

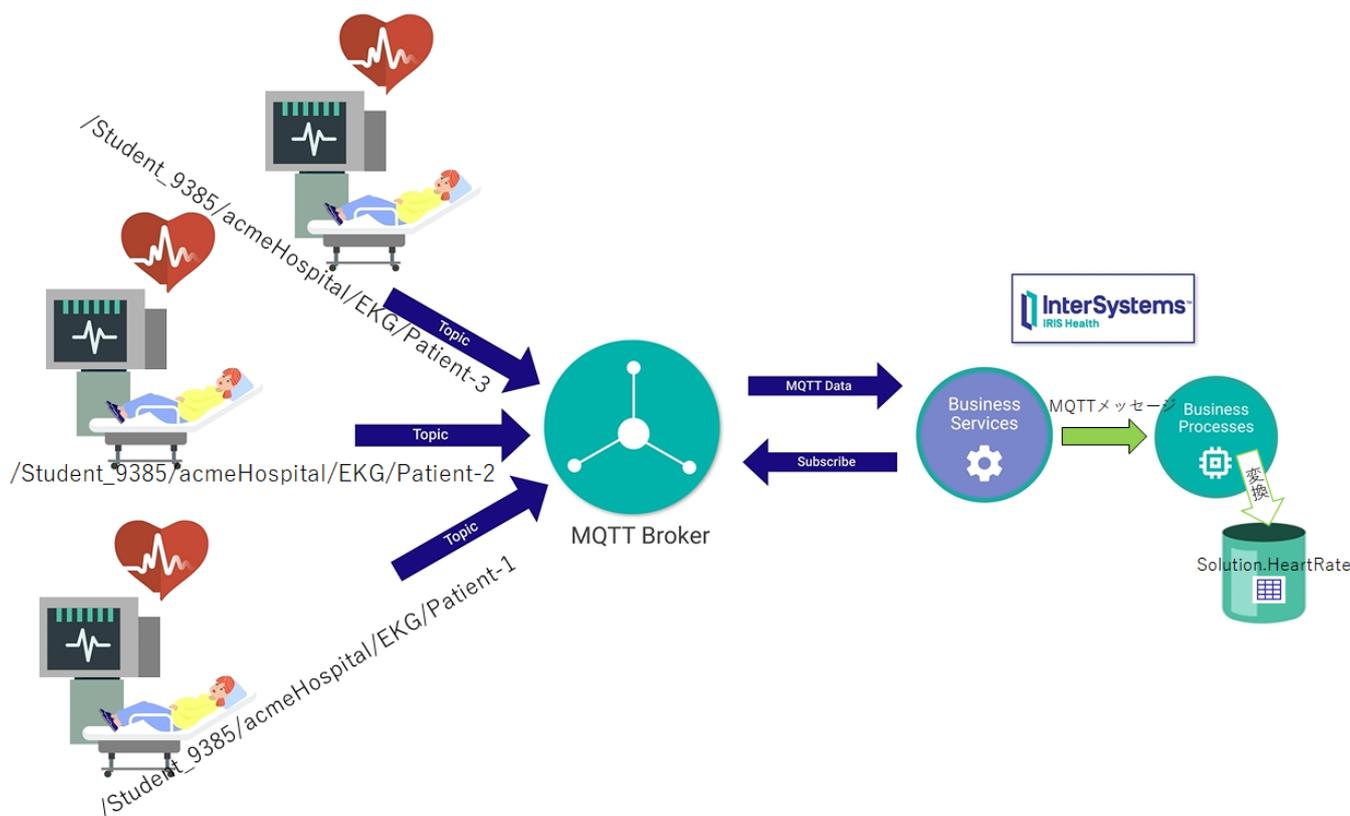
記事

[Mihoko Iijima](#) · 2021年11月4日 6m read

【GettingStarted with IRIS】MQTT を使った遠隔モニタリング (IRIS の MQTT アダプタを試してみよう！)

開発者の皆さん、こんにちは！

この記事では、[【GettingStarted with IRIS】シリーズ](#)の MQTT アダプタを簡単に試せるサンプルの利用方法についてご紹介します！



(MQTTブローカーはインターネット上に公開されている[テスト用ブローカー](#)を利用しています)

サンプルは、こちら <https://github.com/Intersystems-jp/Samples-MQTT-EKG-Devices> (コンテナで動作します)

IRIS / IRIS for Health のバージョン2020.1から、IoTの世界でよく利用される MQTT プロトコルに対応できる MQTT アダプタが追加されました。

MQTTインバウンドアダプタでは、メッセージの Subscribe が行え、MQTTアウトバウンドアダプタでは、メッセージの Publish が行えます。

サンプルでは、MQTT を使った遠隔モニタリングをテーマに、患者さんに装着した心電図から心拍数 (BPM) をリアルタイムに近い状態で取得し、モニタ画面に患者さん毎の心拍数を表示します (IRIS の MQTT インバウンドアダプタを利用したメッセージの Subscribe をご体験いただけます)。

Publish されるトピックについて

サンプルでは、演習環境毎にユニークになるようにコンテナ開始時に以下の形式でトピックを作成しています（末尾の # はワイルドカードの指定です）。

```
/Student_4629/acmeHospital/EKG/#
```

実際に Publish されるトピックは患者さんに装着した心電図のデータになるので、# の部分は、Patient-1 や Patient-2 などのように患者さんを特定できる文字列が入ります。

サンプルのシナリオは、1つの医療機関の患者情報を取得する流れにしています（本来であれば複数の医療機関の患者情報をモニタできるようにしたほうが良いのですがシンプルに試すため、1つの医療機関の患者情報を取得する流れにしています）。

トピックの流れ

サンプルには1つHTMLファイルが用意されています。このファイルをブラウザで開くと MQTT ブローカーに接続し、データ（心拍数）を1秒ごとにブローカーへ Publish します。

Patient-0



Beats per Minute (BPM)

181



Patient-1

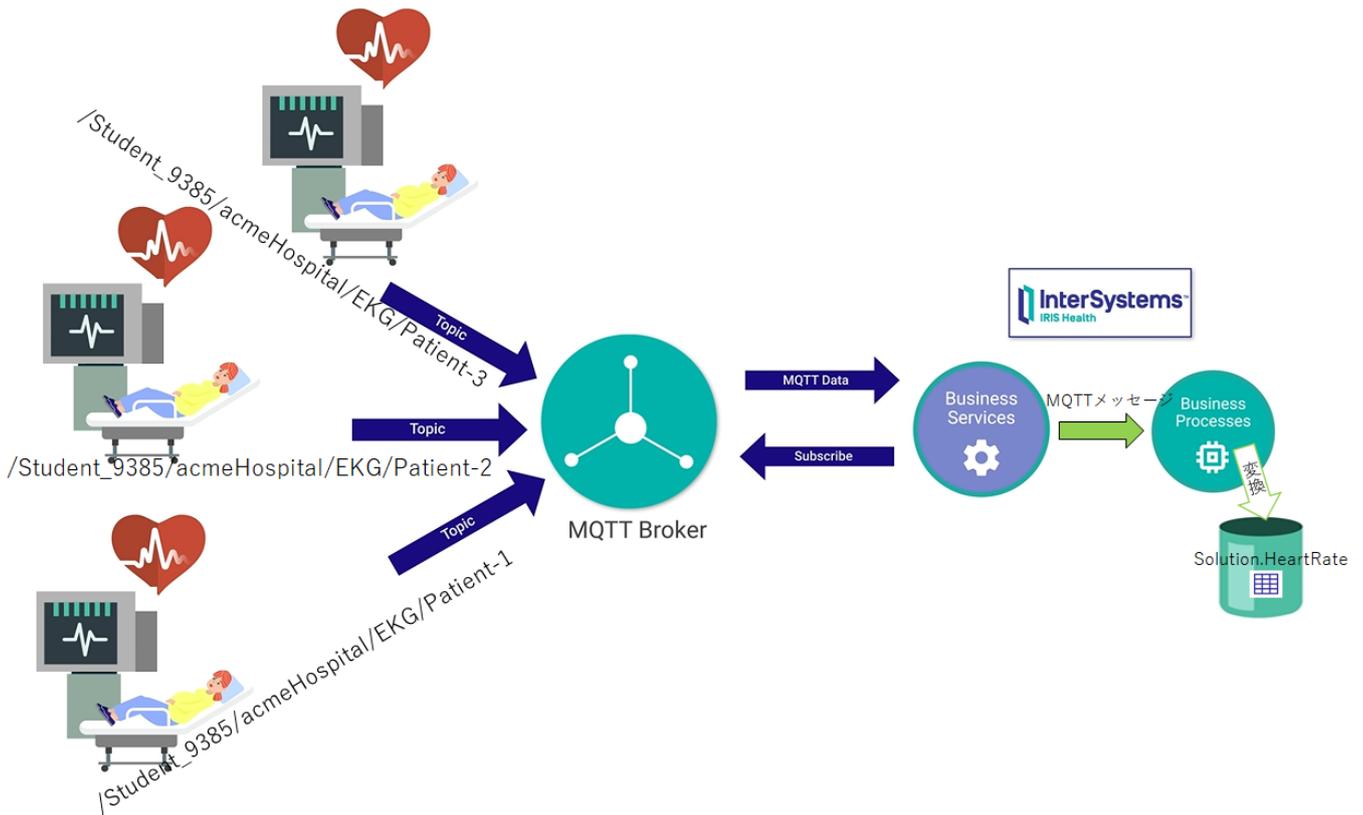


Beats per Minute (BPM)

91

Patient-2

IRIS は、MQTTブローカーから指定のトピックを Subscribe します。



MQTT ブローカーを Subscribe する設定を行った IRIS のサービスは、ブローカーからトピックを取得します。
 実際の設定は以下の通りです。

The screenshot shows the configuration page for the 'From_EKG_MQTT' service. The configuration details are as follows:

- Service Name: From_EKG_MQTT
- Process: Process_MQTT_Request
- Topic: /Student_3913/acmeHospital/EKG#
- URL: tcp://mqtt.eclipseprojects.io:1883

Red callouts provide additional information:

- このサービスで MQTTブローカーに接続しトピックをサブスクライブしています (This service connects to the MQTT broker and subscribes to the topic.)
- ここでトピックを指定しています。(演習環境毎異なります) (Here, the topic is specified. (It varies by exercise environment))
- MQTTブローカーのURL (MQTT broker URL)

受信したトピックは IRIS の中ではメッセージ (EnsLib.MQTT.Message) として扱われ、次のコンポーネントであるプロセス (図では ProcessMQTTRequest) に渡します。

Interoperability > プロダクション構成 - (Solution.MQTT)

プロダクション構成

開始する

停止する

プロダクション実行中

カテゴリ: 全て

凡例

サービス

プロセス

オペ



プロセスでは、受信した MQTT 用メッセージからモニタ表示に利用するデータ (Solution.HeartRate) に変換するため、データ変換を呼び出します。

プロセスエディタの開き方と、中で行われているデータ変換の呼び出しの設定を確認する方法は以下の通りです。

1. プロセスの名称をクリックします

2. 情報を提供する設定を展開します。

3. クラス名の右横にある をクリックします

4. DTLエディタボタンを押すとデータ変換クラスが開きます。

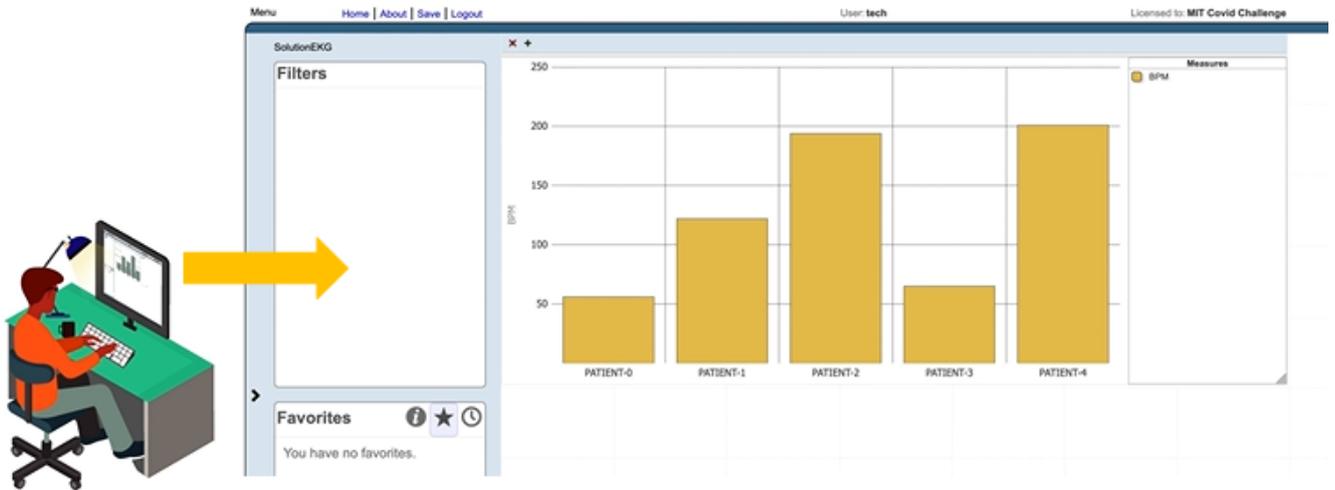
MQTT用メッセージからサンプルが利用する監視用データに変換しています。

Subscribeした情報から患者IDだけを取得し設定しています。

#	アクション	条件	プロパティ	値	キー/変換
1	set		target.PatientID	..Piece(source.Topic,"/";**	**
2	set		target.BPM	source.StringValue	**

患者ごとの測定値を参照する流れ

MQTT ブローカーから Subscribe した心拍数をリアルタイムに近い状態で画面表示するため、データ変換で作成された Solution.HeartRate から1秒ごとに患者ごとの心拍数を収集し、画面に表示しています。



この表示を行うため、1秒間隔で Solution.HeartRate に対するSELECT文が実行されています。

このクエリを実行しているのが、メトリックと呼ぶクラスです。プロダクションでは以下の場所に設定されています。

The screenshot shows the 'プロダクション構成' (Production Configuration) page for 'Solution.MQTT'. The page has a header with '開始する' (Start) and '停止する' (Stop) buttons. Below the header, there are sections for 'サービス' (Services) and 'プロセス' (Processes). Under 'サービス', 'Solution.MetricMQTT' is listed and highlighted with a red box. To the right, a configuration panel for 'Solution.MetricMQTT' is visible, showing various settings. The '呼び出し間隔' (Call Interval) is set to 1, which is also highlighted with a red box.

メトリックでは、指定の呼び出し間隔で Solution.HeartRate から患者ごとの BPM を収集しています。詳細は[ソースコード](#)をご参照ください。

ということで、実際にサンプルを動かして動作を確認してみましょう！手順は以下の通りです。

- (1) git clone
- (2) コンテナビルド & 開始
- (3) 演習環境毎のトピック作成

(4) プロダクション開始

(5) サンプルHTMLをブラウザで開いて Publish 開始！

(6) モニタ画面で状況確認

(1) git clone

```
git clone https://github.com/intersystems/Samples-MQTT-EKG-Devices
```

(2) コンテナの開始

git clone で作成されるディレクトリに移動します。

```
cd Samples-MQTT-EKG-Devices
```

Windows 以外でお試しいたいている場合は、setup.sh を実行します。

```
./setup.sh
```

コンテナを開始します。

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

(3) 演習環境毎のトピック作成

```
docker-compose exec iris iris session iris -U %SYS "##class(App.Installer).Initialize  
Docker()"
```

この実行で演習環境で使用するトピックを指定しています。

注意：

app.htmlで使

用するJavaScriptとプロダクシ

ョン設定に演習環境用のトピックを追記しています。この実行を行わないとサンプルは動作しません。

IRISの起動が完了していないと、上記メソッド実行後、以下のメッセージが出力されます。

```
Sign-on inhibited: Startup or Installation in progress
```

このメッセージが表示される場合は、しばらく待ってから再度実行してください。

正常に実行できると、以下のように演習環境用のトピックが表示されます。

```
You have successfully initiated the MQTT exercise  
Please take note of your topic top-  
level string: /Student_4908/acmeHospital/EKG/#  
press enter to continue
```

Enterで元の画面に戻ります。

(4) プロダクション開始

管理ポータルを開きます。 <http://localhost:52773/csp/sys/UtilHome.csp>

ユーザ名： SuperUser

パスワード (大文字で設定されています)： SYS

Interoperability > INTEROPネームスペース選択 > 一覧 > プロダクション

ようこそ, SuperUser

表示: [アイコン]

%SYS ネームスペースはプロダクションをサポートし:

別のネームスペース:
 プロダクションに使用可能なネ
 HSLIB
 HSSYS
INTEROP
 USER

ホーム
 Health
 Analytics
Interoperability
 システムオペレーション

構成
 ビジネス・プロセス
 構築
 データ変換
 表示
 ビジネス・ルール
一覧
 レコード・マップ
 モニタ
プロダクション
 管理
 相互運用
 テスト

Interoperability > プロダクション一覧

プロダクション一覧 新規 開く エクスポート 削除 ドキュメント

プロダクション名	ステータス	最終開始時刻	最終停止時刻
» Solution.MQTT	停止	2021-10-29 03:39:12.735	2021-10-29 06:20:53.873

Solution.MQTT の行を選択してから「開く」ボタンをクリックします

Interoperability > プロダクション構成 - (Solution.MQTT)

プロダクション構成 開始する 停止する

プロダクションが停止しました

サービス: From_EKG_MQTT, Solution.MetricMQTT
 プロセス: Process_MQTT_Request

プロダクションの開始

プロダクション開始中: Solution.MQTT
 少々お待ちください。進行状況を示すメッセージを上に表示します...

06:23:48.710:Ens.Director: プロダクション 'Solution.MQTT' 開始中...
 06:23:49.774:Ens.Director: プロダクション 'Solution.MQTT' が開始しました。完了しました。

「プロダクションの開始」の子画面が出ます。OKボタンをクリックし閉じてください。

Interoperability > プロダクション構成 - (Solution.MQTT)

プロダクション構成 開始する 停止する

プロダクション実行中

サービス: From_EKG_MQTT, Solution.MetricMQTT
 プロセス: Process_MQTT_Request
 オペレーション

(5) サンプルHTMLをブラウザで開いて Publish 開始!

app.html を開きます <http://localhost/app.html>

サーバ名はご利用の環境に合わせてご変更ください。

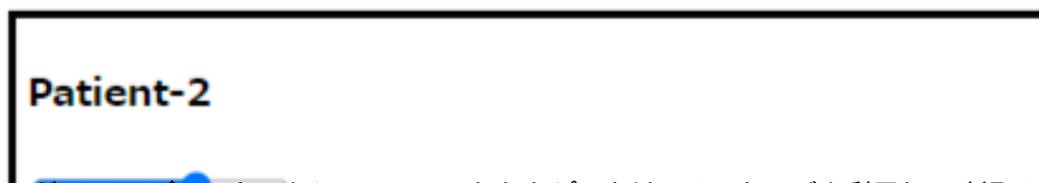
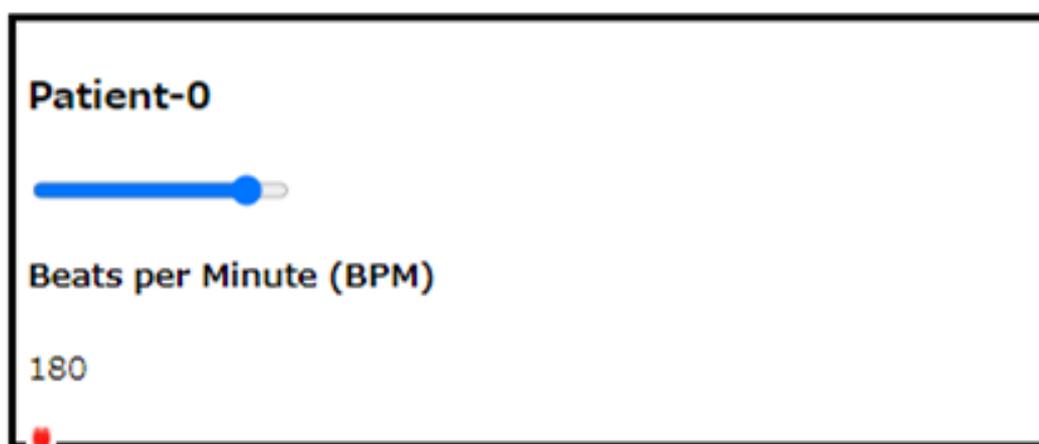
このサンプルでは、心電図を装着した3人の患者さんから心電図データの取得をサーバーに接続し、1秒ごとに心拍数を送信します。

追加 ボタンを使って、患者を追加することもできます。

あなたのMQTTトピックは /Student_3913/acmeHospital/EKG/# です

患者情報追加 (追加ボタンをクリックしてください)

追加



IRIS が MQTT ブローカーから Subscribe したトピックは、メッセージを利用して確認できます。

開いたプロダクション画面で、サービス : FromEKG MQTT をクリックし、画面右の「メッセージ」タブを選択します。ヘッダの列にある番号をクリックするとトレース画面が開き、受信したトピックとその値が確認できます。

Interoperability > プロダクション構成 - (Solution.MQTT)

プロダクション構成

開始する 停止する

ソート: 名前 ステータス 番号 表示: [アイコン]

プロダクション実行中

サービス From_EKG_MQTT Process_MQTT_Request

セッションID: 1173

ヘッダ	日時	ステータス	ソース	ターゲット
1173	18:41:40	Completed	From_EKG_MQTT	Process_MQTT_Request
1172	18:41:39	Completed	From_EKG_MQTT	Process_MQTT_Request
1171	18:41:39	Completed	From_EKG_MQTT	Process_MQTT_Request
1170	18:41:39	Completed	From_EKG_MQTT	Process_MQTT_Request
1169	18:41:39	Completed	From_EKG_MQTT	Process_MQTT_Request
1168	18:41:38	Completed	From_EKG_MQTT	Process_MQTT_Request
1167	18:41:38	Completed	From_EKG_MQTT	Process_MQTT_Request
1166	18:41:38	Completed	From_EKG_MQTT	Process_MQTT_Request
1165	18:41:38	Completed	From_EKG_MQTT	Process_MQTT_Request

EnsLib.MQTT.Message

<オブジェクトId> 1171

StringValue	180
Topic	/Student_3913/acmeHospital/EKG/Patient-0

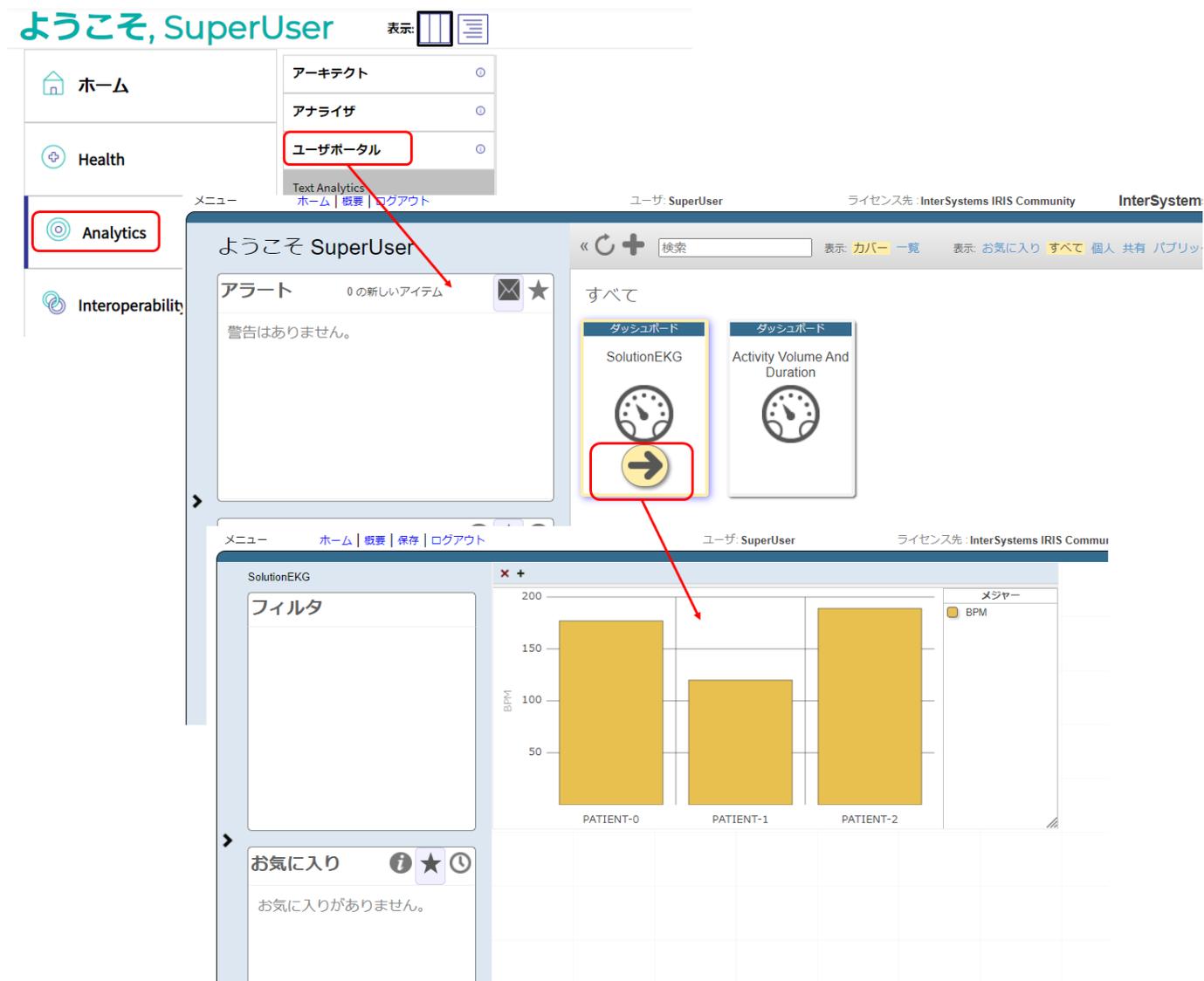
トピックと値を確認できます。

(6) モニタ画面で状況確認

モニタ画面を開き、患者ごとの心拍数を確認します。

管理ポータル > Analytics > ユーザポータル > SolutionEKG

管理ポータル <http://localhost:52773/csp/sys/UtilHome.csp>



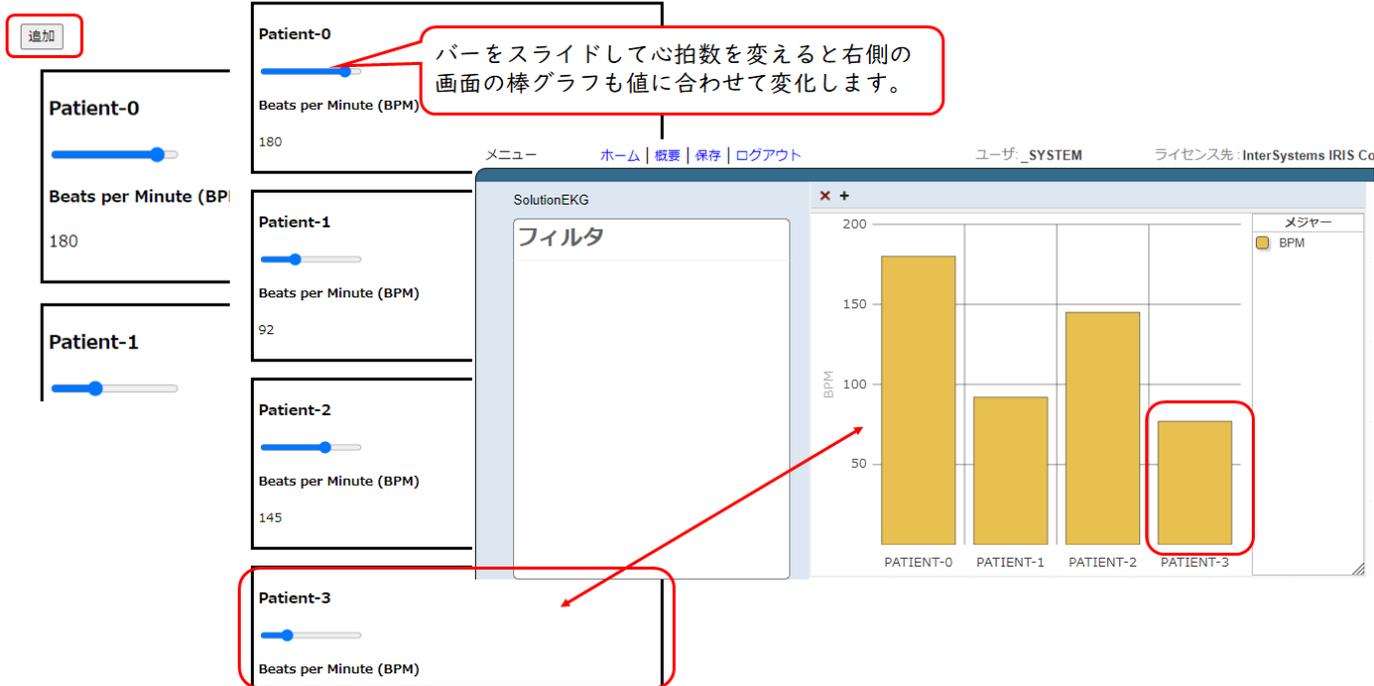
初期状態では3名の患者データが表示されます。

app.html にある「追加」ボタンで心電図を付けた患者を増やすことができます。

このサンプルでは、心電図を装着した3人の患者さんから心電図データが送信され、**追加** ボタンを使って、患者を追加することもできます。

あなたのMQTTトピック | **患者情報追加** (追加ボタンをクリックしてください)

患者情報追加 追加



app.html で患者を増やすと、モニタ用画面にも表示が増えます。

また、各患者のバーをスライドさせ、心拍数を変化させると、グラフもそれに合わせて変化し、MQTT ブローカーから Subscribe できていることがわかります。

以上でサンプルの動作確認は終了です。

app.html を閉じると MQTT ブローカーへの Publish が終了します。

最後に、コンテナを停止します。

```
docker-compose stop
```

コンテナを破棄する場合は、以下のコマンドを実行してください。

```
docker-compose down
```

サンプルには、完成形の Solution パッケージの他に、Demo パッケージが用意されていてプロダクションの定義やプロセスの作成を試すこともできます。

作成方法については、次の記事でご紹介する予定です！

[#Docker #Git #IoT #相互運用性 #InterSystems IRIS #InterSystems IRIS for Health](#)

[82%92%E4%BD%BF%E3%81%A3%E3%81%9F%E9%81%A0%E9%9A%94%E3%83%A2%E3%83%8B%E3%82%BF%E3%83%AA%E3%83%B3%E3%82%B0%EF%BC%88iris-%E3%81%AE-mqtt-%E3%82%A2%E3%83%80%E3%83%97%E3%82%BF%E3%82%92%E8%A9%A6%E3%81%97%E3%81%A6%E3%81%BF%E3%82%88%E3%81%86%EF%BC%81%EF%BC%89](#)