

# MIDI Sync

Online MIDI Player

<https://github.com/mvamgc/midi-sync>

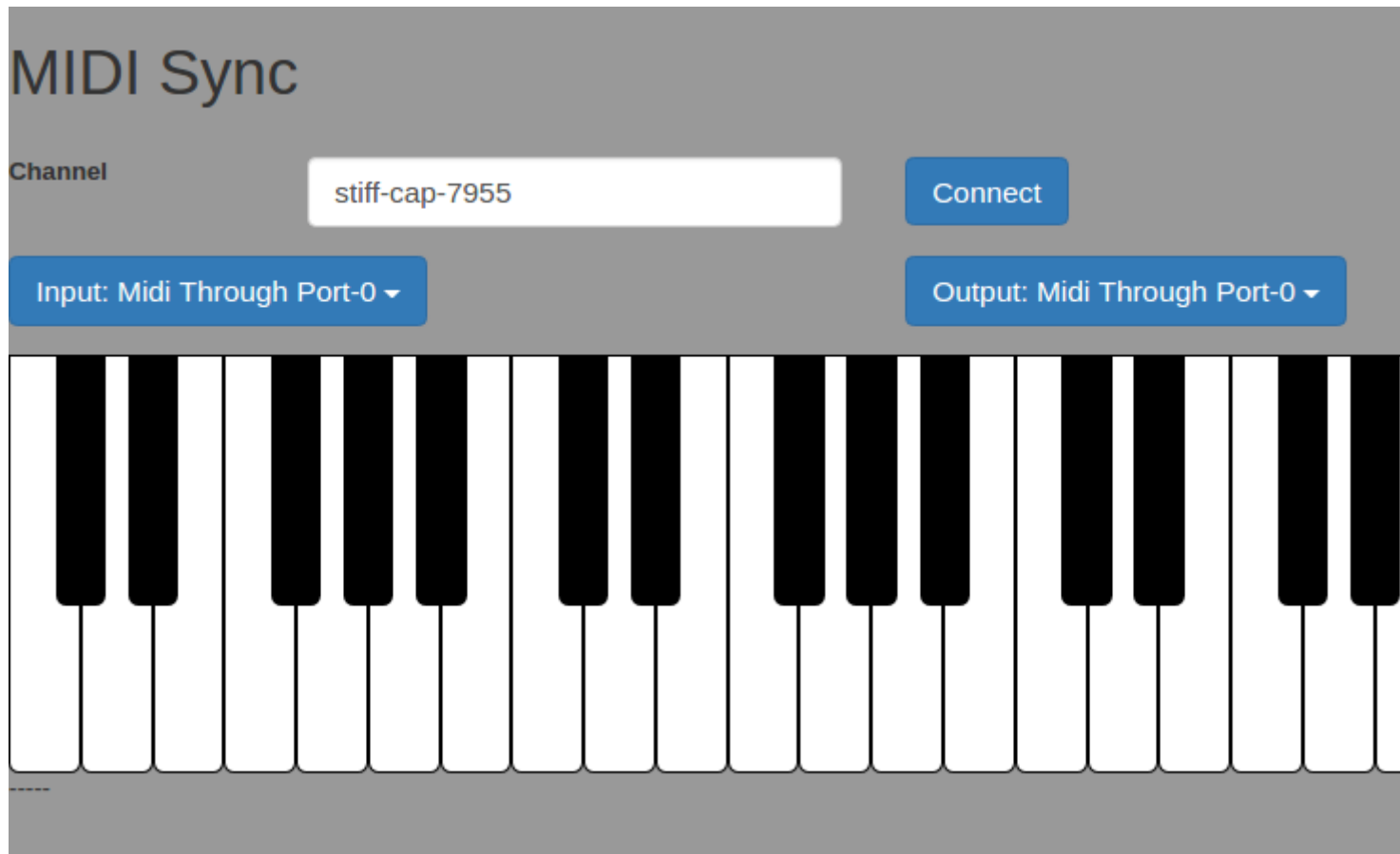
<https://gitlab.com/mvamgc/midi-sync-ref>

<https://floating-hamlet-22000.herokuapp.com/>

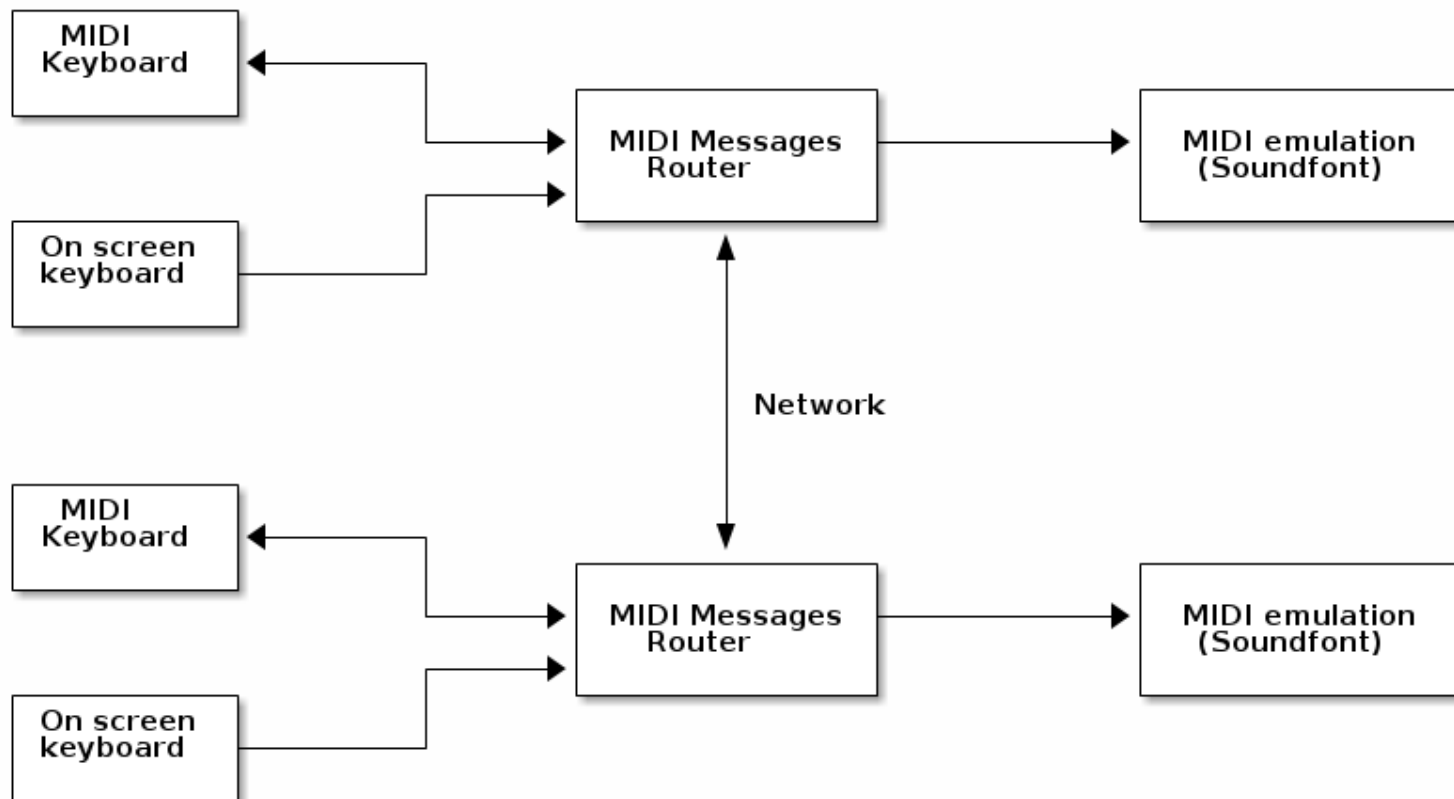
# Project description

- Play on a MIDI keyboard (piano) online with other people.
- Exchange with MIDI messages over the Internet with low latency.
- Remote MIDI message are playing on local MIDI device or on the Sound card.
- Teach another person to play a piano.
- Preferably no local installations, web-based program.

# Project description



# Project description



# Implementation Details

- Create a prototype
- Prototype uses Angular 1, TypeScript, Webpack
- Websocket / SockJS connection
-

# Implementation Details - Second Phase

- Angular 2/4
- Long Term
- WebRTC

# Environment

- Developer experience
- Angular, Webpack, Karma, Jasmine
- TypeScript, TSLint, Karma reporters

```
if(midiMsg.type !== 'noteoff') {  
  this.soundFontService.sendMidiMessage(midiMsg);  
}
```

TSLint: !== should be === (triple-equals)

PhantomJS 2.1.1 (Linux 0.0.0): Executed 9 of 9 SUCCESS (0.571 secs / 0.668 secs)

```
===== Coverage summary =====  
Statements : 90.76% ( 167/184 )  
Branches   : 25% ( 2/8 )  
Functions  : 74.42% ( 32/43 )  
Lines      : 91.62% ( 153/167 )  
=====
```

# Environment CI

- Karma + Jasmine
- Unit Test reporter
- Coverage reporter
-



# Environment CI

- Jenkins
- JUnit and Cobertura report from Karma

# Project midi-sync-ref

 [add description](#)

[Disable Project](#)



[Coverage Report](#)



[Workspace](#)



[Last Successful Artifacts](#)

 [midi-sync.tar.gz](#)

2.26 [view](#)  
MB



[Recent Changes](#)



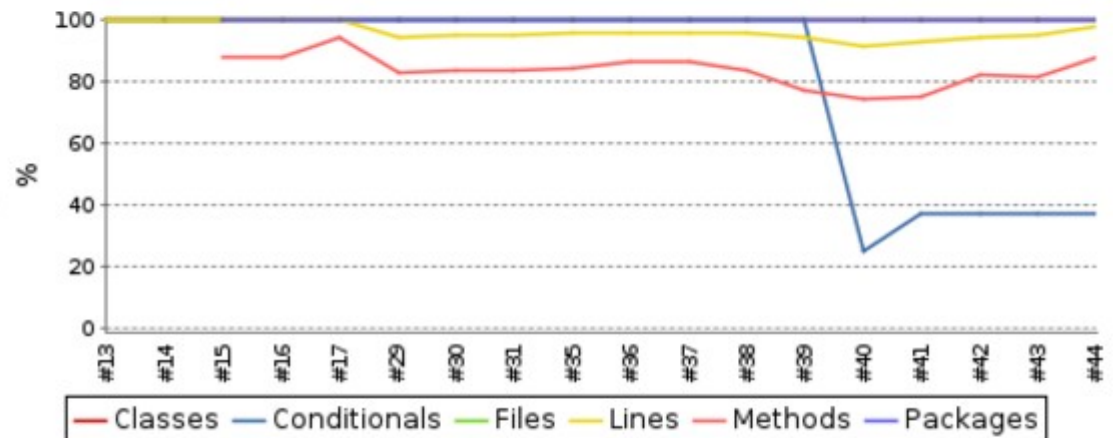
[Latest Test Result](#) (no failures)

## Permalinks

- [Last build \(#44\)](#), 9 hr 3 min ago
- [Last stable build \(#44\)](#), 9 hr 3 min ago
- [Last successful build \(#44\)](#), 9 hr 3 min ago
- [Last failed build \(#34\)](#), 11 hr ago
- [Last unsuccessful build \(#34\)](#), 11 hr ago
- [Last completed build \(#44\)](#), 9 hr 3 min ago

## Code Coverage

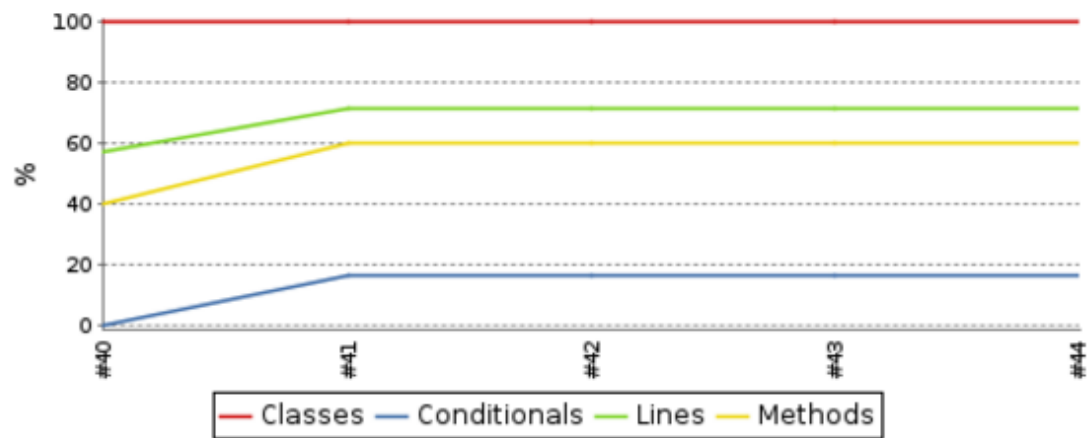
**Packages** 100% **Files** 100% **Classes** 100% **Methods** 88% **Lines** 98%  
**Conditionals** 38%



## Test Result Trend



[\(just show failures\)](#) [enlarge](#)



### File Coverage summary

Name	Classes	Methods	Lines	Conditionals
SoundFontPlayerService.ts	100% <div><div>1/1</div></div>	60% <div><div>3/5</div></div>	71% <div><div>10/14</div></div>	17% <div><div>1/6</div></div>

### Coverage Breakdown by Class

Name	Methods	Lines	Conditionals
<a href="#">SoundFontPlayerService.ts</a>	60% <div><div>3/5</div></div>	71% <div><div>10/14</div></div>	17% <div><div>1/6</div></div>

### Source

```
src/app/services/SoundFontPlayerService.ts
1 1 import * as Soundfont from 'soundfont-player';
2
3 1 import {Injectable} from '@angular/core';
4   import {IMidiMessageConsumer} from '../IMidiMessageConsumer';
5   import {IMidiMessage} from '../MidiMessage';
6
7   @Injectable()
8 1 export class SoundFontPlayerService implements IMidiMessageConsumer {
9 1   instrument = 'acoustic_grand_piano';
10   piano: any;
11 1   initializing = false;
12
```