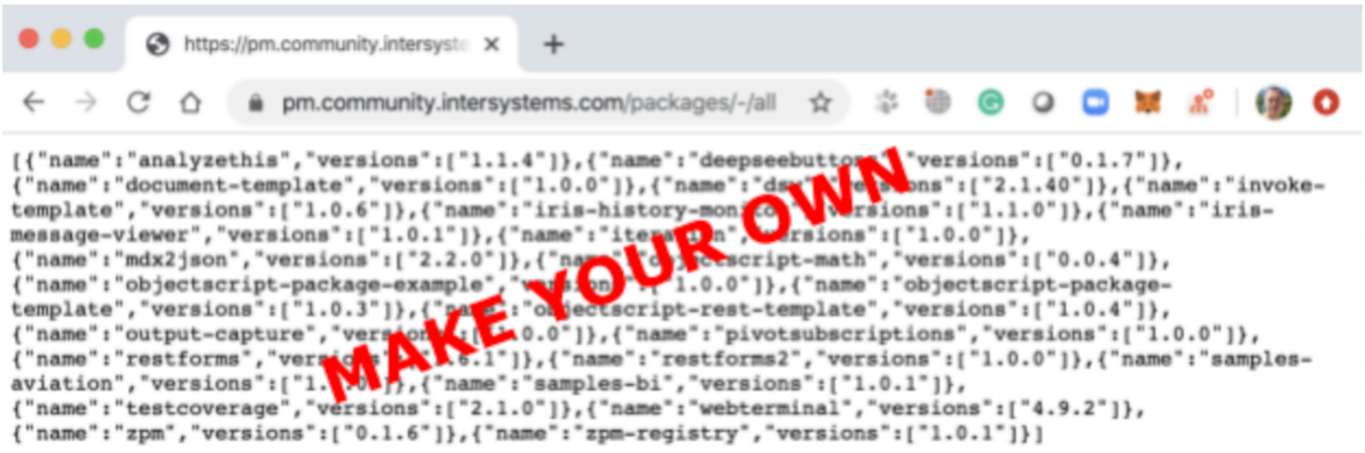
Article
Evgeny Shvarov · Feb 15, 2020 5m read

Open Exchange

Setting Up Your Own InterSystems ObjectScript Package Manager Registry

Hi Developers!


https://pm.community.intersyste
pm.community.intersystems.com/packages/-/all
MAKE YOUR OWN

```
[{"name":"analyzethis","versions":["1.1.4"]},{"name":"deepseebutto[illegible]"versions":["0.1.7"]},
{"name":"document-template","versions":["1.0.0"]},{"name":"ds[illegible]ns":["2.1.40"]},{"name":"invoke-
template","versions":["1.0.6"]},{"name":"iris-history-mon[illegible]versions":["1.1.0"]},{"name":"iris-
message-viewer","versions":["1.0.1"]},{"name":"ite[illegible]n",[illegible]rsions":["1.0.0"]},
{"name":"mdx2json","versions":["2.2.0"]},{"na[illegible]cript-math","versions":["0.0.4"]},
{"name":"objectscript-package-example",[illegible]1.0.0"]},{"name":"objectscript-package-
template","versions":["1.0.3"]},{[illegible]:"o[illegible]ectscript-rest-template","versions":["1.0.4"]},
{"name":"output-capture","ver[illegible].0.0"]},{"name":"pivotsubscriptions","versions":["1.0.0"]},
{"name":"restforms","ver[illegible].1"]},{"name":"restforms2","versions":["1.0.0"]},{"name":"samples-
aviation","versions":["1.[illegible]},{"name":"samples-bi","versions":["1.0.1"]},
{"name":"testcoverage","versions":["2.1.0"]},{"name":"webterminal","versions":["4.9.2"]},
{"name":"zpm","versions":["0.1.6"]},{"name":"zpm-registry","versions":["1.0.1"]}]
```

As you know the concept of ObjectScript Package Manager consists of ZPM client - client application for IRIS which helps you to install packages from the registry. And the code which works "on the other side" is ZPM Registry - server which hosts packages and exposes API to submit, list and install it. Now when you install the ZPM client it installs packages from community package registry, which si hosted on pm.community.intersystems.com

But what if you want your own registry? E.g. you produce different software packages for your clients and you want to distribute it via private registry? Also, you may want to use your own registry to deploy solutions with different combinations of packages.

Is it possible? The answer is YES! You can have it if you deploy ZPM registry on your server with InterSystems IRIS.

To make it happen you would need to set up your own registry server.

How to do that?

ZPM Registry can be installed as a package **zpm-registry**. So you can install zpm-client first (article, video), or take a docker-image with zpm-client inside and install zpm-registry as a package:

```
USER>zpm
zpm: USER>install zpm-registry
```

When zpm-registry is installed it introduces the REST end-point on *server:port/registry* with a set of REST-API routes. Let ' s examine it and let's open the GET requests oncommunity registry as an example.

/

- root entry shows the version of the registry software.

/ping

```
{"message":"ping"}
```

- entry to check the working status.

/spec

- swagger spec entry

/packages/-/all

- displays all the available packages.

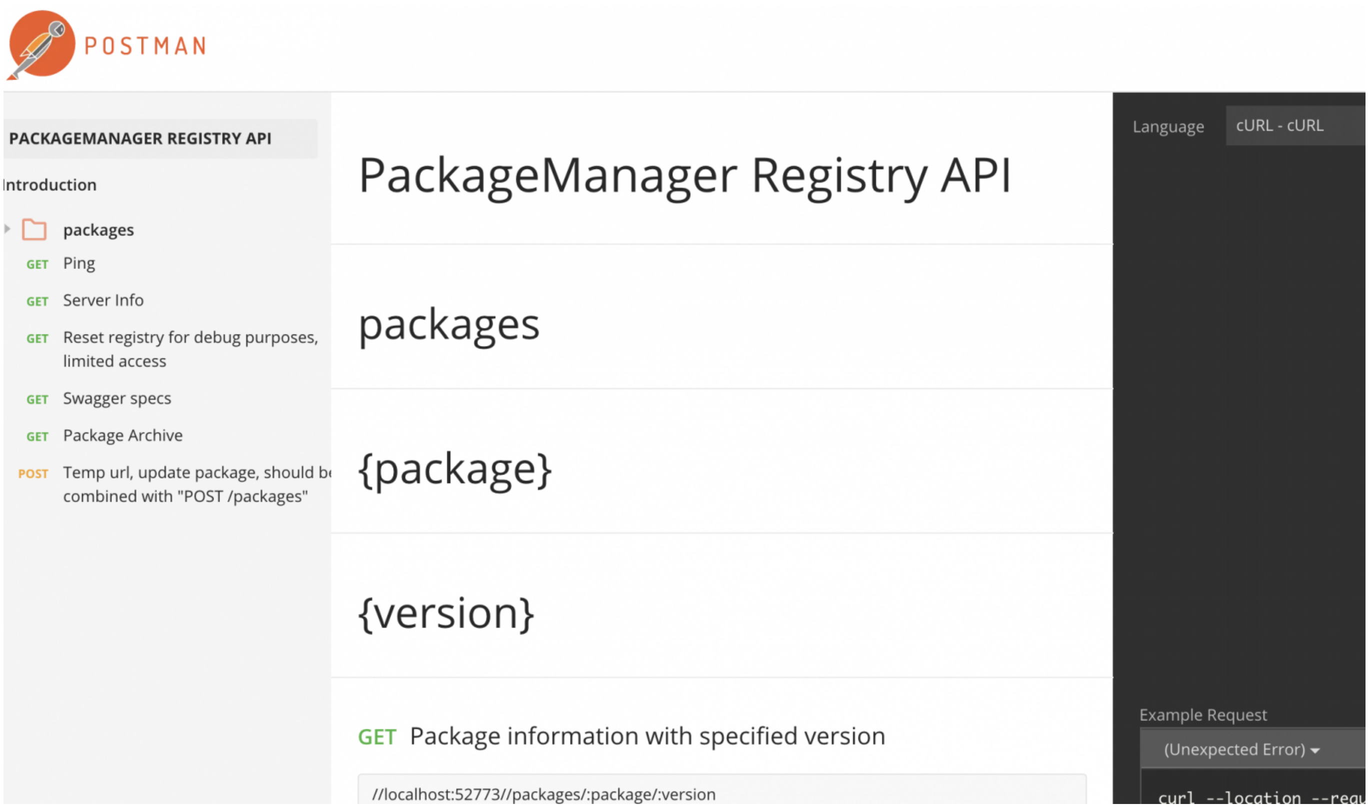
```
/packages/:package
```

- the set of GET entries for package deployment. ZPM client is using it when we ask it to install a particular package.

```
/package
```

- POST entry to publish a new package from the Github repository.

This and all other API you can examine e.g. in full documentation online generated with the help of /spec entry:


POSTMAN
PACKAGEMANAGER REGISTRY API
Introduction
packages
GET Ping
GET Server Info
GET Reset registry for debug purposes, limited access
GET Swagger specs
GET Package Archive
POST Temp url, update package, should b
combined with "POST /packages"
PackageManager Registry API
packages
{package}
{version}
GET Package information with specified version
Language
cURL - cURL
Example Request
(Unexpected Error)

```
//localhost:52773//packages/:package/:version
```

```
curl --location --requ
```

OK! Let's install and test ZPM-registry on a local machine.

1. Run docker container with IRIS 2019.4 and preinstalled ZPM-client (from developer community repo on docker hub).

```
% docker run --name my-iris -d --publish 52773:52773 intersystemsdc/iris-
community:2019.4.0.383.0-zpm
f40a06bd81b98097b6cc146cd61a6f8487d2536da1ffaf0dd344c615fe5d2844
% docker exec -it my-iris iris session IRIS

Node: f40a06bd81b9, Instance: IRIS

USER>zn "%SYS"

%SYS>Do ##class(Security.Users).UnExpireUserPasswords("*")

%SYS>zn "USER"

USER>zpm
zpm: USER>install zpm-registry

[zpm-registry]    Reload START
[zpm-registry]    Reload SUCCESS
[zpm-registry]    Module object refreshed.
[zpm-registry]    Validate START
[zpm-registry]    Validate SUCCESS
[zpm-registry]    Compile START
[zpm-registry]    Compile SUCCESS
[zpm-registry]    Activate START
[zpm-registry]    Configure START
[zpm-registry]    Configure SUCCESS
[zpm-registry]    Activate SUCCESS
zpm: USER>
```

2. Let ’ s publish a package in our new privately-launched zpm-registry.

To publish a registry you can make a POST request on registry/package end-point and supply the URL of the repository, which contains module.xml in the root. E.g. let's take the repo of objectscript-math application on Open Exchange by @Peter Steiwer : https://github.com/psteiwer/ObjectScript-Math

```
$ curl -i -X POST -H "Content-
Type:application/json" -u user:password
 -d '{"repository":"https://github.com/psteiwer/ObjectScript-
Math"}' 'http://localhost:52773/registry/package'
```

make sure to change the user and password to your credentials.

```
HTTP/1.1 200 OK
Date: Sat, 15 Feb 2020 20:48:13 GMT
Server: Apache
CACHE-CONTROL: no-cache
EXPIRES: Thu, 29 Oct 1998 17:04:19 GMT
PRAGMA: no-cache
```

```
CONTENT-LENGTH: 0
Content-Type: application/json; charset=utf-8
```

As we see, the request returns 200 which means the call is successful and the new package is available in the private registry for installation via ZPM clients. Let ’ s check the list of packages again:

Open in browser http://localhost:52773/registry/packages/-/all:

```
[{"name":"objectscript-math","versions":["0.0.4"]}]
```

Now it shows us that there is one package available.

That ’ s perfect!

Next question is how to install packages via ZPM client from an alternative repository.

By default when ZPM client is being installed it ’ s configured to work with public registry pm.community.intersystems.com.

This command shows what is the current registry set up:

```
zpm: USER>repo -list

registry
    Source:     https://pm.community.intersystems.com
    Enabled?    Yes
    Available?    Yes
    Use for Snapshots?    Yes
    Use for Prereleases?    Yes

zpm: USER>
```

But this could be altered. The following command changes the registry to your one:

zpm: USER>repo -n registry -r -url http://localhost:52773/registry/ -user *username* -pass *password*

Change here username and password to what is setup on your server available for /registry REST API.

Let ’ s check that alternative registry is available:

```
zpm: USER>repo -list


registry
    Source:     http://localhost:52773/registry/
    Enabled?    Yes
    Available?    Yes
    Use for Snapshots?    Yes
    Use for Prereleases?    Yes
    Username:     _SYSTEM
    Password:     <set>

zpm: USER>
```

So ZPM client is ready to work with another ZPM registry.

So, ZPM registry lets you build your own private registries which could be filled with any collection of packages either from public or from your own.

And ZPM client gives you the option to switch between registries and install from public or from any private registries.

Also check the article by @Mikhail Khomenko which describes how to deploy InterSystems IRIS docker-container with ZPM registry in Kubernetes cloud provided by Google Kubernetes Engine.

Happy coding and stay tuned!

#Docker #InterSystems Package Manager (IPM) #InterSystems IRIS #Open Exchange
Check the related application on InterSystems Open Exchange