

記事

[Toshihiko Minamoto](#) · 2022年4月7日 4m read

[Open Exchange](#)

mermaid ライブラリを使ったダイアグラムの基本的な描画方法

皆さん、こんにちは！

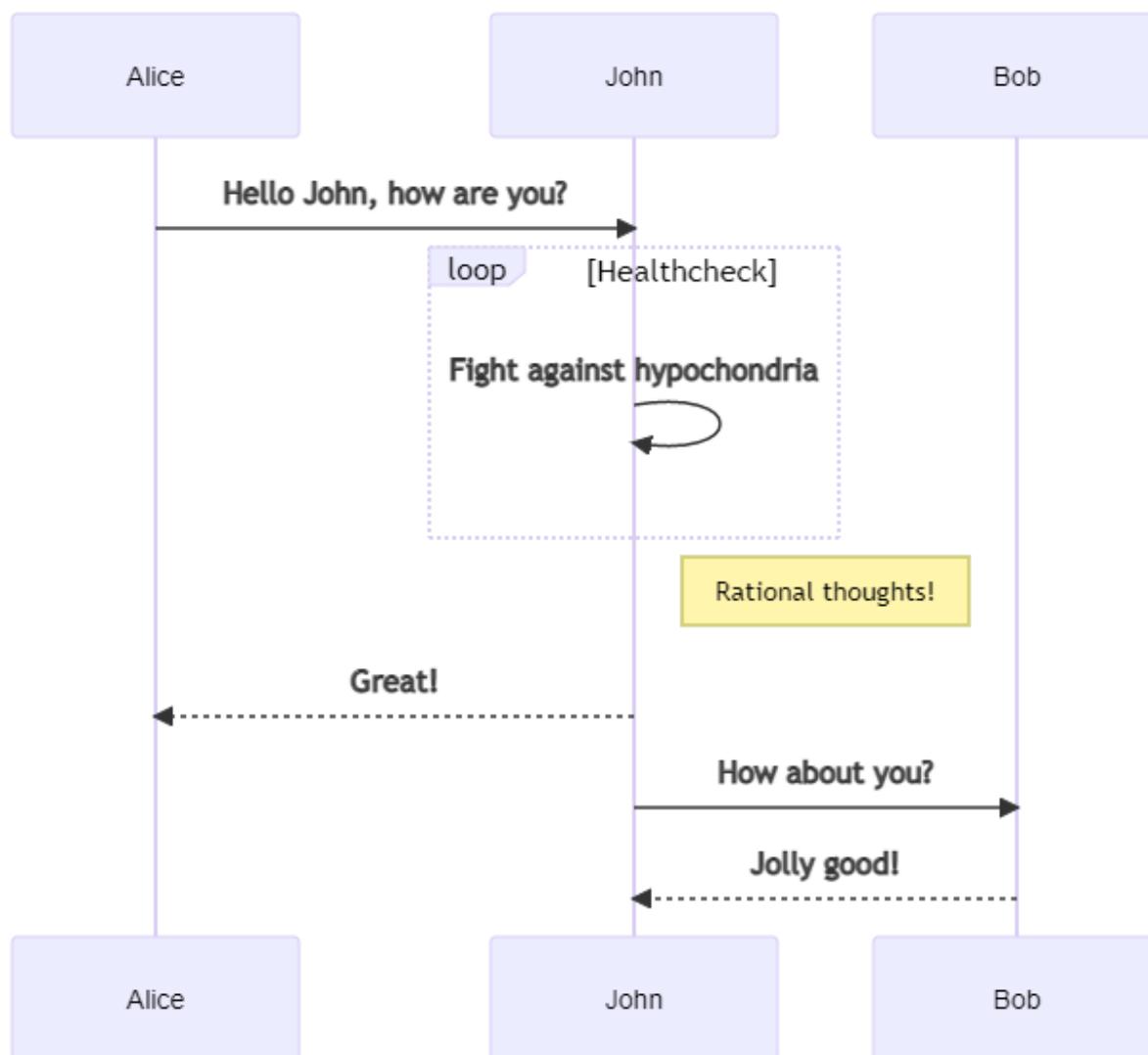
@Henrique.GoncalvesDias と私は、[MessageViewer](#) の最近のアップデートにおいて、IRIS Interoperability でメッセージを可視化する新しい方法を提案しました。そのアップデートでは、UML シーケンス図に基づく可視化を提供しようとしてしました。詳細は、[前](#)の記事をご覧ください。

このようなダイアグラムを描画するために必要な困難な幾何学計算を行えるように、素晴らしい [mermaid JS](#) オープンソースライブラリを使用しました。この記事で紹介したいのは、このライブラリの使用方法です。ここでは、シーケンス図のみに焦点を当てますが、このライブラリでは[非常にたくさん](#)のことを行えることを忘れないでください。

mermaid では Markdown に着想を得た構文を使用して、ダイアグラムを定義します。非常に直感的であるため、退屈な文章を長々と書くのではなく、例を紹介したいと思います。

```
sequenceDiagram
Alice->>John: Hello John, how are you?
loop Healthcheck
    John->>John: Fight against hypochondria
end
Note right of John: Rational thoughts!
John-->>Alice: Great!
John->>Bob: How about you?
```

mermaid エンジンはこの定義に基づいて、SVG を使って直接 Web ページに以下のダイアグラムをレンダリングします。



この例は mermaid のドキュメントから得たもので、この [オンラインエディター](#) で試すことができます。色々試すことのできる [構成がたくさん](#) あります。

ご覧の通り、ダイアグラムの定義では、アクターと参加者、相互に送信しているイベントとメッセージを指定するだけです。

また、Web ページにダイアグラムを表示するには、div コンテナにダイアグラムを指定し、mermaid エンジンを初期化してダイアグラムをレンダリングする JS コードのみが必要です。

```
<div class="mermaid">
sequenceDiagram
Alice->>John: Hello John, how are you?
loop Healthcheck
    John->>John: Fight against hypochondria
end
Note right of John: Rational thoughts!
John-->>Alice: Great!
John->>Bob: How about you?
Bob-->>John: Jolly good!
</div>
```

```
mermaid.initialize({
  startOnLoad: true,
  theme: 'forest'
});
```

この例は、こちらの[フィドル](#)にあります。

これは、提案された作業のフロントエンドベースです。バックエンドでは、IRIS 相互運用性セッションからメッセージを取得し、適切な JSON オブジェクトにフォーマットしてフロントエンドに送り返す REST エンドポイントを設定する必要があります。この記事の焦点はフロントエンドのコードであるため、バックエンドの実装には触れませんが、[dispatch](#) と [service](#) クラスでこれを確認できます。

バックエンドは、以下のような JSON を送り返します。

```
{
  "participants": [
    "diashenrique.messageviewer.Service.SendMessage",
    "diashenrique.messageviewer.Operation.ConsumeMessageClass"
  ],
  "messages": [
    {
      "id": "1182",
      "from": "diashenrique.messageviewer.Service.SendMessage",
      "to": "diashenrique.messageviewer.Operation.ConsumeMessageClass",
      "message": "2021-10-05 03:16:56.059 SimpleMessage"
    },
    {
      "id": "1183",
      "from": "diashenrique.messageviewer.Operation.ConsumeMessageClass",
      "to": "diashenrique.messageviewer.Service.SendMessage",
      "message": "2021-10-05 03:16:56.06 NULL"
    }
  ]
}
```

最後に、簡単な JS 関数を使用して、この JSON を mermaid シーケンス図の仕様に合わせて以下のように変換します。

```
sequenceDiagram
  autonumber
  participant P0 as diashenrique.messageviewer.Service.SendMessage
  participant P1 as diashenrique.messageviewer.Operation.ConsumeMessageClass
  P0->>P1: 2021-10-05 03:16:56.059 SimpleMessage
  P1->>P0: 2021-10-05 03:16:56.06 NULL
```

そして、これがレンダリングされたシーケンス図です。

完全な JS コードは[こちら](#)で確認できます。

以上です。この記事があなたの素晴らしいプロジェクトにいくらかでも役立てられれば幸いです。

それではまた！

[#JavaScript](#) [#フロントエンド](#) [#相互運用性](#) [#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

ソースURL:

<https://jp.community.intersystems.com/post/mermaid-%E3%83%A9%E3%82%A4%E3%83%96%E3%83%A9%E3%83%AA%E3%82%92%E4%BD%BF%E3%81%A3%E3%81%9F%E3%83%80%E3%82%A4%E3%82%A2%E3%82%B0%E3%83%A9%E3%83%A0%E3%81%AE%E5%9F%BA%E6%9C%AC%E7%9A%84%E3%81%AA%E6%8F%8F%E7%94%BB%E6%96%B9%E6%B3%95>