggleClass("fancy-menu");
 });
 ...
});
```

36

# Example: Results (Before and After Styling)



37

© 2010 Marty Hall



## Autocompleter with Array of Objects

Customized Java EE Training: <http://coursescoreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.  
Developed and taught by well-known author and developer. At public venues or onsite at your location.

# Overview

- **Idea**
  - Textfield with dropdown list of matching choices
    - Choices come from local array or array from server
    - Array contains JavaScript objects with at least 2 properties
      - “label” – what to show in dropdown for that entry
      - “value” – what to insert in textfield if that entry is selected
- **HTML**
  - <input type="text" id="field-id"/>
- **Basic JavaScript**
  - Call “autocomplete()” on the textfield.  
**Supply array, URL, or function as the “source” option.**
    - \$("#field-id").autocomplete({ source: ... });
      - But in all 3 cases, final result is array of object literals, not strings

39

# Simplified Example

- **Idea**
  - Display dropdown values in mixed case
  - Insert into textfield in upper case
- **JavaScript code**

```
var choices = [{ label: "Java", value: "JAVA" },
 { label: "Python", value: "PYTHON" },
 ...
];
$("#field-id").autocomplete({ source: choices });
```

40

## Example: HTML

```
<fieldset>
<legend>Email the jQuery Team</legend>
...
TO:
<input type="text" id="emailField1" size="40"/>
...
(some HTML that mimics a form for sending
email to member of the jQuery team)
</fieldset>
```

When the user types into the textfield, the full names of matching members of the jQuery and jQuery UI teams will be displayed. But, when the user selects a name from the list, both the name and the email address will be inserted into the textfield.

41

## Example: JavaScript

```
$(function() {
 $("#emailField1").autocomplete({
 source: "contact-completer"
 });
 ...
});
```

URL of servlet that takes "term" request parameter  
and returns JSON array of objects that have "label"  
(full name) and "value" (full name <email@address>)  
properties.

42

## Example: Server Code

```
public class ContactCompleter extends HttpServlet {
 public void doGet(HttpServletRequest request,
 HttpServletResponse response)
 throws ServletException, IOException {
 String namePrefix = request.getParameter("term");
 List<Contact> contacts = findContacts(namePrefix);
 response.setContentType("application/json");
 PrintWriter out = response.getWriter();
 out.print(new JSONArray(contacts));
 }
}
```

Relative URL of servlet is contact-completer.

Uses the json.org utilities to turn Java List or array into JSON array. Since the Contact objects have getLabel and getValue methods, the JSON objects will have label and value properties. See separate lecture on automatic JSON generation.

43

## Example: Server Code (Continued)

```
private List<Contact> findContacts(String namePrefix) {
 namePrefix = namePrefix.toUpperCase();
 List<Contact> contacts = new ArrayList<Contact>();
 Contact[] jQueryContacts = ContactUtils.jqueryContacts();
 for(Contact contact: jQueryContacts) {
 String firstName =
 contact.getFirstName().toUpperCase();
 String lastName =
 contact.getLastName().toUpperCase();
 if(firstName.startsWith(namePrefix) ||
 lastName.startsWith(namePrefix)) {
 contacts.add(contact);
 }
 }
 return(contacts);
}
```

Case insensitive match at the start of either first or last name

44

## Example: Server Code (Contact Class)

```
public class Contact {
 private final String firstName, lastName,
 email, image;

 // Constructor and five getter methods for above vars

 public String getLabel() {
 return(firstName + " " + lastName);
 }

 public String getValue() {
 return(String.format("%s %s <%s>",
 firstName, lastName, email));
 }
}
```

The json.org utilities automatically turn getter methods into properties. So, an instance of this class will be sent as  
{ "firstName": "...", lastName: "...", ... , "label": "...", "value": "..." }

45

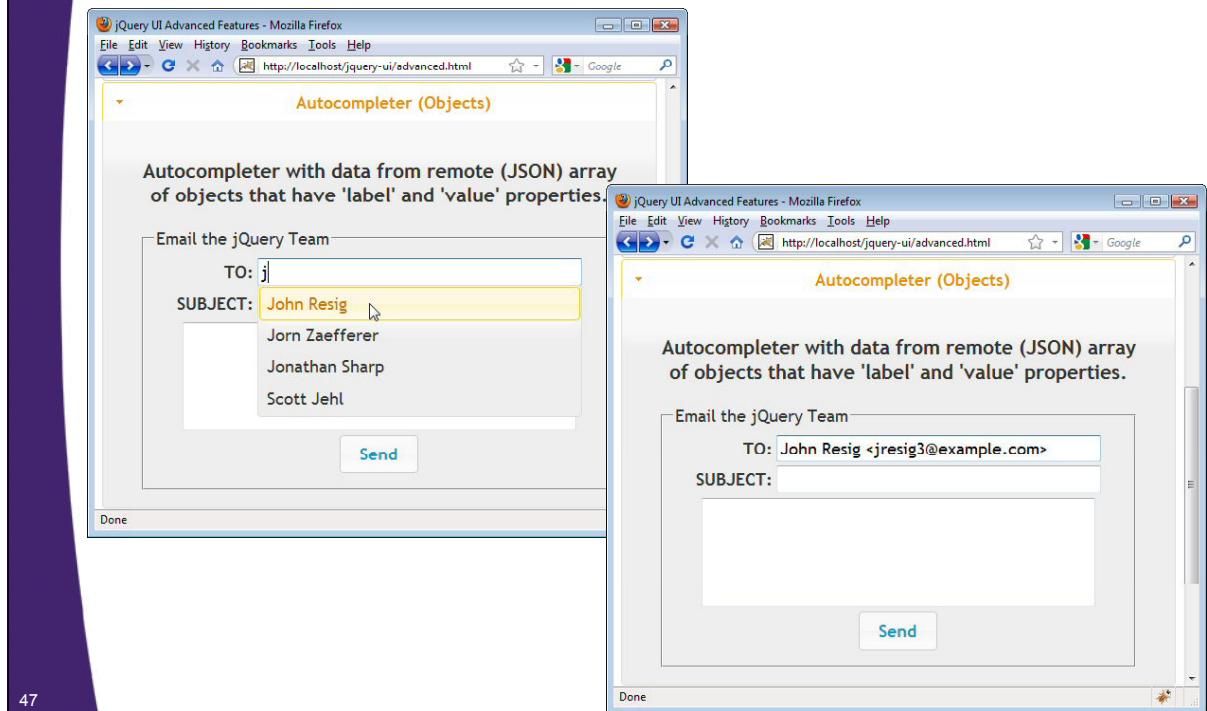
## Example: Server Code (ContactUtils)

```
public class ContactUtils {
 private static Contact[] jQueryContacts = {
 new Contact("John", "Resig", "jresig3@example.com",
 "http://static.jquery.com/.../john.jpg"),
 new Contact("Brandon", "Aaron", "brandon@example.com",
 "http://static.jquery.com/.../brandon.jpg"),
 ...
 };
```

This gives the names and pictures of all members of the jQuery and jQuery UI teams, as given at <http://jquery.org/team>.  
Email addresses are fictionalized to preserve privacy. The images are not used in this example, but will be in later example.

46

# Example: Results



47

© 2010 Marty Hall



## Autocompleter: Using “select” to Perform Side Effects

Customized Java EE Training: <http://coursescoreservlets.com/>

Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.  
Developed and taught by well-known author and developer. At public venues or onsite at your location.

# Overview

- **Idea**

- Use “select” to perform side effects
  - You can use the “select” option to tell jQuery UI to do additional side effects other than just inserting the selected entry (or “value” property of entry) into the textfield
    - Function takes event and ui object as argument. ui.item is the entry from the array. Use ui.item directly if you have an array of strings, or use ui.item.value if you have an array of objects. The objects can have extra properties with which you can perform additional tasks.

- **JavaScript**

- Call “autocomplete()”. Supply function as the “select” option.
  - ```
function selectFunct(event, ui) {
    $("#field-id").val(ui.item.value); // Insert into textfield
    $("#other-id").doSomething(ui.item.otherProp);
}
```
 - ```
$("#field-id").autocomplete({ source: ..., select: selectFunct });
```

49

# Simplified Example

- **Idea**

- In dropdown, show list of model numbers
- When one selected, insert into textfield in upper case
  - But also put text description in page

- **JavaScript code**

```
var choices =
 [{ label: "a1234", value: "A1234", desc: "Cool Widget" },
 { label: "b1234", value: "B1234", desc: "..." }, ...];
function selectHandler(event, ui) {
 $("#field-id").val(ui.item.value); // Insert into textfield
 $("#description-id").html(ui.item.desc);
}
$("#field-id").autocomplete({ source: choices,
 select: selectHandler });
```

50

## Example: HTML

```
<fieldset>
<legend>Email the jQuery Team</legend>
...
TO:
<input type="text" id="emailField2" size="40"/>
...

(some HTML that mimics a form for sending email to member of the jQuery team)
</fieldset>
```

When the user types into the textfield, the full names of matching members of the jQuery and jQuery UI teams will be displayed. When the user selects a name from the list, both the name and the email address will be inserted into the textfield. In addition, the picture of the corresponding person will be shown next to the textfield.

51

## Example: JavaScript

```
$(function() {
 $("#emailField2").autocomplete({
 source: "contact-completer",
 select: function(event, ui) {
 $("#emailField2").val(ui.item.value);
 $("#email-image").attr("src", ui.item.image);
 }
 });
 ...
});
```

Put "value" property of item (i.e., "Joe User <user@example.com>") into textfield

Set src attribute of image to the "image" property of the selected item

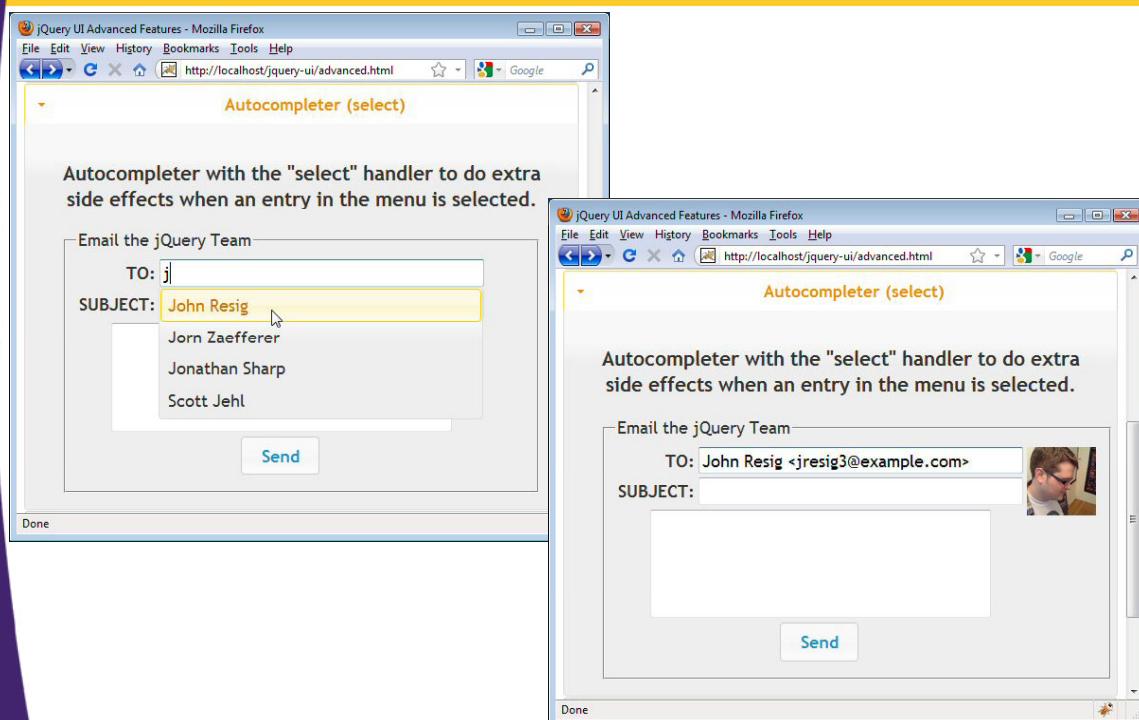
52

# Example: Server Code

- **Unchanged from previous example**
  - In last example, we returned a JSON array of objects that had label, value, and image properties.
    - Client code ignored the image property.
  - We will use the same server code in this example
    - But this time the client code will take the image property and use it for the src attribute of an image to the right of the input field.

53

# Example: Results



54



# Animation Effects

Customized Java EE Training: <http://coursescoreservlets.com/>  
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Overview

- **Idea**
  - You can make HTML elements appear and disappear in various interesting ways. You can also highlight elements
- **Basic JavaScript**
  - Run effect once
    - `$("selector").effect(effectName, optionalArgs)`
      - Effect names are 'blind', 'bounce', 'clip', 'drop', 'explode', 'fold', 'highlight', 'puff', 'pulsate', 'scale', 'shake', 'size', 'slide', 'transfer'.
      - Some effects (e.g., 'size') take additional arguments
  - Make element appear or disappear, or toggle effect
    - `$("selector").show(effectName, optionalArgs)`
    - `$("selector").hide(effectName, optionalArgs)`
    - `$("selector").toggle(effectName, optionalArgs)`
      - Same effect names minus 'highlight'

# Example: HTML

```
<div class="ui-widget-content ui-corner-all effects"
 id="effects-test">
<h3 class="ui-widget-header ui-corner-all effects-header">
 Testing Effects
</h3>
Blah, blah, blah.

Yadda, yadda, yadda.

Foo, bar, baz.
</div>
```

## Testing Effects

Blah, blah, blah.  
Yadda, yadda, yadda.  
Foo, bar, baz.

The HTML shown here just makes the small dialog-like region shown. The buttons that perform the visual effects on that region will be dynamically added by the JavaScript.

57

# Example: JavaScript (Insert Buttons)

```
uiDemo.addEffectButtons = function(selector) {
 var effects = ["blind", "clip", "drop", "explode", "fold",
 "puff", "slide", "scale", "pulsate"];
 for(var i=0; i<effects.length; i++) {
 var effect = effects[i];
 var attributes = {
 type: "button",
 value: effect
 };
 var button = $("<input>", attributes);
 $(selector).append(button);
 }
 $(selector + " :button").click(function(event) {
 uiDemo.toggleEffect(event.currentTarget.value,
 "#effects-test");
 });
};
```

The top part inserts a bunch of buttons that have the labels "blind", "clip", etc.

The bottom part attaches a click handler that calls uiDemo.toggleEffect (next slide) with the button label and a region id.

58

# Example: Main JavaScript

```
uiDemo.toggleEffect = function(buttonLabel, selector) {
 $(selector).toggle(buttonLabel);
};

$(function() {
 uiDemo.addEffectButtons ("#effects");
 ...
});
```

When button is pressed, call \$("#effects-test").toggle("blind"), \$("#effects-test").toggle("clip"), \$("#effects-test").toggle("drop"), etc. The name of the effect is the button label.

59

# Example: Results



60



# Themes

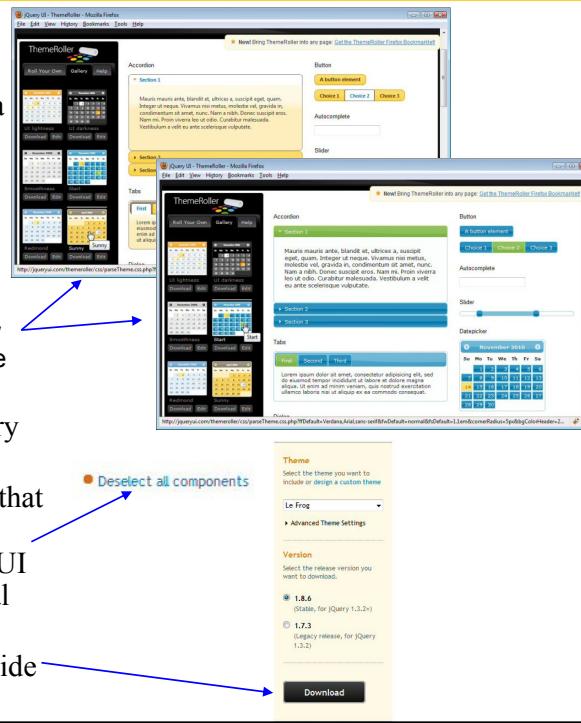
Customized Java EE Training: <http://coursescoreservlets.com/>  
Servlets, JSP, JSF 2.0, Struts, Ajax, GWT 2.0, Spring, Hibernate, SOAP & RESTful Web Services, Java 6.  
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

## Overview

- **Idea**
  - jQuery UI supports themes (sometimes called “skins”)
    - Many prebuilt themes
    - Tool for interactively customizing themes
    - All rich GUI elements consistently use the same CSS style names and image names, so changing the style sheet and associated image folder totally changes the look and feel.
- **Using themes**
  - Can change theme for all jQuery UI elements by switching to new style sheet (with image folder)
  - By using standard CSS names for your *own* HTML markup, can ensure that *your* HTML adapts to the theme
  - Can change theme at runtime by creating and inserting a new style sheet link into the head

# Downloading Themes

- You already have a theme
  - When you download jQuery UI as shown in previous lecture, you get a theme (ui-lightness is default)
- Getting new themes
  - <http://jqueryui.com/themeroller/>
  - Click on “Gallery”
  - Find a theme and click on it
    - Sample elements on right will now take on look and feel of that theme
  - Click “download” under the theme, which will return you to main jQuery UI download page.
  - Click “deselect all components” so that you will download CSS and images only, not the JavaScript for jQuery UI (which you already have from initial download)
  - Click “Download” button on right side



63

# Using Downloaded Themes

- Only the “css” folder is needed
  - Copy the folder inside “css” to your app (e.g., in Eclipse, to the WebContent/css folder)
    - However, you probably rename folder and CSS file first, as described on upcoming slides
  - From your HTML page, load main CSS file that is in the theme folder
    - But again, you will usually load the renamed file
- Note
  - This assumes you already have jQuery and jQuery UI installed as described in previous tutorial.
    - So, you are installing extra themes only

64

# Customizing Themes

- **Three main options**

- Use a prebuilt theme
  - There are many choices, and each theme has consistent use of colors
- Make a variation of an existing theme
  - Choose a theme, then either click the “Edit” button under the theme, or click on the “Roll Your Own” tab. Then, interactively change various settings before downloading
- Edit the raw CSS file that comes with either of the above two options
  - Mostly reserved for CSS experts

- **Firefox bookmarklet**

- Firefox users can get easy access to ThemeRoller via <http://jqueryui.com/themeroller/developertool/>



65

## Planning Ahead to Change Theme

- **Usual approach**

- You change theme by changing files on the Web site. But, it is easier to *replace* many files on a Web site than to *edit* many files on a Web site.

- **Strategies**

- Use generic theme names
  - Rename theme (skin) folder from ui-lightness (or sunny, or blitzer or whatever you chose) to **jquery-ui**
  - Rename CSS file inside theme (skin) folder from jquery-ui-1.8.x.custom.css to **jquery-ui.css**
- Use jQuery UI CSS names for your *own* markup
  - ui-widget-header, ui-widget-content, etc.
  - For details, see <http://jqueryui.com/docs/Theming/API>
- Changing theme
  - Replace the jquery-ui folder with version for new theme

66

# Setting Up Page: Example

```
<head><title>jQuery UI</title>
<link rel="stylesheet"
 href="./css/jquery-ui/jquery-ui.css"
 type="text/css"/>
<link rel="stylesheet"
 href="./css/extra-styles.css"
 type="text/css"/>
<script src="./scripts/jquery.js"
 type="text/javascript"></script>
<script src="./scripts/jquery-ui.js"
 type="text/javascript"></script>
<script src="./scripts/jquery-ui-examples.js"
 type="text/javascript"></script>
</head>
```

Folder was called ui-lightness (one of the theme names) in original download  
File was called jquery-ui-1.8.x.custom.css in original download

My CSS file that styles non-jQuery-UI elements. Limit custom styles so you can adapt to custom themes more easily.

File was called jquery-1.4.x.min.js in original download

File was called jquery-ui-1.8.x.custom.min.js in original download

My JavaScript file that calls jQuery and jQuery UI functions

67

# Changing Theme at Runtime

- **Idea**
  - Have CSS folders for multiple themes on Web site.
  - To change theme, create new style sheet link and append it on the end of the “head” section
    - Although you most commonly change the theme by simply changing the CSS folder as on previous slide, this approach lets the end user change the theme

- **JavaScript for inserting a style sheet**

```
var themeName = "...";
var attributes = {
 type: "text/css",
 rel: "stylesheet",
 href: "css/" + themeName + "/jquery-ui.css"
};
var styleSheetLink = $("<link>", attributes);
$("head").append(styleSheetLink);
```

68

## Example: HTML

```
<div class="ui-widget-content ui-corner-all"
 id="themes-div">
 <input type="radio" name="themes" value="black-tie"/>Black Tie
 <input type="radio" name="themes" value="blitzer"/>Blitzer
 <input type="radio" name="themes" value="le-frog"/>Le Frog
 <input type="radio" name="themes" value="smoothness"/>Black Tie

 <input type="radio" name="themes" value="start"/>Start
 <input type="radio" name="themes" value="trontastic"/>Trontastic
 <input type="radio" name="themes" value="ui-darkness"/>UI-Darkness
 <input type="radio" name="themes" value="jquery-ui"/>UI Lightness (Default)
</div>
```

69

## Example: JavaScript

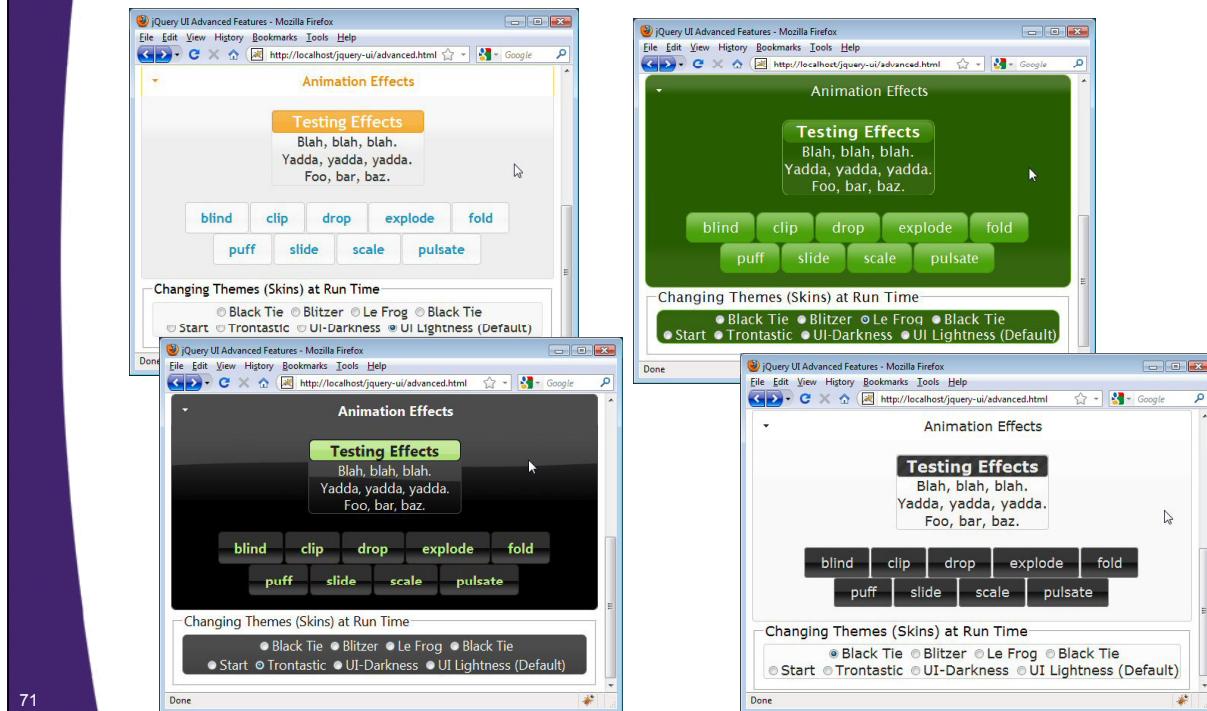
```
uiDemo.switchToTheme = function(themeName) {
 var attributes = {
 type: "text/css",
 rel: "stylesheet",
 href: "css/" + themeName + "/jquery-ui.css"
 };
 var styleSheetLink = $("<link>", attributes);
 $("head").append(styleSheetLink);
};

uiDemo.switchToSelectedTheme = function() {
 var theme = $(this).val();
 uiDemo.switchToTheme(theme);
};

$(function() { ...
 $("#themes-div input").click(uiDemo.switchToSelectedTheme); })
```

70

# Example: Results



71

© 2010 Marty Hall



## Wrap-up

# Summary

- **Autocompleter**
  - `$("#field-id").autocomplete({ source: arrayOfChoices });`
  - `$("#field-id").autocomplete({ source: "urlReturningArray" });`
  - `$("#field-id").autocomplete({ source: matcherFunction });`
- **Animation effects**
  - `($("selector").effect("highlight");`
  - `($("selector").toggle("anotherEffectName");`
- **Themes**
  - Rename the theme folder and CSS file inside
  - Use jQuery-UI CSS names in your own HTML
  - Change theme by replacing the theme folder

73

© 2010 Marty Hall



# Questions?