Article
Robert Cemper · Feb 10, 2021 3m read

using ZPM for Node.js

Inspired by @Evgeny Shvarov and his recent article
Deploying InterSystems IRIS Embedded Python Solutions with ZPM Package Manager
I propagated the idea forward to do the same also for modules in Node.js.
The case is based on my example of IRIS Native API for Node.js

UPDATE

- All additional **required** Node.js components can now be installed at runtime.
- Only *intersystems-iris-native* module is pre-installed during Docker build.
- When Node.js is launched its Process id is displayed for debugging purpose.
- Some optical cosmetics and runtime improvements.

This step and adjusting access rights are covered in Dockerfile.
But all your .js modules are handled by ZPM.

The logic of the example hasn't changed.

- the Nodes.js service is started
- the selected address of the echo server is passed to it > **E**
- you compose the text to be transmitted > **N**
- you send it and see how the replies are dropping in > **S**
- stop service and exit > **X**

Different from the original example, the service is now running in background.
So its output is invisible but writes a log file that can be view by the action > **L**

Prerequisites

Make sure you have git and Docker desktop installed.

Installation

Clone/git pull this repo into any local directory

```
$ git clone https://github.com/rcemper/Using-ZPM-for-Node.js
```

Open the terminal in this directory and run:

```
$ docker-compose build
```

this may take some time to complete

Run the IRIS container with this project:

```
$ docker-compose up -d
```

How to Test it

Using IRIS terminal:

```
$ docker-compose exec iris iris session iris "##class(rccjs.WSockNodeJs).Run()"
```

```
*** Welcome to WebSocket by Node.js Native API Demo ***

********* Node.js process id = 1650 *********

Known Hosts (*=Exit) [1]:
1 wss://ws.postman-echo.com/raw
2 --- server 2 ----
3 --- server 3 ----
select (1): 1 ==> wss://ws.postman-echo.com/raw




Enter text to get echoed from WebSocketClient Service
Terminate with * at first position
or get generated text by %
or append new text with @

1 hello this is connected over
2 IRIS Native API for Node.js
3 -----------------
4 *

Select action for WebClient Service
New EchoServer (E), New Text(N), Send+Listen(S)
Show Log (L), Exit+Stop WsClient(X) [S] :s
%%%%%%%%%%%%%%%%%%%%%%%%%%

******* 3 Replies *******
1 hello this is connecte over
2 IRIS Native API for Node.js
3 -----------------

Select action for WebClient Service
New EchoServer (E), New Text(N), Send+Listen(S)
Show Log (L), Exit+Stop WsClient(X) [S] :L
%%%%%%%%%%%%%%%%%%%%%%%%%%

platform = linux: ubuntu
```

```
    *****************************
    *** no IRIS host defined ****
    Connect to IRIS on: localhost


Successfully connected to InterSystems IRIS.
*** wait 3sec for request ***
******* Startup done ********


    *** wait 3sec for request ***
    *** wait 3sec for request ***
    *** wait 3sec for request ***
    *** wait 3sec for request ***
    echoserver:  wss://ws.postman-echo.com/raw
    ** Lines to process: 3 **
    ********* next turn *********
    * WebSocket Client connected *
    ****** Client is ready ******


Line: 1 text> 'hello this is connecte over '
Received: 1 > 'hello this is connecte over '
Line: 2 text> 'IRIS Native API for Node.js '
Received: 2 > 'IRIS Native API for Node.js '
Line: 3 text> '----------------- '
Received: 3 > '----------------- '


    ******* lines sent: 3 ******
    *** replies received: 3 ****

    *** wait 3sec for request ***
    *** wait 3sec for request ***


Select action for WebClient Service
New EchoServer (E), New Text(N), Send+Listen(S)
Show Log (L), Exit+Stop WsClient(X) [S] :x
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

GitHub

#API #Node.js #InterSystems IRIS

Source URL:https://community.intersystems.com/post/using-zpm-nodejs