JSON versus JSONP Tutorial - Live

JSONP is a simple way to overcome browser restrictions when sending JSON responses from different domains from the client.

But the practical implementation of the approach involves subtle differences that are often not explained clearly.

Here is a simple tutorial that shows JSON and JSONP side by side.

All the code is freely available at [Github](https://github.com/craic/json_jsonp_tutorial) and a live version can be found at [http://json-jsonp-tutorial.craic.com](http://json-jsonp-tutorial.craic.com/)

[JSON (Javascript Object Notation)](http://en.wikipedia.org/wiki/JSON) is a convenient way to transport data between applications, especially when the destination is a Javascript application.

[JQuery](http://jquery.com/) has functions that make Ajax/HTTPD calls from a script to a server very easy and [$.getJSON()](http://api.jquery.com/jQuery.getJSON/) is a great shorthand function for fetching a server response in JSON.

But this simple approach fails if the page making the ajax call is in a different domain from the server. The [Same Origin Policy](http://en.wikipedia.org/wiki/Same_origin_policy) prohibits these cross-domain calls in some browsers as a security measure.

At the time of writing, Google Chrome version 24 and Mozilla Firefox version 17 do not appear to apply this restriction but Internet Explorer version 9 does.

The security implications of allowing cross domain requests should be considered carefully in your application but if you do want to allow them then you need a way to overcome the browser restrictions.

[JSONP (JSON with Padding)](http://en.wikipedia.org/wiki/JSONP) makes this possible in all browsers.

JSONP wraps up a JSON response into a JavaScript function and sends that back as a Script to the browser. A script is not subject to the Same Origin Policy and when loaded into the client, the function acts just like the JSON object that it contains.

This page shows the two approaches with snippets of Server and Client code. Look at the source of this page to see the full JS code.

The code consists of a Server application, implemented here as a Ruby [Sinatra](http://www.sinatrarb.com/) application. You should be familiar with Ruby, Gems and basic Sinatra apps, if you want to run a live demo.

You can run it locally with the commands 'bundle install' and 'rackup -p 4567' and then going to 'http://localhost:4567' with your browser.

NOTE: To really demonstrate the cross domain operation you need to put the client and server code on TWO DIFFERENT machines. Run this page on your localhost and change the 'host' in the JS to point to 'http://json-jsonp-tutorial.craic.com'.

The examples make use of the [JSON.stringify](https://github.com/douglascrockford/JSON-js/blob/master/json2.js) method from Douglas Crockford to 'pretty print' the returned JSON. You can find that [HERE](https://github.com/douglascrockford/JSON-js/blob/master/json2.js).

**JSON**

Here is a minimal example that uses JSON as the transport for the server response. The client makes an ajax request with the JQuery shorthand function $.getJSON. The server generates a hash, formats it as JSON and returns this to the client. The client formats this and puts it in a page element.

Server:

get '/json' do

content\_type :json

content = { :response => 'Sent via JSON',

:timestamp => Time.now,

:random => rand(10000) }

content.to\_json

end

Client: