
Question

[Nicky Zhu](#) · Oct 16, 2017

Weird No 'Access-Control-Allow-Origin' header issue with Cors

Hi there,

I'm trying to develop a program calling Cache REST service twice with Cors. The first is fetching some information of a patient with GET request, the second is using obtained information to call another REST service(implemented in the same class though) with POST to perform other tasks.

The REST service was implemented as below:

```
Class WS.Service.REST.AIService Extends %CSP.REST
{

Parameter HandleCorsRequest = 1;

ClassMethod GetPatient(pID As %String) As %Status
{
    #dim tSC As %Status = $$$OK
    #dim e As %Exception.AbstractException
    Try {
        #dim patientInstance As HIS.Patient=##class(HIS.Patient).%OpenId(pID)
        set dynamicObject = {}
        set dynamicObject.Id = patientInstance.%Id()

        ....some codes to set the properties....

        set jsonStr = dynamicObject.%ToJSON()
        w jsonStr
    } Catch (e) {
        Set tSC=e.AsStatus()
    }
    Quit tSC
}

ClassMethod GetRisk() As %Status
{
    #dim tSC As %Status = $$$OK
    Do %response.SetHeader("Access-Control-Allow-Origin","*")
    set tSc = ##class(%ZEN.Auxiliary.jsonProvider).%ConvertJSONToObject(%request.Content, "%DynamicObject", .msgInstance)
    #dim patient As WS.Message.Patient = ##class(WS.Message.Patient).%New()

    ...some codes to set the properties...

    d %response.SetHeader("Access-Control-Allow-Origin","*")
    zw clinicMessage

    quit $$$OK
}
```

```
XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="/GetRisk" Method="POST" Call="GetRisk" />
<Route Url="/GetPatient/:Id" Method="GET" Call="GetPatient" />
</Routes>
}
```

Please notice that I've already set Parameter HandleCorsRequest = 1;

also, I override the following two methods in the same service class:

```
/// This is the CORS request handler. User should override this method in their subclass
/// if they don't want the default behavior
ClassMethod OnHandleCorsRequest(pUrl As %String) As %Status
{
    #dim tSC As %Status = $$$OK
    #dim e As %Exception.AbstractException

    #dim %request As %CSP.Request
    #dim %response As %CSP.Response

    #dim tOrigin,tHeaders,tMethod As %String

    Try {
        #; Get the origin
        Set tOrigin=$Get(%request.CgiEnvs("HTTP_ORIGIN"))

        #; Allow requested origin
        Do %response.SetHeader("Access-Control-Allow-Origin","*")

        #; Set allow credentials to be true
        Do ..SetResponseHeaderIfEmpty("Access-Control-Allow-Credentials","true")

        #; Allow requested headers
        Set tHeaders=$Get(%request.CgiEnvs("HTTP_ACCESS_CONTROL_REQUEST_HEADERS"))
        Do ..SetResponseHeaderIfEmpty("Access-Control-Allow-Headers",tHeaders)

        #; Allow requested method
        Set tMethod=$Get(%request.CgiEnvs("HTTP_ACCESS_CONTROL_REQUEST_METHOD"))
        Do ..SetResponseHeaderIfEmpty("Access-Control-Allow-Methods","GET,POST,OPTIONS")

    } Catch (e) {
        Set tSC=e.AsStatus()
    }
    Quit tSC
}

ClassMethod OnPreHTTP() As %Boolean [ ServerOnly = 1 ]
{
```

```
d %response.SetHeader("Access-Control-Allow-Credentials", "true")
d %response.SetHeader("Access-Control-Allow-Methods", "GET,POST,PUT,OPTIONS")
d %response.SetHeader("Access-Control-Allow-Origin", "*")
quit 1
}
```

Then I coded a web page using JQuery to sequentially call the two methods :

The start of the sequence, calling GetPatient first then call GetRisk

```
var restAddress = "http://192.168.113.132:57772/csp/ai/GetPatient/"
?var riskAddress = "http://192.168.113.132:57772/csp/ai/GetRisk/"
```

```
function checkPatientRisk(patientID){
var fullUrl = restAddress + patientID
var data = ""
$.ajax
(
({
type: "GET",
url: fullUrl,
dataType: 'json',
async: true,
success: function (data){
displayPatient(data)
},
beforeSend: function (xhr) {
xhr.setRequestHeader ("Authorization", "Basic " + btoa("_system" + ":" + "SYS"));
}
}));
}
```

```
function getRisk(patientData){
$.ajax({
type: "POST",
```

```
url: riskAddress,

dataType: 'json',

data: JSON.stringify(patientData),

success: function(data){

alert(data);

},

beforeSend: function (xhr) {

xhr.setRequestHeader ("Authorization", "Basic " + btoa("_system" + ":" + "SYS"));

}

});

}
```

The weird thing with these two calls is that the first GET request is handled while the second call to the POST method failed as following:

13:56:19.997	▶ XHR finished loading: OPTIONS " http://192.168.113.132:57772/csp/ai/GetPatient/1 ".
13:56:20.002	(index):210 ▶ {Id: "1", Name: "才英耿", Department: "外科", LastLabTime: "2017-10-13 14:50:51", pwbc: "10.6", ...}
13:56:20.021	XHR finished loading: GET " http://192.168.113.132:57772/csp/ai/GetPatient/1 ".
13:56:20.428	▶ XHR finished loading: OPTIONS " http://192.168.113.132:57772/csp/ai/GetRisk/ ".
✘ 13:56:20.429	POST http://192.168.113.132:57772/csp/ai/GetRisk/ :1 404 (Not Found)
✘ 13:56:20.429	POST http://192.168.113.132:57772/csp/ai/GetRisk/ :1 404 (Not Found)
✘ 13:56:20.429	Failed to load http://192.168.113.132:57772/csp/ai/GetRisk/ : No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin ' http://192.168.113.132 ' is therefore not allowed access. The response had HTTP status code 404.
13:56:20.451	XHR failed loading: POST " http://192.168.113.132:57772/csp/ai/GetRisk/ ".

I wonder why this happens and how may I handle the POST call correctly? Thanks in advance.

[#Caché](#) [#REST API](#)