

## Access to XMLHttpRequest at 'http://localhost:52773/IrisVSCode/app/test' from origin 'null' has been blocked by CORS policy

Question

[Amit Prajapati](#) · Aug 23, 2019

## Access to XMLHttpRequest at 'http://localhost:52773/IrisVSCode/app/test' from origin 'null' has been blocked by CORS policy

Hi All,

I have created a REST class in which have the Parameter HandleCorsRequest = 1;

I can able to access the API using Postman, but not with my web application. It throws the below error.

**Access to XMLHttpRequest at '<http://localhost:52773/IrisVSCode/app/test>' from origin 'null' has been blocked by CORS policy: Response to preflight request doesn't pass access control check: No 'Access-Control-Allow-Origin' header is present on the requested resource.**

Please note that I have tried all option which are mentioned on following URL:

<https://cedocs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?K...>

I have also provided all rights to my **\_system** user, which I am using while calling the webAPI.

I am provided the screenshots of my IRIS code, jQuery code (to call webAPI) and IRIS setting.

**Code Of TestAPI.cls file:**

```
Class IrisVSCode.TestAPI Extends %CSP.REST
{
Parameter HandleCorsRequest = 1;

XData UrlMap [ XMLNamespace = "http://www.intersystems.com/urlmap" ]
{
<Routes>
<Route Url="/test" Method="GET" Call="HelloWorld" Cors="true" />
</Routes>
}

ClassMethod HelloWorld() As %Status
{
WRITE "Hello World"
QUIT 1
}

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

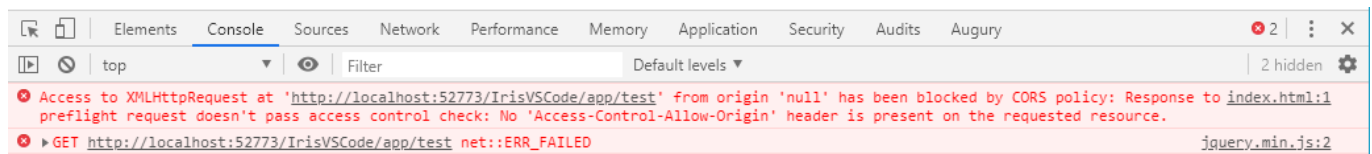
TRY {
#; Allow requested headers
SET tHeaders=$Get(%request.CgiEnvs("HTTP_ACCESS_CONTROL_REQUEST_HEADERS"))
DO ..SetResponseHeaderIfEmpty("Access-Control-Allow-Headers",tHeaders)

#; Allow requested method
SET tMethod=$Get(%request.CgiEnvs("HTTP_ACCESS_CONTROL_REQUEST_METHOD"))
DO ..SetResponseHeaderIfEmpty("Access-Control-Allow-Method",tMethod)
} CATCH (e) {
SET tSC=e.AsStatus()
}
QUIT tSC
}
}
```

jQuery Code to all the API:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
function btnClick() {
  alert("Button Clicked");
  $.ajax({
    type: "GET",
    crossDomain: true,
    url: "http://localhost:52773/IrisVSCode/app/test",
    headers: {
      'Authorization': 'Basic ' + btoa('_system:SYS'),
      'Access-Control-Allow-Origin': '*',
      'Content-Type': 'application/json',
      'Access-Control-Allow-Methods': 'GET,POST,OPTIONS'
    },
    contentType: "application/json;",
    dataType: "json",
    success: function (data) {
      alert(data);
      console.log(data);
    },
    failure: function (data) {
      alert(data.responseText);
    },
    error: function (data) {
      alert(data.responseText);
    }
  });
}
</script>
```

**Error Which I am facing while calling the API:**



**Working fine for POSTMAN:**

The screenshot shows a REST client interface. At the top, there's a method dropdown set to 'GET' and a URL field containing 'http://localhost:52773/IrisVSCode/app/test'. Below this, there are tabs for 'Authorization', 'Headers (1)', 'Body', 'Pre-request Script', and 'Tests'. The 'Authorization' tab is active, showing 'Basic Auth' selected. Fields for 'Username' (containing '\_system') and 'Password' (containing 'SYS') are visible. A 'Show Password' checkbox is checked. The response body is shown as 'Hello World'.

### Web application setting:

The screenshot shows the 'Edit Web Application' configuration page for '/IrisVSCode/app'. The page has a breadcrumb trail: 'System > Security Management > Web Applications > Edit Web Application'. The configuration includes: Name: '/IrisVSCode/app', Description: (empty), Namespace: 'IRISVSCODE', and 'Default Application for IRISVSCODE: /csp/healthshare/irisvscode'. The 'Enable Application' checkbox is checked. Under 'Enable', 'REST' is selected with a 'Dispatch Class' of 'IrisVSCode.TestAPI'. Under 'Security Settings', 'Unauthenticated' and 'Password' authentication methods are checked. Under 'Session Settings', 'Session Timeout' is set to 900 seconds and 'Use Cookie for Session' is set to 'Always'.

Edit definition for web application /IrisVSCode/app:

This is a detailed view of the configuration form. It has three tabs: 'General', 'Application Roles', and 'Matching Roles'. The 'General' tab is active. The form fields are: Name: '/IrisVSCode/app', Description: (empty), Namespace: 'IRISVSCODE', 'Default Application for IRISVSCODE: /csp/healthshare/irisvscode', 'Enable Application' (checked), 'Enable' (REST selected, Dispatch Class: 'IrisVSCode.TestAPI'), 'CSP/ZEN' (unchecked), 'Security Settings' (Resource Required: (empty), Group By ID: (empty), Allowed Authentication Methods: Unauthenticated and Password checked), 'Session Settings' (Session Timeout: 900 seconds, Event Class: (empty), Use Cookie for Session: Always, Session Cookie Path: '/IrisVSCode/app/').

Please let me know if you need some more information from me regarding this.

Thanks,

Amit Prajapati

[#Angular2](#) [#JavaScript](#) [#SOAP](#) [#InterSystems](#) [IRIS](#)

00 2 0 4 11,932

Log in or sign up to continue  
Add reply

**Source URL:** <https://community.intersystems.com/post/access-xmlhttprequest-httplocalhost52773irisvscodeapptest-origin-null-has-been-blocked-cors>