Article
Istvan Hahn · Dec 7, 2016 3m read

Enable CORS for CSP/ ZEN Applications

As web application gets more complex, more technologies are involved into the application development. Once it gets deployed in large scale the configuration gets more complex too. For sure one of the most difficult part of the story is the security. In a complex solution when independent servers are feeding single web pages with contents, it is indeed challenging to keep the integrity of such system. HTML5 introduced a (weak) security constraint, the Cross Origin Resource Sharing (CORS). This article tells how to enable CORS for CSP/ ZEN applications.

The use case

Assume the following. We have a web page written in ZEN. All page requests are served by a single ZEN server. The performance is quite OK. Now some of the pages contains ZEN Report results. You want it to refresh periodically and communicate to the rest of the page. IFRAME? No! We want it as contents of a DIV. Alright. CSP hyper events are doing the job and able to refresh the DIV with contents, which is nothing else but the result of a ZEN Report. So far so good. The „only ” issue, is that some of the queries behind a ZEN Report can consume significant server resources. Suddenly the well performing site suffers from lack of server resources. Solution? Off load the report generation to another server… In other words: bye-bye hyper events, so long to DIV & welcome back the good old IFRAME. Any other idea? Well, we could use XMLHttpRequest object of the browser. Or any convenience API of it from jQuery or AngularJS.

So the use case: we have ZEN server at zen.test.com. We access like http://zen.test.com/csp/app/app.ui.Page.zen . The origin of the page is http://zen.test.com. This page has an AJAX call to report.test.com. This call returns contents of a DIV. The inner HTML of the DIV is loaded from http://report.test.com/csp/app/app.report.Ballance.zen . Obviously the origin of the report is (http://report.test.com) is different from the page therefor we must do a Cross Origin AJAX call. But once we do, the browser starts complaining that the report server response does not have the required header fields. The CORS fields.

Adding CORS fields to HTTP Response

The REST is straightforward. We must add the required fields to the %response object in the CSP/ ZEN class. The only place to put it is the OnPreHTTP method.

The fields required:

n Access-Control-Allow-Origin: list of web servers which are allowed to include this page or “ * ” .

n Access-Control-Allow-Methods: list of HTTP methods.

n Access-Control-Allow-Headers: list of header fields which are not standard part of an OPTION response header. For CSP at least the Content-Type must be listed.

For more details please visit http://enable-cors.org/.

And finally a short example implementation.

```
Class cors.TestPage Extends %CSP.Page
{

ClassMethod OnPage() As %Status
{
	&html<{"test":"test"}>
	Quit $$$OK
}

Parameter CONTENTTYPE = "application/json";

/// Event handler for <b>PreHTTP</b> event: this is invoked before
/// the HTTP headers for a CSP page have been sent.  All changes to the
/// <class>%CSP.Response</class> class, such as adding cookies, HTTP headers,
/// setting the content type etc. must be made from within the OnPreHTTP() method.
/// Also changes to the state of the CSP application such as changing
/// %session.EndSession or %session.AppTimeout must be made within the OnPreHTTP() me
thod.
/// It is prefered
 that changes to %session.Preserve are also made in the OnPreHTTP() method
/// as this is more efficient, although it is supported in any section of the page.
/// Return <b>0</b> to prevent <method>OnPage</method> from being called.
ClassMethod OnPreHTTP() As %Boolean [ ServerOnly = 1 ]
{
	#dim %response as %CSP.Response
	set %response.Headers("Access-Control-Allow-Origin")="*"
	set %response.Headers("Access-Control-Allow-Methods")="GET,POST,PUT"
	set %response.Headers("Access-Control-Allow-Headers")="Content-Type"
	quit 1
}

}
```

#Frontend #Caché

Source URL:https://community.intersystems.com/post/enable-cors-csp-zen-applications