

Question

[Murali krishnan](#) · May 8, 2017

Fetch files from GitHub

let know the source code to fetch files from git hub repository..while running the attached file we are getting error.
ERROR #5001: Repository doesn't exist OR you don't have accesscoming1 1

Please help us to resolve the error.

Please find the code used :

```
Class Util.download Extends %Persistent
{
ClassMethod Update(Owner As %String, Repository As %String, Branch As %String = "", Username As %String, Password As %String, Namespace = {$Namespace}) As %Status
{
    write !, Username
    write !, Password
    Set req = ..CreateRequest(Username, Password)
    write !, req
    //Set req.Location = "repos/" _ Owner _ "/" _ Repository _ "/contents" // as described in https://developer.github.com/v3/repos/
    set req.Location = "http://github.com/vadsamm/sam.git"
    //set req.Location = ""
    //Do:(Branch'='') req.SetParam("ref", Branch) // if omitted the repository's default branch (usually master) would be used

    Set links = ##class(%ListOfDataTypes).%New()
    //Set st = ..ProcessDirectory("", req, .links)
    //write !, "coming1 " _ $system.Status.DisplayError(st)
    // Return:$$$ISERR(st) st
    write !, "coming2"
    Set namespace = $Namespace
    Zn Namespace
    Set st = ..DownloadFiles(links, req, .list)
    Set st2 = $system.OBJ.CompileList(.list, "cuk /checkuptodate=expandedonly")
    Zn namespace
    WRITE !, st
    WRITE !, st2

    Return $$$ADDSC(st, st2)
}

/**
Process one directory of GitHub repository. Recursive.<br>
<b>Path</b> -Internal repository path. Root is empty string<br>
<b>Request</b> - Authenticated/Set %Net.HttpRequest object.<br>
<b>Links</b> - List of links to raw files (which satisfy <b>IsCacheFile</b> condition s) from repository.<br>
**/
```

```

ClassMethod ProcessDirectory(Path As %String = "", Request As %Net.HttpRequest, ByRef
Links As %ListOfDataTypes) As %Status
{
    Set location = Request.Location
    write !, "location: "_location
    Set Request.Location = Request.Location _ Path

    Set st = Request.Get()
    write !, "st: "_st
    Return:$$$ISERR(st) st
    Return:(Request.HttpResponse.StatusCode = 404) $$$ERROR($$$GeneralError,"Repository doesn't exist OR you don't have access")
    Return:((Request.HttpResponse.StatusCode = 403) || (Request.HttpResponse.GetHeader("X-RATELIMIT-REMAINING")=0)) $$$ERROR($$$GeneralError,"API rate limit exceeded. Try logging in.")
    Return:(Request.HttpResponse.StatusCode != 200) $$$ERROR($$$GeneralError,"Received " _ Request.HttpResponse.StatusCode _ " expected 200")

    #dim objects As List of %ZEN.proxyObject
    #dim obj As %ZEN.proxyObject
    Set st = ##class(%ZEN.Auxiliary.jsonProvider).%ConvertJSONToObject(Request.HttpResponse.Data, .objects, 1)
    Return:$$$ISERR(st) st

    /*write !, objects.Count()

    For i = 1:1:objects.Count() {
        Set obj = objects.GetAt(i)
        If (obj.type = "dir") {
            Set st = ..ProcessDirectory("/ "_obj.name, Request, .Links)
            Return:$$$ISERR(st) st
        } ElseIf (obj.type = "file") {
            Do:..IsCacheFile(obj) Links.Insert(obj."download_url")
        } Else {
            // obj.type = "symlink" or obj.type = "submodule"
        }
    } */
    Set Request.Location = location // to keep track of where in the repository tree we are
    Return $$$OK
}

/**
Check that incoming file is the one you need.
**/
ClassMethod IsCacheFile(File As %ZEN.proxyObject) As %Boolean
{
    Set extensions = ",xml,cls,csp,csr,mac,int,bas,inc,dbl,prj,obj,pkg,gof,"
    Return:($L(File.name,".")=1) 0 //no extension
    Set File.Extension = $P(File.name,".", $L(File.name,"."))
    Return $F(extensions,"_",_ZCMT(File.Extension,"l")_"_",")
}

/**
Download list of files on https://raw.githubusercontent.com/ server.<br>
<b>Links</b> - List of links to raw files.<br>
<b>Request</b> - Authenticated/Set %Net.HttpRequest object.<br>
<b>loadedlist</b> - Returns an array of the items loaded.
**/

```

Fetch files from GitHub

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

```
ClassMethod DownloadFiles(Links As %ListOfDataTypes, Request As %Net.HttpRequest, Output Items) As %Status
{
    Kill Items
    //Set Request.Server = "raw.githubusercontent.com"
    Set Request.Server = "https://github.com/vadsamm/sam.git/"
    Set st = $$$OK

    For i = 1:1:Links.Count() {
        Set streq = Request.Get($e(Links.GetAt(i),35,*)) // Remove "https://raw.githubusercontent.com/" from URL.
        Set:$$$ISERR(streq) st=$$$ADDSC(st, streq)
        Set stload = $system.OBJ.LoadStream(Request.HttpResponse.Data,"",.error,.items)
        Set:$$$ISERR(stload) st=$$$ADDSC(st, stload)
        Merge Items = items
    }

    //Set Request.Server="api.github.com"
    Return st
}

ClassMethod CreateRequest(Username As %String = "", Password As %String = "") As %Net.HttpRequest
{
    Set namespace = $Namespace
    Set SSLConfig = "GitHub"

    Zn "%SYS"
    Do:'##class(Security.SSLConfigs).Exists(SSLConfig) ##class(Security.SSLConfigs).Create(SSLConfig)
    Zn namespace

    Set req=##class(%Net.HttpRequest).%New()
    write !,"req: "_req
    Set req.Https=1
    Set req.SSLConfiguration=SSLConfig
    Set req.Server="api.github.com"
    Do req.SetHeader("Accept","application/vnd.github.v3+json") // we want 3rd version of api

    If ((Username="") || (Password="")) { // supply Username and Password, if both are provided. GitHub accept Basic Auth
        Set req.Username = Username // https://developer.github.com/v3/auth/
        Set req.Password = Password
    }
    write !,"req: "_req
    Return req
}
```

[#Git](#) [#GitHub](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/fetch-files-github>
