It turns out there are some ridiculously picky little requirements that browsers have in order to send JSON to an origin other than the one the page came from. I had already put most of the requested headers into the server for “CORS” capability, but that is only the beginning. The browser (Mozilla, but I understand it is following some strange standard) requires several things to be set in very specific way, and requires that the JSON be labeled as “text/plain” in order to allow it to be transmitted. And the browser gives terribly inaccurate error messages, while refusing to deliver the requests or the responses. To top things off, it “pre-caches” some decisions, and holds on to those, and ignores to see changes that are made recently. Be aware, this is very frustrating.

I believe I have it all working now.

**Access this page:**

<http://leaves.interstagebpm.com/ssofiClient/jsLogin.htm>

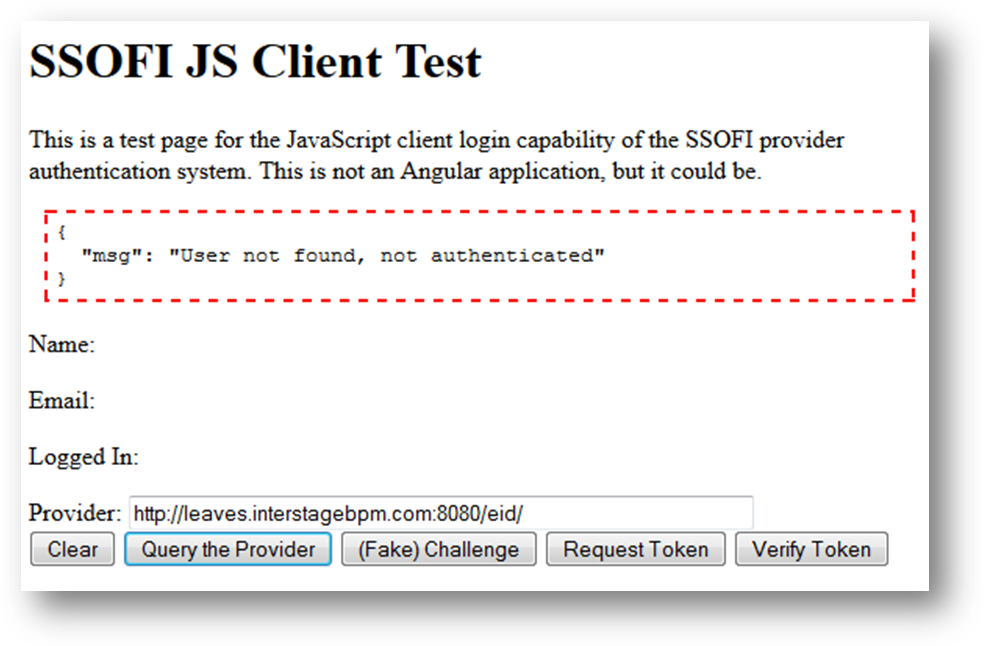
This is a simple little JS test page. View the source to see what it does.

There is an input box for the provider, and the default one is a working version of SSOFI provider:

<http://leaves.interstagebpm.com:8080/eid/>

You will need to log in there, and you might need to register a password for that.

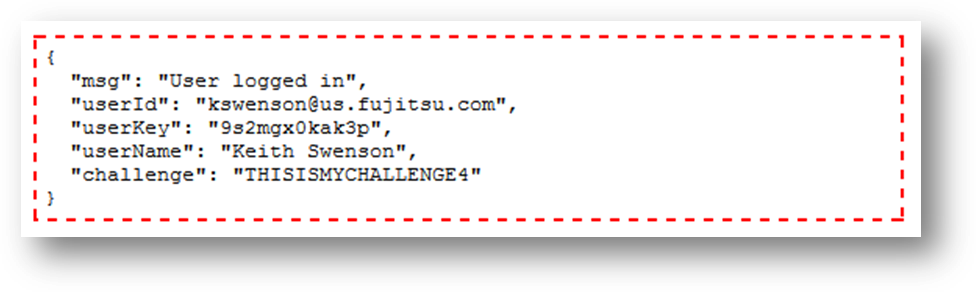
1. Try it without login: Click the query the provider button. The red box shows the actual JSON result from the SSOFI provider.



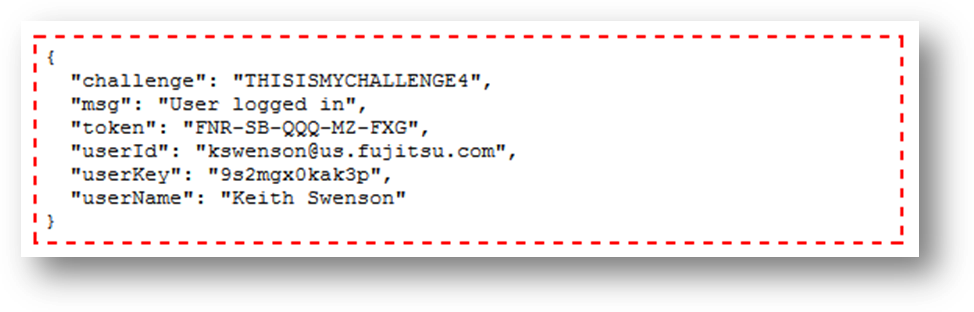
It tells you that you are not logged in. Then go to the provider and log in. Try again:



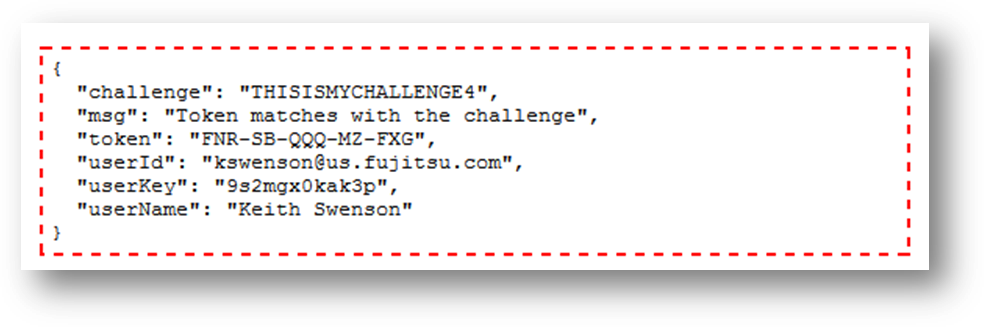
Because I am logged in, it picked up my information. The code in the browser can trust this and believe this. But convincing the server is harder. This would normally start with requesting a challenge from the server, but I have dummied up a fake generation of a challenge value. Clicking the (Fake) Challenge will add a challenge token to the record:



The challenge is just a random value that is unique. Now click the “Request Token” button. This will send a POST request to the SSOFI provider, asking it to add a token, and here is what you get back:

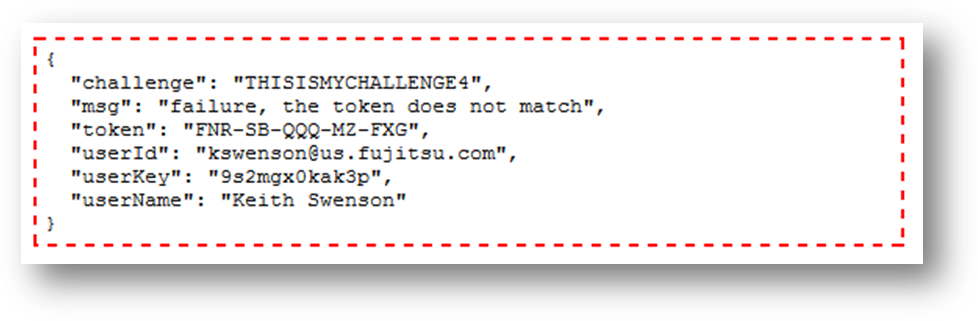


Finally, with both the challenge and the token, you can press the “Verify Token” button the first time:



It says that the token matches (and the response code was 200. The VERIFY will be done by the server, and it is this operation that guarantees to the server that you are really logged in. I do it here in the browser just for demonstration but we need a server side with two calls: one for generate challenge, and one for verify token. This is not yet testable.

Press the verify a SECOND time, and it will fail (with a 400 return code). You can only verify a challenge/token ONCE:



The CLOUD-BASED SSOFI provider is not yet updated to the latest code, and it will not work with this testing page. For now, use this in-house SSOFI provider for testing.

-Keith

Keith D Swenson

cell: 408-859-1005