

# jQuery Ajax Support: Part 4 – Promises and Deferred Objects

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.



For customized training related to JavaScript or Java, email [hall@coreservlets.com](mailto:hall@coreservlets.com)  
Marty is also available for consulting and development support

Taught by lead author of *Core Servlets & JSP*,  
co-author of *Core JSF* (4<sup>th</sup> Ed), and this tutorial.

Available at public venues, or  
custom versions can be held on-site at your organization.

• **Courses developed and taught by Marty Hall**

- JavaScript, jQuery, Ext JS, JSF 2.3, PrimeFaces, Java 8 programming, Spring Framework, Spring MVC, Android, GWT, custom mix of topics
- Courses available in any state or country.

- Maryland/DC companies can also choose afternoon/evening courses.

• **Courses developed and taught by coreservlets.com experts (edited by Marty)**

- Hadoop, Hibernate/JPA, HTML5, RESTful Web Services

Contact [hall@coreservlets.com](mailto:hall@coreservlets.com) for details



## Topics in This Section

- **Using Promises for Ajax**
  - To simplify adding additional handlers later
- **Combining Promises**
  - To test if multiple calls have finished
- **Using Deferred Objects and Promises in Your Own Code**
  - To manage your own delayed or asynchronous tasks

4

coreservlets.com – custom onsite training



# Using Promises and Deferred Objects for Ajax

He loses his thanks who promises and delays.

-- Old proverb that we are about to disprove

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

## Motivation

- **Problem**
  - Standard jQuery supported only a single callback handler
    - So, if you wanted to add more functionality, you had to modify the code of the original handler, making refactoring hard and breaking things if that handler was shared
- **Solution**
  - Add handlers separately using Deferred and Promise API
- **Extra benefit**
  - Familiar to users of other frameworks, since this API (CommonJS Promises/A Interface) is used elsewhere

6

## Big Idea

- **Replace this:**

```
$.ajax({ url: remoteAddress,  
        success: successHandler,  
        complete: completeHandler,  
        error: errorHandler });
```
- **With this:**

```
var req = $.ajax({ url: remoteAddress });  
req.done(successHandler);  
req.always(completeHandler);  
req.fail(errorHandler);
```
- **But now:**
  - If you later want to add an additional success handler:

```
req.done(additionalSuccessHandler);
```

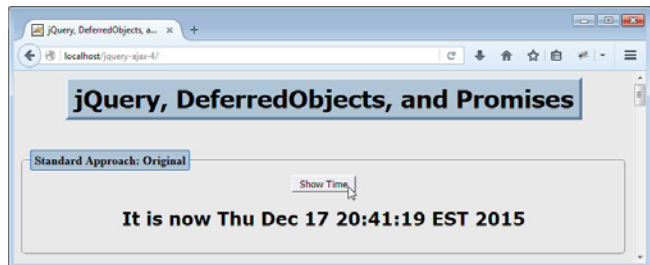
7

## Example: Displaying Remote Data

- **Standard approach**

```
$.ajax({  
  url: address,  
  success:  
    function(text) {  
      insertText(text, resultRegion);  
    }  
});
```

This is the code that was shown in the earlier sections, and it works fine in most cases.



8

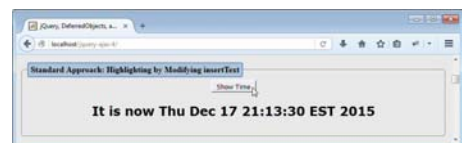
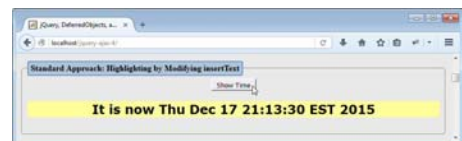
## Refactoring: Highlighting Result

- **Option 1: modify insertText**

```
function insertText2(text, resultRegion) {  
  $(resultRegion).html(text);  
  highlight(resultRegion);  
}
```

- **Option 2: modify main \$.ajax call**

```
$.ajax({  
  url: address,  
  success:  
    function(text) {  
      insertText(text, resultRegion);  
      highlight(resultRegion);  
    }  
});
```



The highlight function uses jQuery UI to briefly flash background.

```
function highlight(selector) {  
  $(selector).hide().show("highlight");  
}
```

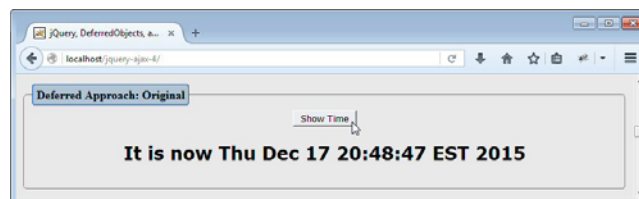
9

## Example: Displaying Remote Data

- **Deferred approach**

```
var req = $.ajax({ url: address });  
req.done(function(text) { insertText(text, resultRegion); });
```

Works the same as the first example.



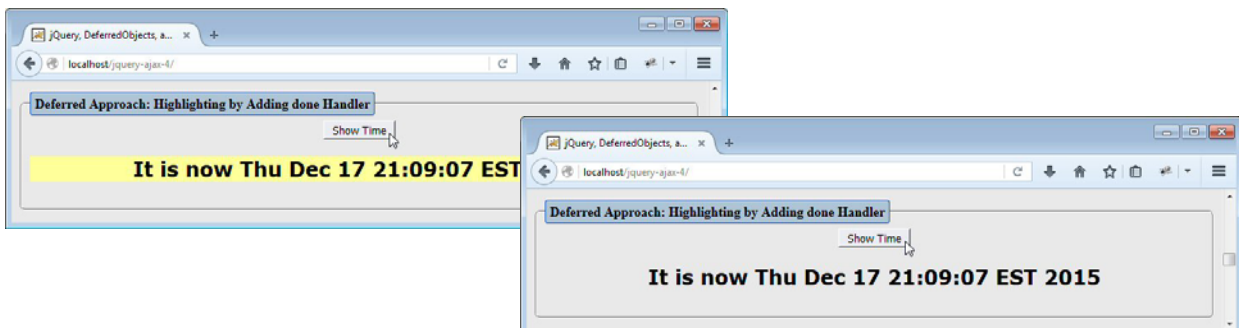
10

## Refactoring: Highlighting Result

- **Easier**

- An additional line instead of changing old code

```
var req = $.ajax({ url: address });  
req.done(function(text) { insertText(text, resultRegion); });  
req.done(function(text) { highlight(resultRegion); });
```



11

# Displaying Temporary Messages During Ajax Requests

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

## Overview

- **Task done in earlier sections: show message while waiting**
  - Display info in region that has “display: none”, before sending Ajax request, change region to use “display: inline”, when request finishes, hide the region again
- **Comparing the two approaches (next slides)**
  - They are about the same: no clear advantage to either approach
  - Conclusions in principle
    - No need to redo existing legacy code
      - In most situations, deferred approach has no clear advantages
    - Use deferred approach for new applications
      - It is superior in some situations (highlighting example and upcoming examples with \$.when)
      - It is familiar to users of some other libraries
      - It is (arguably) no longer or harder to understand even when it is not better
  - Observations in practice
    - Most jQuery developers still use the standard approach

## Shared HTML

```
<fieldset>
<legend>...: Showing Temporary Messages</legend>
  <input type="button" value="Show Time"
    id="..." />
  <h2 id="..." style="display: none">
    
    Waiting for server...
  </h2>
  <h2 id="..."></h2>
</fieldset>
```

14

## Shared Server Code

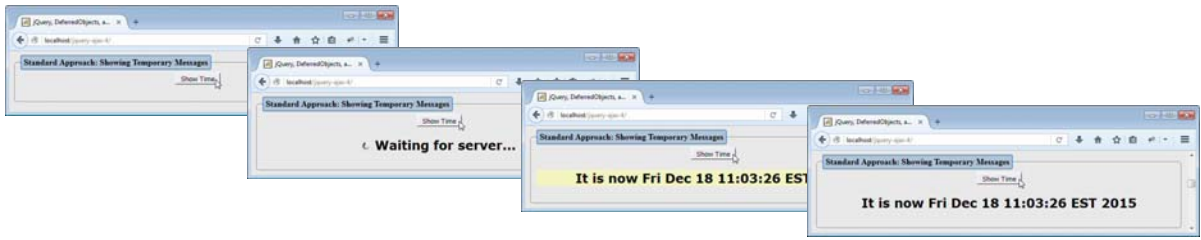
```
<% try { Thread.sleep(5000); } catch(Exception e) {} %>
It is now <%= new java.util.Date() %>
```

Waits 5 seconds and then outputs the current date and time

15

## Standard Approach

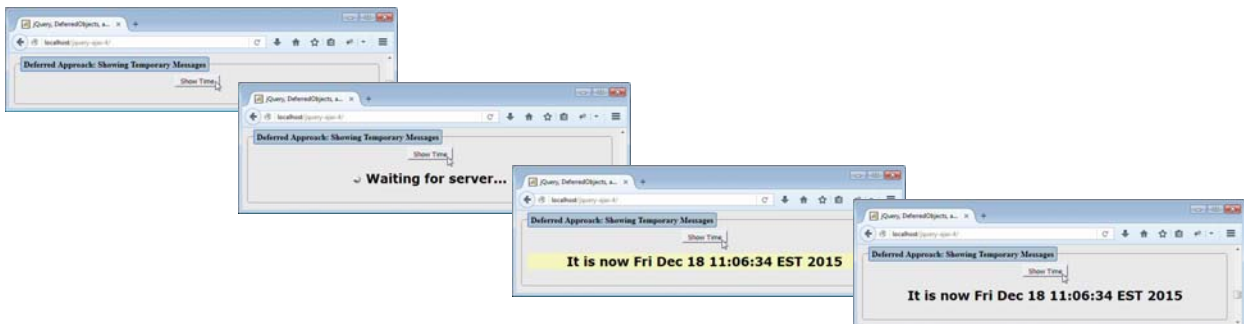
```
$(resultRegion).html(""); // Erase any previous results
$(workingRegion).show();
$.ajax({
  url: address,
  success: function(text) {
    insertText(text, resultRegion);
    highlight(resultRegion);
  },
  complete: function(text) { $(workingRegion).hide(); }
});
```



16

## Deferred Approach

```
$(resultRegion).html(""); // Erase any previous results
$(workingRegion).show();
var req = $.ajax({ url: address });
req.done(function(text) { insertText(text, resultRegion); });
req.done(function(text) { highlight(resultRegion); });
req.complete(function(text) { $(workingRegion).hide(); });
```



17



# \$.when: Checking Multiple Promises

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

## Big Idea

- **You can run code after *all* asynch calls**
  - For when you don't know the order in which tasks will complete, but you want to do something only after *all* have finished

- **Syntax**

```
var request1 = $.ajax({ url: "..." });
request1.done(individualSuccessHandler1);
var request2 = $.ajax({ url: "..." });
request2.done(individualSuccessHandler2);
...
$.when(request1, request2, ...)
    .done(compositeSuccessHandler)
    .always(compositeCompleteHandler)
    .fail(compositeFailureHandler);
```

## Example: Temporary Message

- **Fire off two calls to server**
  - Each is slow
  - Each will result in data inserted into page
  - Order of return is not known
- **Display “working...” message**
  - Exactly as shown earlier
- **Hide “working” message**
  - Only when *both* results are finished

20

## Temporary Data: JavaScript

```
function showTimeAndNumber() {
    var address1 = "show-time-slow.jsp";
    var resultRegion1 = "#deferred-when-time-result";
    var address2 = "show-number-slow.jsp";
    var resultRegion2 = "#deferred-when-number-result";
    var workingRegion = "#deferred-when-working";
    $(resultRegion1).html("");
    $(resultRegion2).html("");
    $(workingRegion).show();
    var req1 = $.ajax({ url: address1 });
    req1.done(function(text) { insertText(text, resultRegion1); });
    req1.done(function(text) { highlight(resultRegion1); });
    var req2 = $.ajax({ url: address2 });
    req2.done(function(text) { insertText(text, resultRegion2); });
    req2.done(function(text) { highlight(resultRegion2); });
    $.when(req1, req2).always(function(text) { $(workingRegion).hide(); });
}
```

21

## Temporary Data: HTML

```
<fieldset>
<legend>Deferred Approach: Using $.when</legend>
  <input type="button" value="Show Time and Number"
        id="deferred-when-button"/>
  <h2 id="deferred-when-working" style="display: none">
    
    Waiting for server...
  </h2>
  <h2><span id="deferred-when-time-result"></span><br/>
    <span id="deferred-when-number-result"></span></h2>
</fieldset>
```

22

## Temporary Data: JSP

- **show-time-slow.jsp**

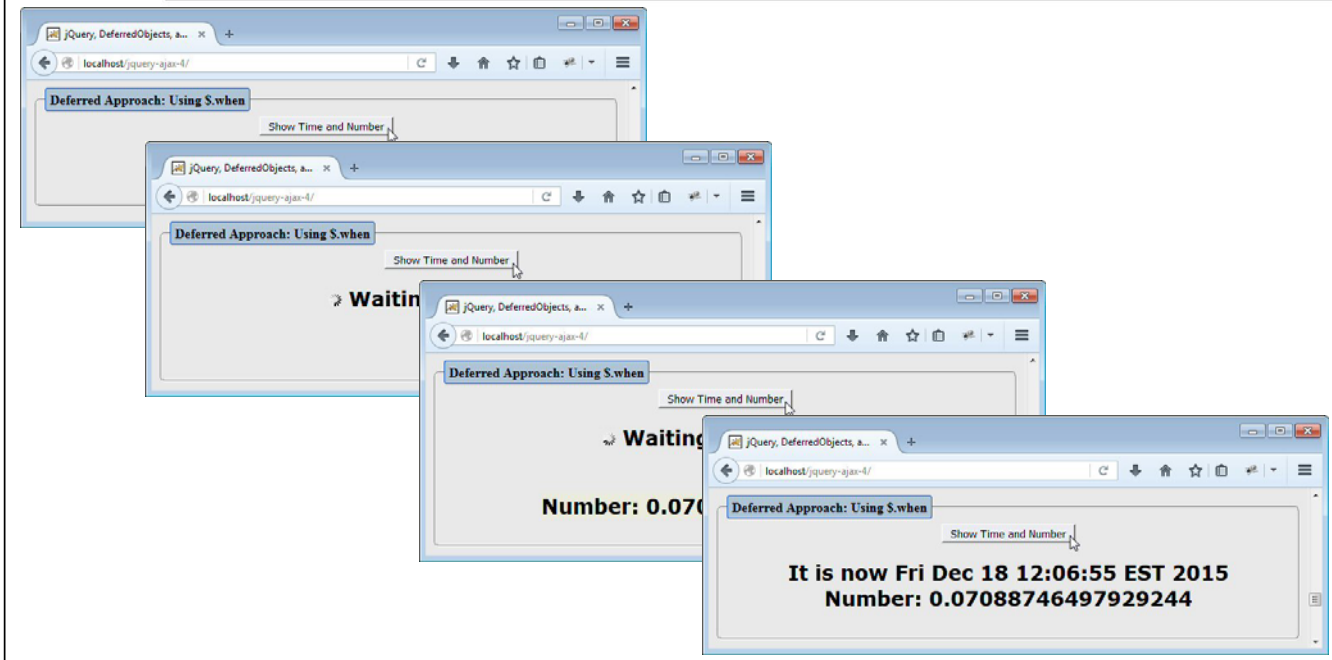
```
<% try { Thread.sleep(5000); } catch(Exception e) {} %>
It is now <%= new java.util.Date() %>
```

- **show-number-slow.jsp**

```
<% try{ Thread.sleep(4000); } catch(Exception e) {} %>
Number: <%= Math.random() %>
```

23

# Temporary Data: Results



coreservlets.com – custom onsite training



# Making Your Own Deferred Objects

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

## Big Idea

- **Create a Deferred Object**  
`var dfd = $.Deferred();`
- **Do a long-running or asynchronous task**
  - Call `dfd.reject()` if it fails
  - Call `dfd.resolve()` if it succeeds
- **Return a Promise**  
`return(dfd.promise());`
- **This return value can be used like “req” before**
  - Can use `.done`, `.always`, `.fail`, `$.when`, etc.

26

## Example

- **Do some slow-but-local task**
  - Use `fadeIn` to slowly make result appear
    - Note that `fadeIn` takes a function as a second argument, so you can supply `dfd.resolve` to it instead of calling `dfd.resolve()` explicitly
  - Return a Promise
- **Display temporary message**
  - Once `slow tasks` returns a Promise, this code will be almost exactly the same as in the last Ajax example

27

## Local Tasks: JavaScript

```
function showNum(resultRegion, millis) {
    var dfd = $.Deferred();
    $(resultRegion).html("Number: " + Math.random());
    $(resultRegion).fadeIn(millis, dfd.resolve);
    return(dfd.promise());
}

function showNumbers() {
    var workingRegion = "#deferred-custom-working";
    var resultRegion1 = "#deferred-custom-number-result-1";
    var resultRegion2 = "#deferred-custom-number-result-2";
    $(resultRegion1).hide();
    $(resultRegion2).hide();
    $(workingRegion).show();
    $.when(showNum(resultRegion1, 6000), showNum(resultRegion2, 3000))
        .done(function(text) { $(workingRegion).hide(); });
}
```

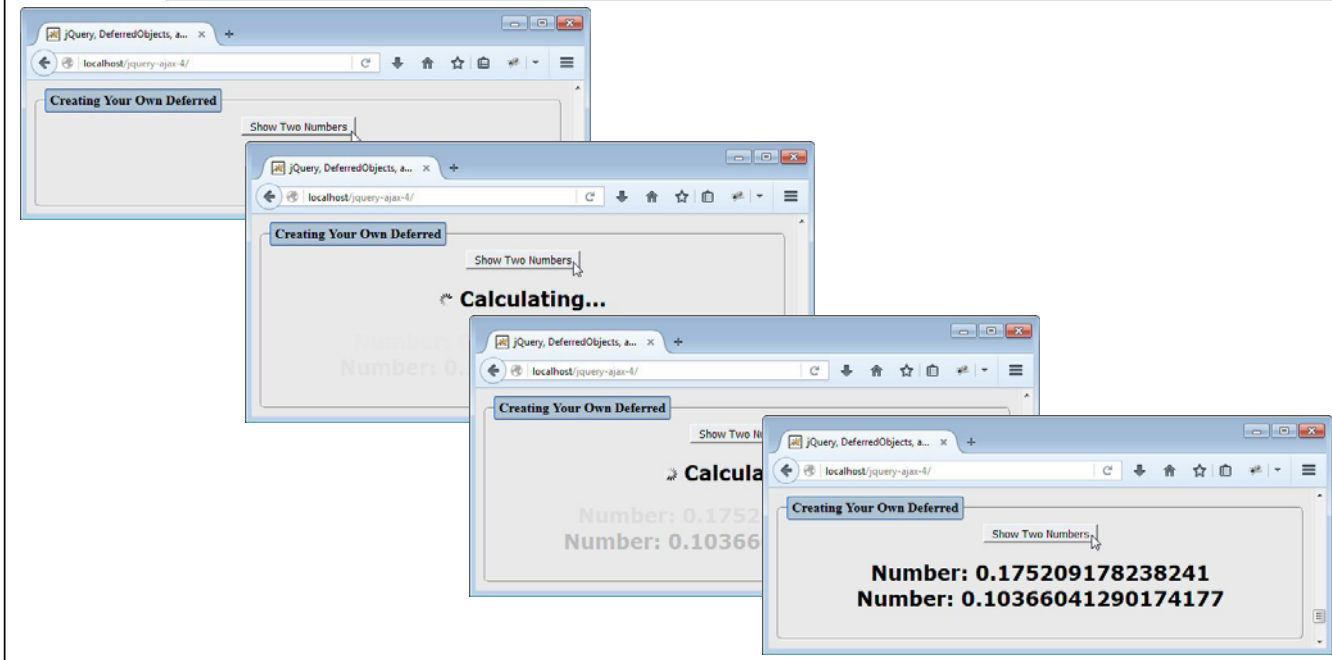
28

## Local Tasks: HTML

```
<fieldset>
<legend>Creating Your Own Deferred</legend>
  <input type="button" value="Show Two Numbers"
    id="deferred-custom-button"/>
  <h2 id="deferred-custom-working" style="display: none">
    
    Calculating...
  </h2>
  <h2><span id="deferred-custom-number-result-1" style="display: none">
    </span><br/>
    <span id="deferred-custom-number-result-2" style="display: none">
    </span>
  </h2>
</fieldset>
```

29

# Local Tasks: Results



coreservlets.com – custom onsite training



# Wrap-Up

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.

# Summary

- **Basics**

```
$.ajax({ url: remoteAddress,  
        success: successHandler,  
        complete: completeHandler,  
        error: errorHandler });
```

```
var req = $.ajax({ url: remoteAddress });  
req.done(successHandler);  
req.always(completeHandler);  
req.fail(errorHandler);
```

- **Checking multiple Promises**

- \$.when(req1, req2).always(...);

- **Managing your own tasks**

- Create Deferred, call resolve(), return promise()

32

coreservlets.com – custom onsite training



# Questions?

More info:

<http://www.coreservlets.com/javascript-jquery-tutorial/> – Tutorial on JavaScript, jQuery, and jQuery UI

<http://courses.coreservlets.com/Course-Materials/java.html> – General Java programming tutorial

<http://www.coreservlets.com/java-8-tutorial/> – Java 8 tutorial

<http://courses.coreservlets.com/java-training.html> – Customized Java training courses, at public venues or onsite at your organization

<http://coreservlets.com/> – JSF 2, PrimeFaces, Java 8, JavaScript, jQuery, Ext JS, Hadoop, RESTful Web Services, Android, HTML5, Spring, Hibernate, Servlets, JSP, GWT, and other Java EE training

Many additional free tutorials at coreservlets.com (JSF, Android, Ajax, Hadoop, and lots more)

Slides © 2016 Marty Hall, hall@coreservlets.com

For additional materials, please see <http://www.coreservlets.com/>. The JavaScript tutorial section contains complete source code for all examples in the entire tutorial series, plus exercises and exercise solutions for each topic.