
Question

[max simpson](#) · Mar 21, 2021

How to cache REST Responses in IRIS?

Hi All,

I am implementing REST API's in IRIS using the below link as a guide.

I chose this approach so i could have a single entry point for requests regardless of namespace

<https://community.intersystems.com/post/restful-exception-handling>

a sample from the HIHLib.REST.Server class i am using is below. my question is does IRIS and CSP cache responses for requests to the CSP.REST class?

and if so what parameters or properties do i need to enable?

i notice the %CSP.Page class that %CSP.REST inherits from has an EXPIRES parameter that can be set, and %CSP.Response has a VaryByParam and Expires setting.

so from what i understand if i set the response VaryByParam setting to "*" and Expires to 300 then the same REST requests with the same parameters would return the same cached results for the next 5 minutes.

the problem with this approach would be that i need some way for the data being changed in the IRIS database to reset the response expiry.

i.e.

1. 15:00 calling application runs a GET request for patient?MRN=A
2. 15:00 CSP Web application calls HIHLib.REST.Server class and forwards request on to namespace for handling (via business services,process and operations)
3. 15:005 response is returned with EXPIRES 300 and VaryByParam set to *
4. 15:10 a seperate internal process results in patient with MRN A - first name changed from pete to bob
5. 15:15 calling application runs a GET request for patient?MRN=A
6. 15:15 web application checks response expiry and returns cached response from step 3 with first name pete

any advice on how you have implemented or reference to articles would be appreciated. at this point the implementation is new and load is light, but i'm concious of trying to make the response process as efficient as possible and reduce unnecessary load

One thought i've had is adding a staging table that the HIHLib.REST.Server class could refer to with response params, expiry and response. I could use the OnPreDispatch to check if there's an entry in the table and current time is less than expiryDate it could just return the cached response. which would mean the process at step 4 could update the staging table to clear the row and effectively clear the cache for that resource. effectively it's just moving the cache logic from CSP web gateway (which is where it appears the cache logic is - i could be wrong) to my HIHLib.REST.Server class, but at least that way it saves unnecessary load from business processes and operations.

```
///  
inspiration for this class from this post https://community.intersystems.com/post/restful-exception-handling  
Class HIHLib.REST.Server Extends %CSP.REST  
{ Parameter HandleCorsRequest = 1; Parameter CONTENTTYPE =  
"application/json"; XData UrlMap [ XMLNamespace =  
"http://www.intersystems.com/urlmap" ]  
{  
<Routes>
```

How to cache REST Responses in IRIS?

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
<Route Url=
"/:namespace/:service/:serviceVers
ion/:methodName" Method="GET" Call="InvokeEnsembleService"/>
<Route Url=
"/:namespace/:service/:serviceVers
ion/:methodName" Method="PUT" Call="InvokeEnsembleService"/>
<Route Url=
"/:namespace/:service/:serviceVers
ion/:methodName" Method="POST" Call="InvokeEnsembleService"/>
</Routes>
} ClassMethod InvokeEnsembleService(namespace As %String, service As %String,
serviceVersion As %String, methodName As %String) As %Status
{
#dim status As %Status = $$$OK
// act as an Oauth resource server and validate the JWT token in the API call https://
/docs.intersystems.com/irislatest/csp/docbook/Doc.View.cls?KEY=GOAUTH_resource
set accessToken=##class(%SYS.OAuth2.AccessToken).GetAccessTokenFromRequest(.sc)
if $$$ISOK(sc) {

    set isValidAccessToken=##class(%SYS.OAuth2.Validation).ValidateJWT("HIH",
accessToken)
    if isValidAccessToken { write !, "isValidAccessToken Huzzah!",! }

    //Azure AD doesnt currently support token introspection
} else {
    write "Error Getting Access Token="_$system.Status.GetErrorText(sc),!
} try { set status = ..
GetParametersFromCSPRequestAsListOfObjects(%request,.parameterList)
#dim serviceName As %String = namespace _ "." _ "REST" _ "." _ service _ "." _
"v" _ serviceVersion _ "." _ "Service"
set originalNamespace = namespace
set $NAMESPACE = namespace
set status = ##class(Ens.Director).CreateBusinessService(serviceName, .restService)
throw:$$$ISERR(status) ##class(HIHLib.REST.Exceptions.NotARegisteredService).%New()
set restRequest = ##class(HIHLib.Messages.REST.Request).%New()
set restRequest.Parameters = parameterList do ..
PopulateRequestHeader(%request,methodName,.restRequest) #dim jsonPayload As
%String = ""
#dim jsonFormatter As %JSON.Formatter = ##class(%JSON.Formatter).%New()
set status = jsonFormatter.FormatToString(%request.Content,.jsonPayload)
set restRequest.JSONpayload = jsonPayload set status = restService.
ProcessRESTRequest(restRequest,.response) If $$$ISERR(response.Status) {
do ##class(%REST.Impl).%ReportRESError(response.HttpStatusCode,response.Status,1)
return status
} if (response.JSONpayload '= "") {

//format the response, it may have already been formatted but format it again just in
case
    set status = jsonFormatter.FormatToString(response.JSONpayload,.jsonResponse)
write !, jsonResponse
} set $NAMESPACE = originalNamespace
// need to be able to return an error status without any error text i.e. an empty 404
response.
//but cant get it to work
#; Set %response.Status=response.HttpStatusCode
#; write !, "response
.HttpStatusCode:" _ response.HttpStatusCod
e //return ..ReportHttpStatusCode(response.HttpStatusCode)
}
```

How to cache REST Responses in IRIS?

Published on InterSystems Developer Community (<https://community.intersystems.com>)

```
catch exception {
    set httpErrorCode = $case(exception.%ClassName(1),
        ##class(
            HIHLib.REST.Exceptions.NotARegisteredService).%ClassName(1):503,
        :500)

    return ..ReportHttpStatusCode(httpErrorCode,exception.AsStatus())
} return status
}
```

[#CSP](#) [#REST API](#) [#InterSystems IRIS](#)

Product version: IRIS 2019.1

Source URL: <https://community.intersystems.com/post/how-cache-rest-responses-iris>