

# jQuery Ajax Support: Sending Data to the Server

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

- **Reading and escaping textfield values**
- **Sending data with the data property of \$.ajax**
  - Explicit string
  - Data object
  - String built by the serialize function
- **Sending data using the load function**

4

coreservlets.com – custom onsite training



# Sending Data to the Server with \$.ajax: Summary

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

- **Most server programs expect input data**

- Usually in the form `name1=val1&name2=val2`
  - Most server programs have support for extracting the values

- **`$.ajax({ url: ..., success: ..., data: ... });`**

- Can be a String, in which case it is sent unchanged.
  - On end of URL or in POST data, depending on HTTP type
  - String can be built automatically using “serialize” function
- Can be an object, in which case query string gets built out of the property names and the URL-encoded property values

- **Equivalent examples**

```
$.ajax({... data: "param1=foo+bar%21&param2=baz"});
```

```
$.ajax({... data: { param1: "foo bar!", param2: "baz" }});
```

6

## Three Alternatives for the data Property

- **Explicit string**

- Use an explicit string for data property

```
$.ajax({url: ..., data: "a=foo&b=bar", success: ...});
```

- **Data object**

- Use an object for data property

```
var params = { a: "value 1", b: "another value!"};
$.ajax({url: ..., data: params, success: ...});
```

- **String built by the serialize function**

- Build parameter string from form, use that string for the data property

```
$.ajax({url: ..., data: $("#form-id").serialize(), success: ...});
```

7

## Quick Aside: the val and escape Functions

- **Idea**

- val() returns the value of an input element, most commonly a textfield
  - val("some text") sets the value instead of reading it
- Textfield values can contain spaces, ampersands, and other characters that have special meaning in URLs. So, you must use escape to URL-encode the value.

- **Example**

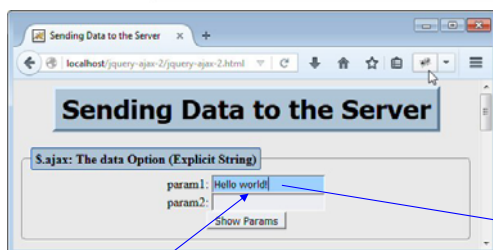
```
var rawText = $("#some-textfield-id").val();  
var escapedText = escape(rawText);
```

- **Notes on val**

- Exception to the rule that functions on sets of elements return the set
  - Most functions return the set so you can do chaining, but val returns a string
- Works for all input elements, even multiselectable select elements (in which case it returns an array)

8

## Practicing with val and escape



`<input type="text" id="field1"/>`

```
> escape("Hello");  
"Hello"  
> escape("Hello world!");  
"Hello%20world%21"  
> $("#field1");  
Object[ input#field1 property value = "Hello world!" attribute value = "null" ]  
> $("#field1").val();  
"Hello world!"  
> escape($("#field1").val());  
"Hello%20world%21"  
> $("#field1").val("New field value");  
Object[ input#field1 property value = "New field value" attribute value = "null" ]
```



# Using an Explicit String

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

- **Explicit string**

- Supply an explicit string for data property, and it will be sent to server exactly as is

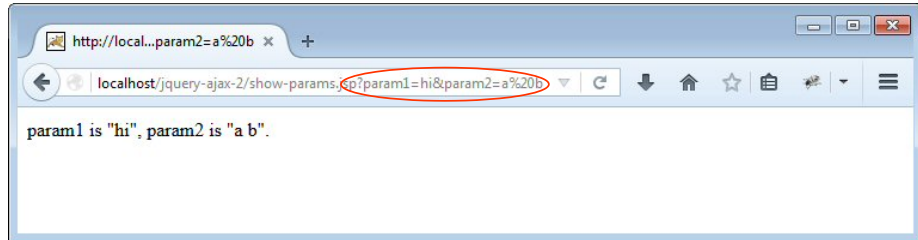
```
$.ajax({url: ..., data: "a=foo&b=bar", success: ...});
```

- **Usage**

- You can send a hardcoded string for practice, but in real applications you usually build the string from the values of textfields or other input elements
- You must escape the textfield values to URL-encode special values like spaces

## The data Option with Explicit String: JSP

```
param1 is ${param.param1},  
param2 is ${param.param2}.
```



In JSP, `${param.someParamName}` returns the decoded value of the specified request parameter, or empty string if that parameter name does not exist. For example, if the JSP is **Parameter is `${param.foo}`** and you access the page with `http://host/app/page.jsp?foo=test`, you get "Parameter is test"

All three examples with the data option use this same JSP page, `show-params.jsp`.

12

## The data Option with Explicit String: JavaScript

```
$(function() {  
    $("#data-button-1").click(showParams1);  
    ...  
});  
  
function showParams1() {  
    var queryString =  
        "param1=" + escape($("#field1").val()) +  
        "&param2=" + escape($("#field2").val());  
    $.ajax({ url: "show-params.jsp",  
            data: queryString,  
            success: showAlert });  
}
```

Note that `showAlert` is the same function used in earlier examples.

13

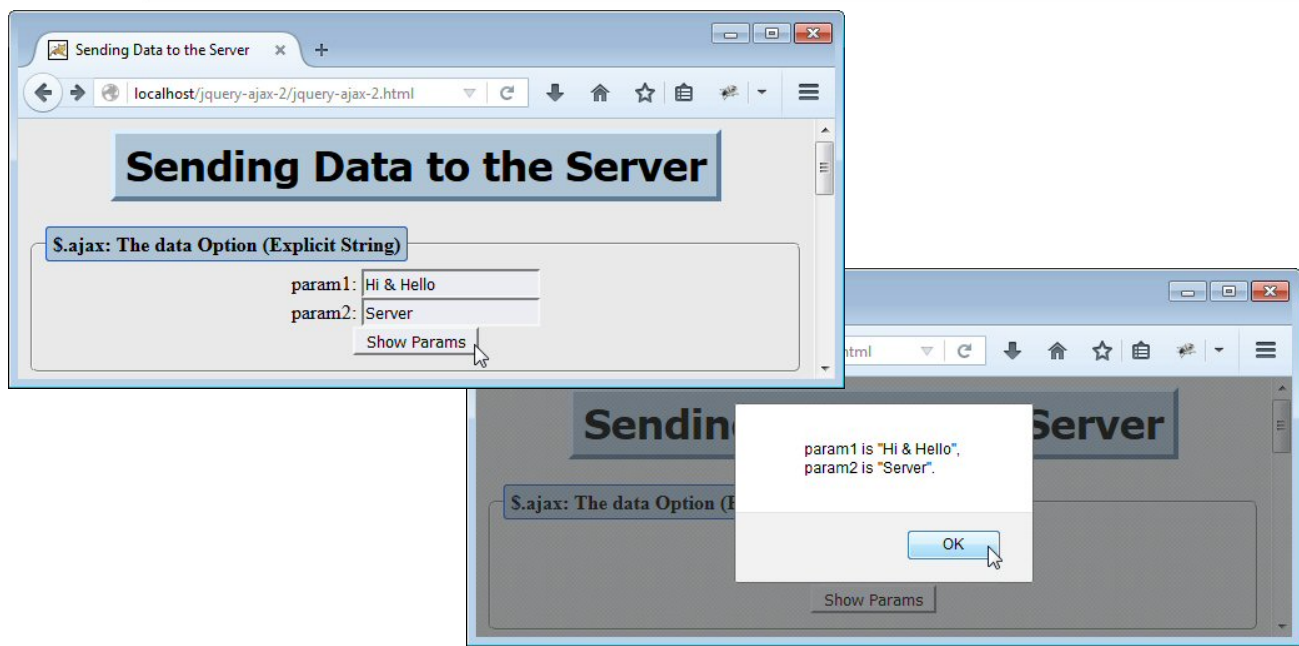
## The data Option with Explicit String: HTML

...

```
<fieldset>
<legend>$.ajax: The data Option (Fixed String)</legend>
  param1:
  <input type="text" id="field1"/><br/>
  param2:
  <input type="text" id="field2"/><br/>
  <input type="button" value="Show Params"
    id="data-button-1"/>
</fieldset>
```

14

## The data Option with Explicit String: Results



# Using a Data Object

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

- **Data object**

- Supply an object for data property
- Property names become param names and URL-encoded property values become param values
- *URL-encoding of values is automatic*

```
var params = { a: "value 1", b: "another value!"};  
$.ajax({url: ..., data: params, success: ...});
```

- **Usage**

- The values usually come from textfields or other input elements
- You do *not* need to escape (URL-encode) the textfield values



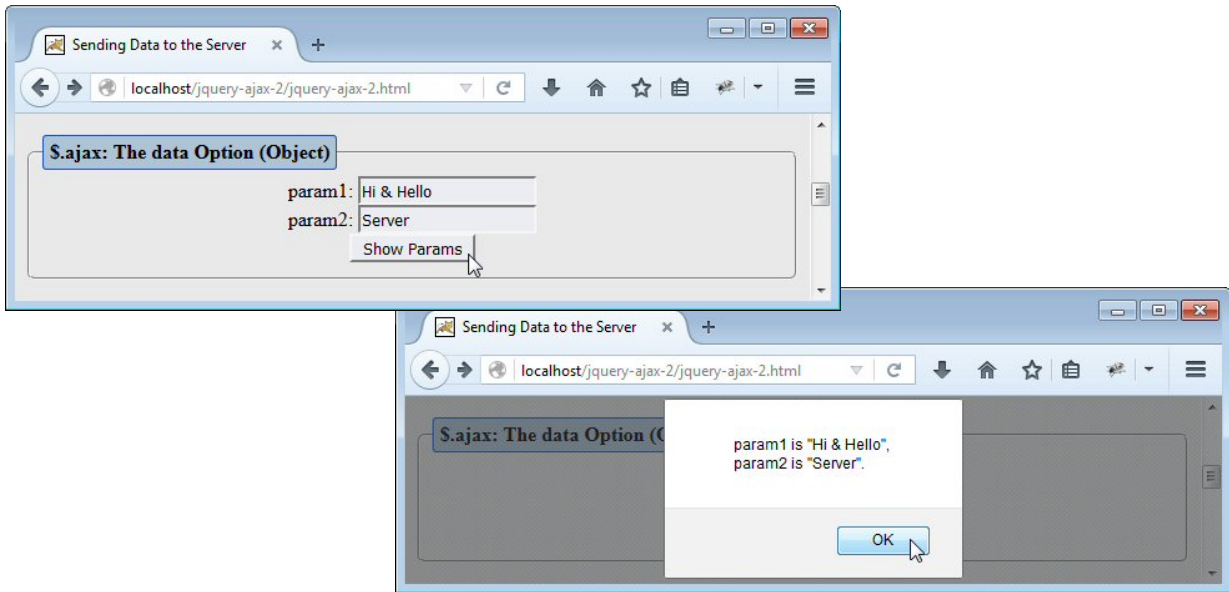
## The data Option with Data Object: JavaScript

```
$(function() {  
    $("#data-button-2").click(showParams2);  
    ...  
});  
  
function showParams2() {  
    var params =  
        { param1: $("#field3").val(),  
          param2: $("#field4").val() };  
    $.ajax({ url: "show-params.jsp",  
            data: params,  
            success: showAlert });  
}
```

## The data Option with Data Object: HTML

```
...  
<fieldset>  
    <legend>$.ajax: The data Option (Object)</legend>  
    param1:  
    <input type="text" id="field3"/><br/>  
    param2:  
    <input type="text" id="field4"/><br/>  
    <input type="button" value="Show Params"  
        id="data-button-2"/>  
</fieldset>
```

## The data Option with Data Object: Results



coreservlets.com – custom onsite training



# Using the serialize Function

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

- **String built by serialize**

- Give id to the form. Give *names* (not ids) to input elements. When you call `serialize` on form, it builds the same query string as a browser would on normal form submission, including URL-encoding all the field values.

```
$.ajax({url: ..., data: $("#form-id").serialize(), success: ...});
```

- **Usage**

- You build parameter string for entire form all at once
- Input elements must now have names
- You do not need to explicitly read the textfield values, so no need to use `escape` for URL-encoding

22

## The data Option with String from serialize: JavaScript

```
$(function() {  
    $("#data-button-3").click(showParams3);  
    ...  
});  
  
function showParams3() {  
    $.ajax({ url: "show-params.jsp",  
            data: $("#data-form").serialize(),  
            success: showAlert });  
}
```

23

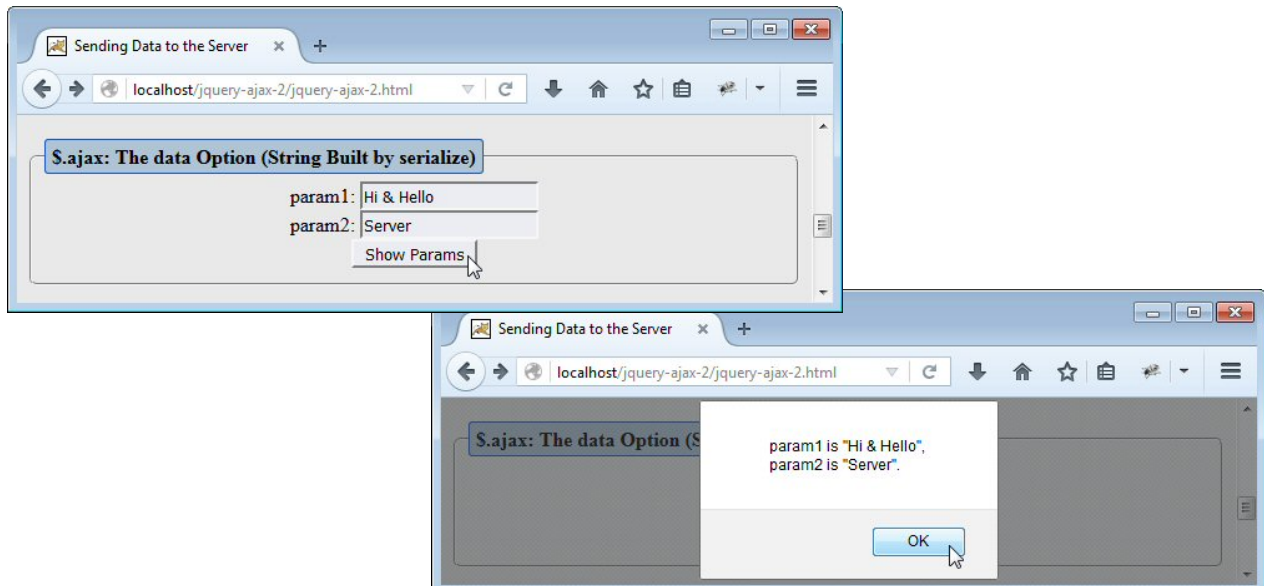
## The data Option with String from serialize: HTML

...

```
<fieldset>
  <legend>$.ajax: The data Option
    (String Built by serialize)</legend>
  <form id="data-form">
    param1:
    <input type="text" name="param1" /><br />
    param2:
    <input type="text" name="param2" /><br />
    <input type="button" value="Show Params"
      id="data-button-3" />
  </form>
</fieldset>
```

24

## The data Option with String from serialize: Results



# Sending Data when Using the load Function

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

- **load greatly simplifies simple content-centric Ajax**
  - It automatically inserts results into the region specified  
`$("#result-area-id").load("url");`
- **But, there is no options object as with \$.ajax**
  - So, you simply supply a string as second argument to load, and this is sent as the parameter string  
`$("#result-area-id").load("url", "name1=val1&name2=val2");`
  - You can build the string however you want, but using serialize is simplest  
`$("#result-area-id").load("url", $("#form-id").serialize());`

## load Example: JavaScript

```
$(function() {  
    $("#load-button").click(insertParams);  
    ...  
});  
  
function insertParams() {  
    $("#params-result").load("show-params.jsp",  
        $("#load-form").serialize());  
}
```

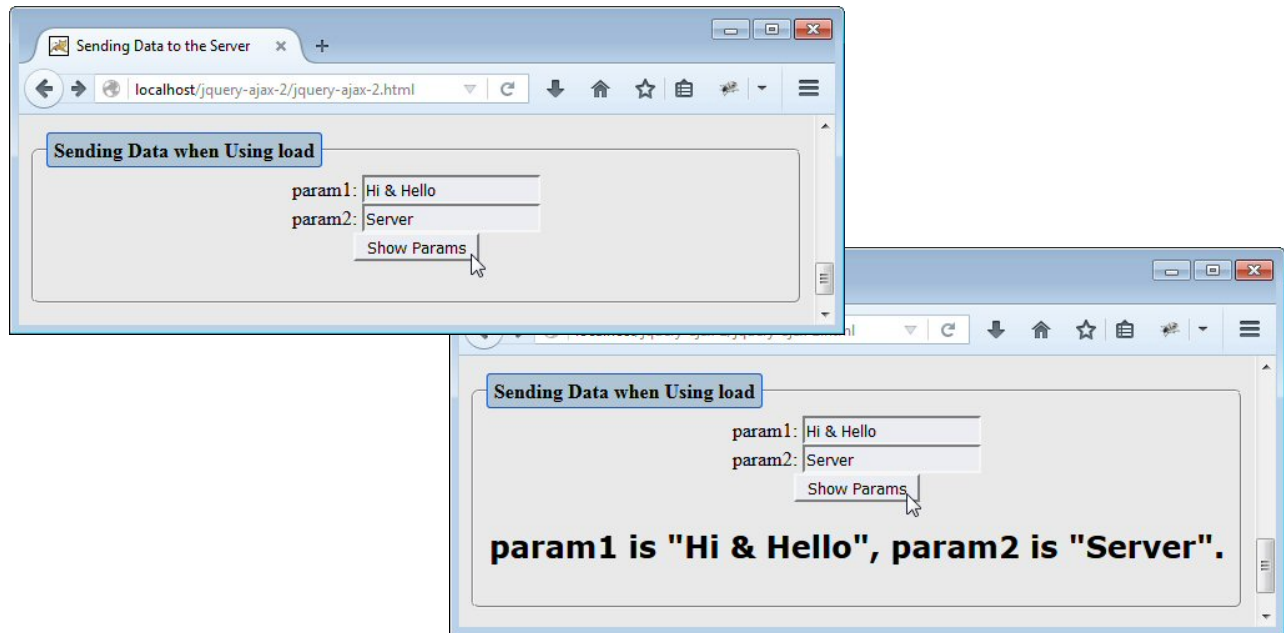
28

## load Example: HTML

```
...  
<fieldset>  
<legend>Sending Data when Using load</legend>  
  <form id="load-form">  
    param1:  
    <input type="text" name="param1"/><br/>  
    param2:  
    <input type="text" name="param2"/><br/>  
    <input type="button" value="Show Params"  
      id="load-button"/>  
    <h2 id="params-result"></h2>  
  </form>  
</fieldset>
```

29

## load Example: 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

- **Sending data with \$.ajax**

```
$.ajax({ url: "relative-address",  
        success: handlerFunction,  
        data: "name1=val1&name2=val2" });
```

```
$.ajax({ url: "relative-address",  
        success: handlerFunction,  
        data: { name1: "val1", name2: "val2" } });
```

```
$.ajax({ url: "relative-address",  
        success: handlerFunction,  
        data: $("#form-id").serialize() });
```

- **Sending data with load**

```
$("#result-id").load("relative-address",  
                    $("#form-id").serialize());
```

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.