

記事

[Toshihiko Minamoto](#) · 2022年2月1日 5m read

[Open Exchange](#)

DockerマイロサーバとしてIRIS Native APIを使用したWebSocketクライアントJS

??? IRIS 2020.2????????????????
????????????????????????
??? InterSystems????????????????

動作中のデモを確認できるデモビデオを以て公開しています。 <https://youtu.be/dSV-0RJ5Olg>

皆さんこんにちは
完全に新しいIRISイメージ **たった4行**
のDockerコマンドを使って実行するイメージを使ってマイロサーバのデモを行います。
2020年6月1日 - rcc

すべてのパーツを1つのコンテナイメージにまとめてコンパクトなオンラインワンパクションが公開されました。
詳細はこちら: [IRIS-NativeAPI-Nodejs-compact](#)
2020年5月24日 - rcc

Dockerを使う簡単インストールを追加しました。コンテキストを参照
2020年5月25日 - rcc

Linux & Windowsに最適な検証済みの強スクリプトはこちら
<https://github.com/rcemper/WSockClientMicroSV/blob/master/READMEwindows.MD>
2020年5月26日 - rcc

このデモは、Cache用にすでに存在するNode.jsに基づくWebSocketクライアントを再設計したものです。
主に以のような変更点があります。

- 新しい [IRIS Native API for Node.js](#)の使用。特に [グローバル配列を擦](#) する場合
- 直接トリガされたクライアントからサーバ設計への変更
- マイロサーバ/マイロサーバの例として、結果を別のdockerイメージに配置
- マイロサーバの実行を制御するための単純なインターフェースをIRISに追加

ユーティリティをIRISホストで直接呼び出す代わりに、System Interoperability(別名 ENSEMBLE)で通常行うようにして、work-packagesをMicroServiceに送信します。
もちろん、2つ以上のWebSocketサーバオプションがあります。
WebSocketクライアントサービスがジョブを完了したら、その結果を取得します。

組み込みのWebSocketクライアントに対するメリットは、すべてのネットワーク、セキュリティ、ファイアウォールの問題を主要なデータサーバに奪付けないことです。Node.js
がこの分野で持っている経験と品質については言うまでもありません。

デモは デフォルトのechoサーバとして使用します。
次に、テキストを数行入力します。
独自のテキストの任意の場所に「Lorem Ipsum」を追加して、コンテンツを長くすると良いでしょう。

そして、それをサービスに送信して、エコを待ちます。
また、制御プログラムをExitまたはサービスをStopする前に、
テキストを変更するオプションもあります。

この処理はすべて非同期で実行されます。
完了を待たずに、それまでにechoサーバから受信した内容を、リスナーが
定期的に表示します。

インストールするには、以下が必要です。

- IRISのdocker イメージ(intersystems/iris-community:2020.2.0.199.0)
- WebSocketマイクサーバのdockerイメージ(`docker pull rcemper/rcc:demoJS`)
- 略されinit: initに次のdockerコマンドを実行してください、`docker run --name ini1 --init -it --rm / -privileged -v $(pwd):/external / rcemper/rcc:demoJS bash /rcc/init.sh ###`
元のアプローチは有効ではありませんが、必須はありません。###
- [IRIS-Docker-micro-Durability](#)を利用するためにOpen Exchangeまたはここから入手しWSockClientMicroSV.tar.gz
- demoデモトリの確認: Dockerイメージはnobodyであるため、`chmod 777`を
に設定します。 ###

これを 実行 するには、まず `demoデモトリ` からIRISを起動します(-dまたは-itを実行して動作を観察します)。(!)

```
docker run --name iris1 --init --rm -d \  
-p 52773:52773 -p 51773:51773 \  
-v $(pwd):/external \  
intersystems/iris-community:2020.2.0.199.0 \  
-b /external/pre.copy
```

次にMicroServerを起動します。

```
docker run --name rccl --init -it --rm \  
rcemper/rcc:demoJS /usr/bin/node /rcc/nodejs/WSockIris.js $(hostname -I)
```

.-itを使って起動し始め、以下が表示されます。

```
platform = linux: ubuntu  
  
*****  
Connect to IRIS on: 192.168.0.23  
Successfully connected to InterSystems IRIS.  
*** wait 3sec for request ***  
***** Startup done *****  
  
*** wait 3sec for request ***  
*** wait 3sec for request ***
```

次に、新しいLinuxターミナルで制御アプリケーションを実行します。

```
docker exec -it iris1 iris session iris ZSocket
```

以下が表示されます。

```
*** Welcome to WebSocket Micoservice demo ***
Known Hosts (*=Exit) [1]:
1  wss://echo.websocket.org/
2  --- server 2 ----
3  --- server 3 ----
select (1): ==> wss://echo.websocket.org/
#
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 socket microServer
2  now you got 2 lines
3  *

Select action for WebClient Service
New EchoServer (E), Send+Listen(S),New Text(N),Exit(X), Exit+Stop Client(Z) [S]s
%%%%%%%%%%

***** 0 Replies *****

***** 2 Replies *****
1  hello socket microServer
2  now you got 2 lines

Select action for WebClient Service
```

マイカサ・ピスでは以下が表示されます。

```
*** wait 3sec for request ***
echoserver:  wss://echo.websocket.org/
** Lines to process: 1 **
***** next turn *****
* WebSocket Client connected *
***** Client is ready *****

Line: 1 text> 'hello socket microServer '
Received: 1 > 'hello socket microServer '

Line: 2 text> 'now you got 2 lines '
Received: 2 > 'now you got 2 lines '

***** lines sent: 2 *****
*** replies received: 2 ****

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

事後警告。イメージのバージョンを必ず確認してください!!
私は `intersystems/iris-community:2020.2.0.204`??????????????

??: ???????????Linux???????????????

```
*** Welcome to WebSocket Micoservice demo ***
Known Hosts (*=Exit) [1]:
1 wss://echo.websocket.org/
2 --- server 2 ----
3 --- server 3 ----
select (1): ==> wss://echo.websocket.org/
#
Enter text to get echoed from WebSocketClient Service
Terminate with * at first position
or get generated text by %
or append new text with @

1 hi this is the final chek
2 what did you expect ?
3 so go
4 *

Select action for WebClient Service
New EchoServer (E), Send+Listen(S),New Text(N),Exit(X), Exit+
Stop Client(Z) [S]S
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
***** 0 Replies *****

***** 3 Replies *****
1 hi this is the final chek
2 what did you expect ?
3 so go

-----
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
*** wait 3sec for request ***
echoserver: wss://echo.websocket.org/
** Lines to process: 3 **
***** next turn *****
* WebSocket Client connected *
***** Client is ready *****
Line: 1 text> 'hi this is the final chek '
Received: 1 > 'hi this is the final chek '
Line: 2 text> 'what did you expect ? '
Received: 2 > 'what did you expect ? '
Line: 3 text> 'so go '
Received: 3 > 'so go '

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

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

[#Docker #JavaScript #Microservices #Node.js #ObjectScript #グローバル #コンテスト #コンテナ #InterSystems IRIS](#)
[InterSystems Open Exchangeで関連アプリケーションを確認してください](#)

ソールURL: <https://jp.community.intersystems.com/post/docker%E3%83%9E%E3%82%A4%E3%82%AF%E3%83%AD%E3%82%B5%E3%83%BC%E3%83%90%E3%83%BC%E3%81%A8%E3%81%97%E3%81%A6iris-native-api%E3%82%92%E4%BD%BF%E7%94%A8%E3%81%97%E3%81%9Fwebsocket%E3%82%AF%E3%83%A9%E3%82%A4%E3%82%A2%E3%83%B3%E3%83%88js%C2%A0>