

# Angular8におけるBootstrap4ベースのng-bootstrapのTimepickerの利用方法

2019.7.14

by *Shuichi Ohtsu*



フルソースコード

<https://github.com/Ohtsu/ng8-bootstrap4-timepicker>

ビデオ解説(日本語)

<https://youtu.be/dgx8Aqe5nJE>

ビデオ解説(英語)

<https://youtu.be/A7qrWNCLBSo>

## プロジェクトの作成方法について

ここでは、Angular8においてBootstrap4ベースのng-bootstrapのTimepickerを利用する方法についてご紹介します。

なお、プロジェクト作成の段階からのBootstrap4ベースのng-bootstrapの設定方法につきましては、すでに、*Angular8におけるBootstrap4環境の設定方法*<https://youtu.be/gF xuS4IZjM>を公開しておりますので、そちらをご覧ください。

今回は、そうした設定済みプロジェクトをGitHubからダウンロードし、さらにそれにTimepicker機能を組

み込む形で進めていきます。

プロジェクトの内容については、そのフルソースコードをご覧ください。

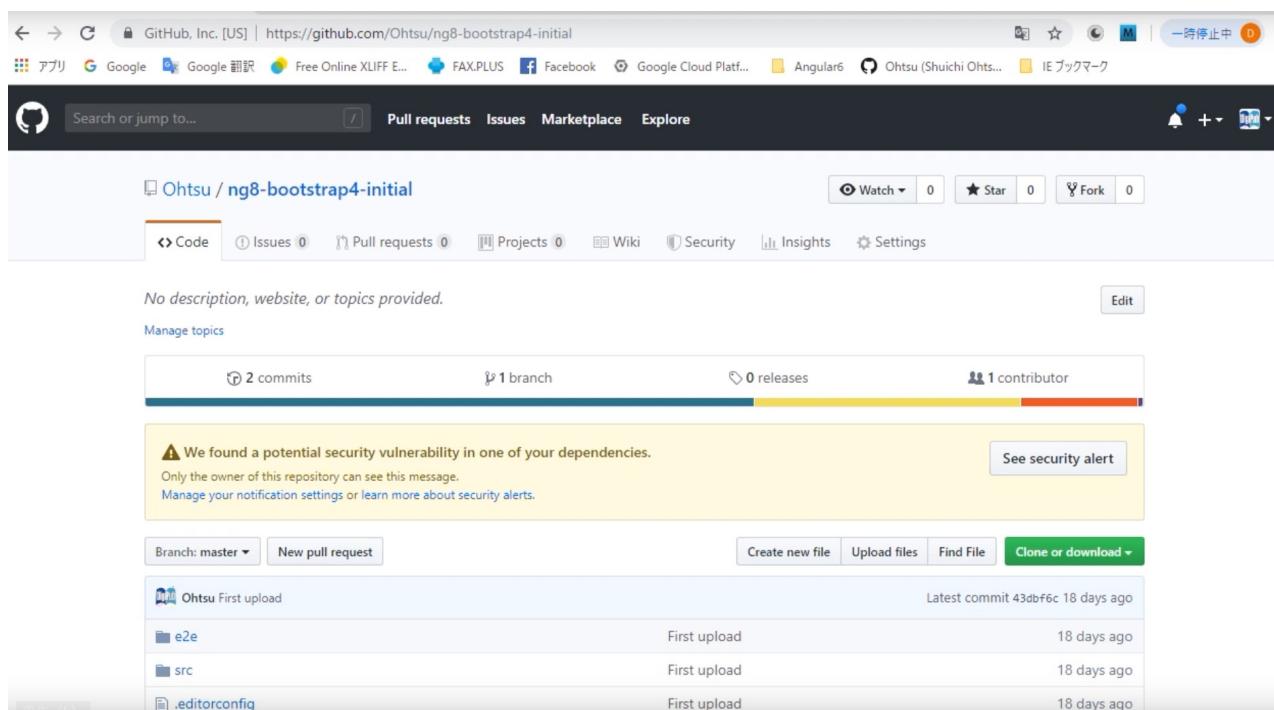
## Bootstrap4を組み込んだプロジェクト・ソースのGitHubからのダウンロード

まず、GitHubからBootstrap4を組み込んだプロジェクトのフルソースをダウンロードします。

そのURLは、

<https://github.com/Ohtsu/ng8-bootstrap4-initial>

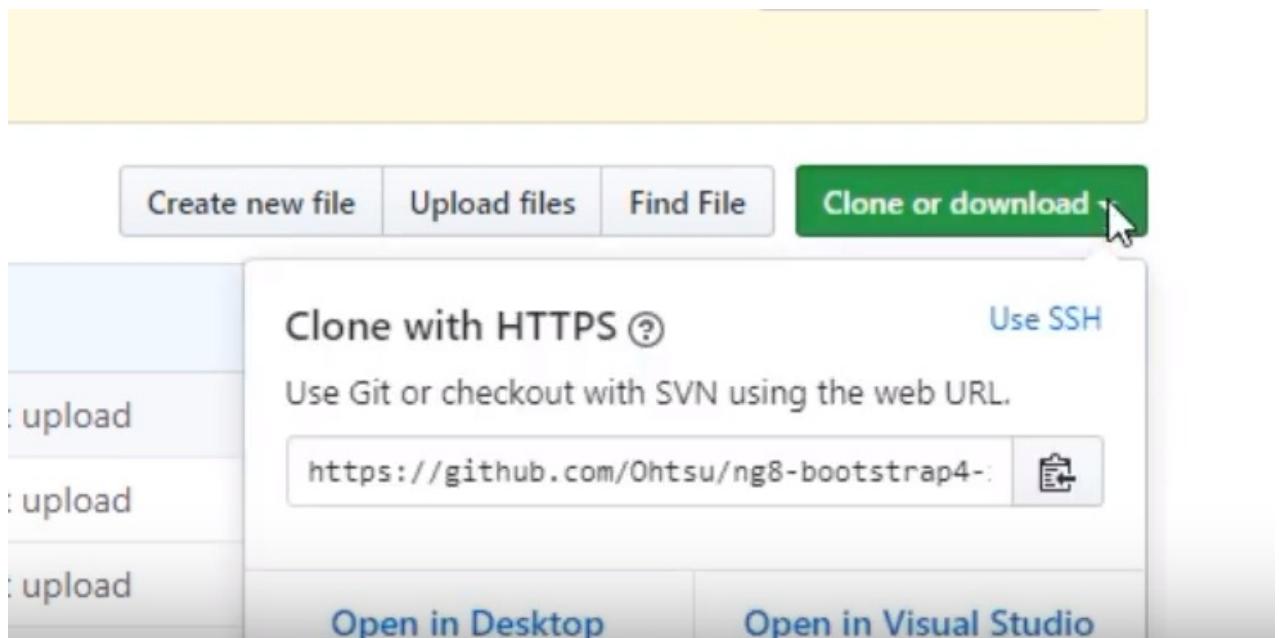
です。



The screenshot shows a GitHub repository page. At the top, there's a navigation bar with links like 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, the repository name 'Ohtsu / ng8-bootstrap4-initial' is displayed. To the right of the name are buttons for 'Watch', 'Star', and 'Fork'. Underneath the repository name, there are tabs for 'Code', 'Issues (0)', 'Pull requests (0)', 'Projects (0)', 'Wiki', 'Security', 'Insights', and 'Settings'. A message 'No description, website, or topics provided.' is shown. Below this, there's a summary bar with metrics: '2 commits', '1 branch', '0 releases', and '1 contributor'. A yellow warning box appears, stating '⚠ We found a potential security vulnerability in one of your dependencies. Only the owner of this repository can see this message. Manage your notification settings or learn more about security alerts.' To the right of the warning box is a 'See security alert' button. At the bottom of the page, there's a table of files uploaded: 'Ohtsu First upload' (Latest commit 43dbf6c 18 days ago), 'e2e' (First upload 18 days ago), 'src' (First upload 18 days ago), and '.editorconfig' (First upload 18 days ago). On the far right, there are buttons for 'Create new file', 'Upload files', 'Find File', and a green 'Clone or download' button.

ページが表示されましたら、右側の緑の*Clone or download*ボタンをクリックします。

するとダウンロード用のアドレスが表示されますので、右側のコピーアイコンをクリックし、そのアドレスをコピーしておきます。



## プロジェクトの生成

コマンド・プロンプトを開き、適当なディレクトリに移動した後、そのディレクトリで、`git clone`に続けて、コピーしたアドレスを貼り付けます。

```
git clone https://github.com/Ohtsu/ng8-bootstrap4-initial.git
```

と入力します。

するとダウンロードが開始されます。

```
ohtsu@Tiger64:/mnt/c/mvprz/test/git$ git clone https://github.com/Ohtsu/ng8-bootstrap4-initial.git
Cloning into 'ng8-bootstrap4-initial'...
```

## Visual Studio Codeの起動

次に、生成されたプロジェクトのディレクトリに入り、

```
cd ng8-bootstrap4-initial
```

Visual Studio Codeを起動します。

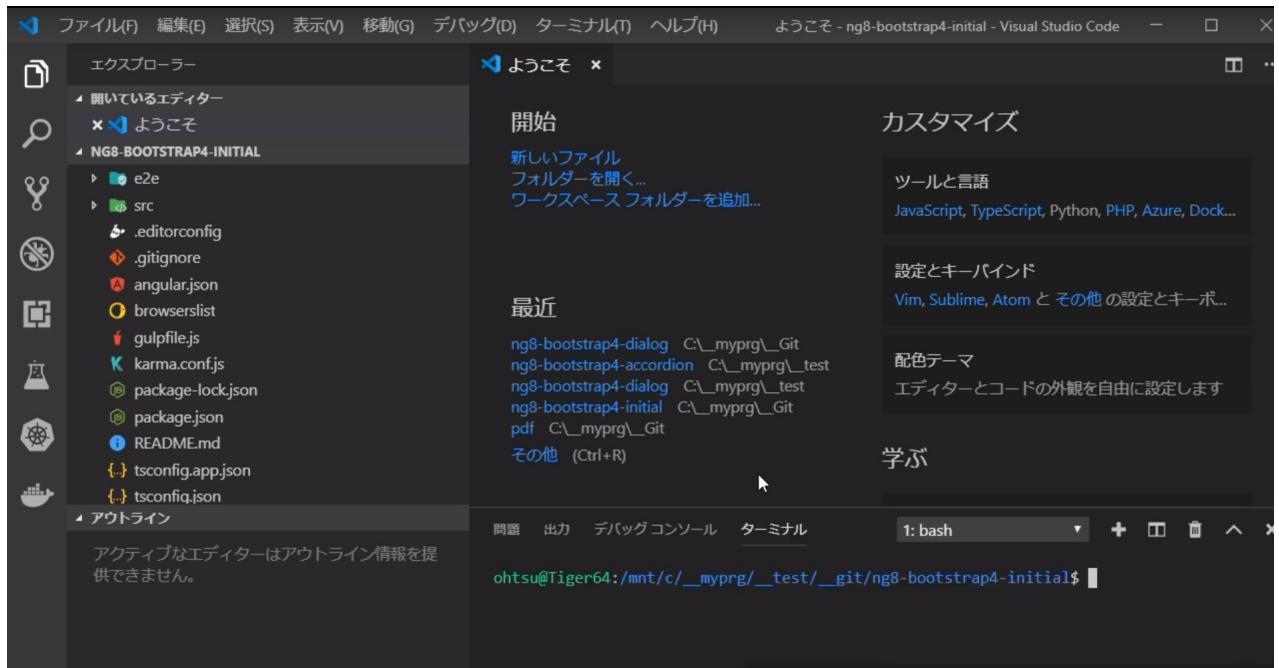
```
code .
```

と入力します。

```
$ cd ng8-bootstrap4-initial  
$ code .
```

VSCodeが起動しましたら、

ctrl+@キーを押し、ターミナル・ウィンドウを開きます。



そして、

```
npm i
```

と入力し、プロジェクトに必要なライブラリをインストールします。

このプロセスには若干時間を要します。

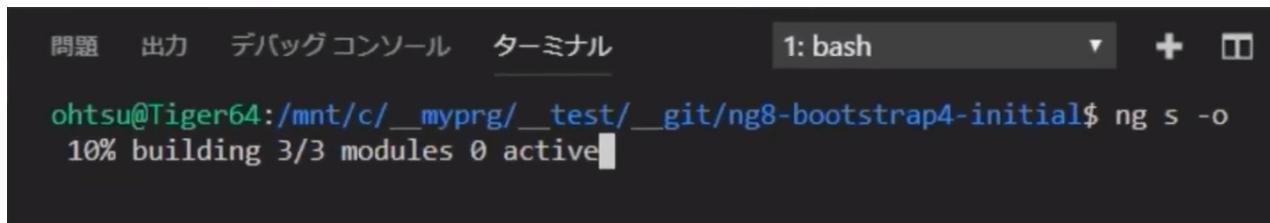
```
問題 出力 デバッグコンソール ターミナル  
1: bash  
ohotsu@Tiger64:/mnt/c/_myprg/_test/_git/ng8-bootstrap4-initial$ npm i  
[redacted] \ extract:@webassemblyjs/helper-wasm-bytecode: sill extract @web
```

## プロジェクトの起動の確認

次に、プロジェクトが起動できるかどうかを確認します。

```
ng s -o
```

と入力します。

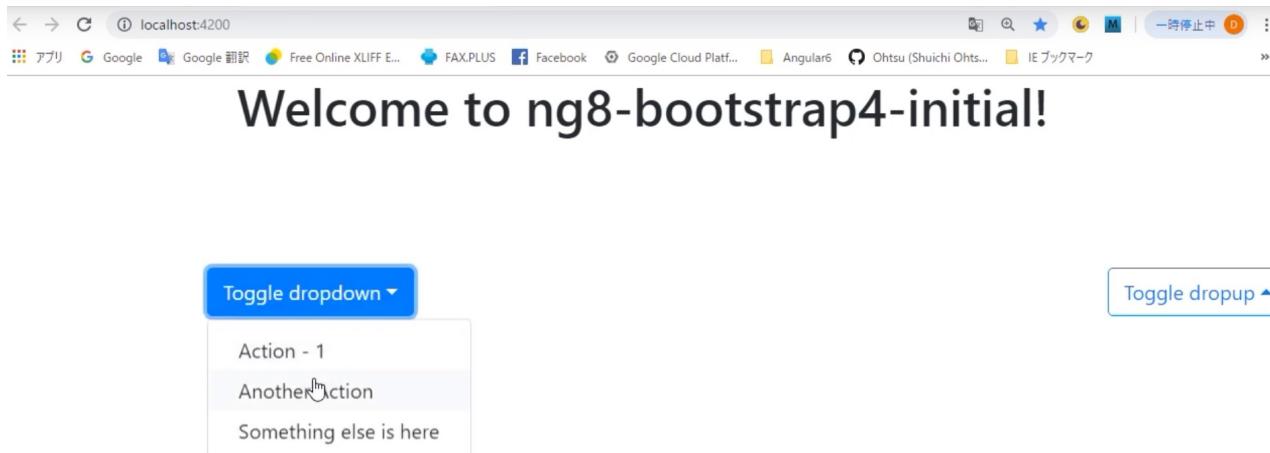


```
問題 出力 デバッグコンソール ターミナル 1: bash ▾ + ■
```

```
ohtsu@Tiger64:/mnt/c/_myprg/_test/_git/ng8-bootstrap4-initial$ ng s -o
```

```
10% building 3/3 modules 0 active
```

ブラウザが自動的に開き、ドロップダウン及びドロップアップが表示されます。



OKです。

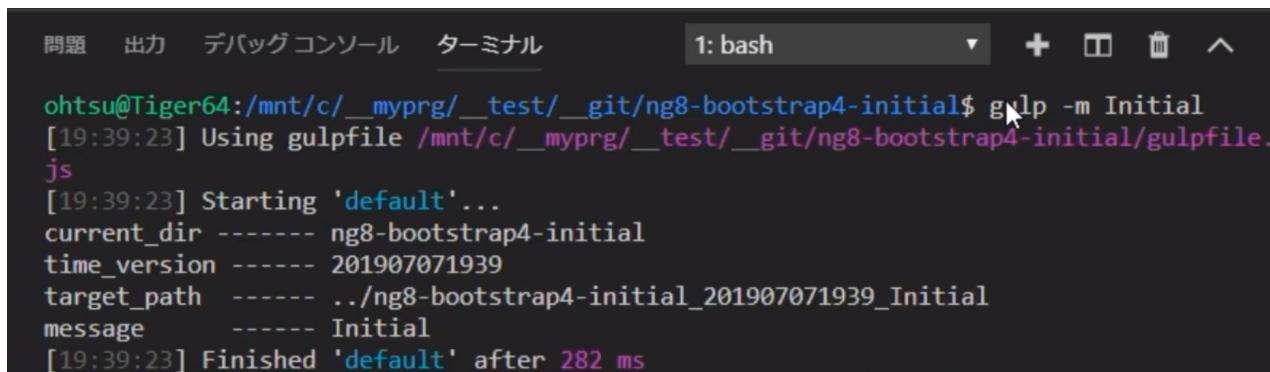
## Gulpによるバックアップの作成

この段階で、変更前ソースのバックアップを取っておきます。但し、関連ライブラリまでバックアップをする必要はないので、`node_modules`以下のファイルは除外することにします。

このようなバックアップを行う`gulpfile.js`をすでにプロジェクトに含めてありますので、それを実行します。

```
gulp -m Initial
```

と入力します。`-m`オプション以下の文字は、作成したプロジェクト名に追加する文字列です。



```
問題 出力 デバッグコンソール ターミナル 1: bash ▾ + ■ ▲
```

```
ohtsu@Tiger64:/mnt/c/_myprg/_test/_git/ng8-bootstrap4-initial$ gulp -m Initial
```

```
[19:39:23] Using gulpfile /mnt/c/_myprg/_test/_git/ng8-bootstrap4-initial/gulpfile.js
```

```
[19:39:23] Starting 'default'...
```

```
current_dir ----- ng8-bootstrap4-initial
```

```
time_version ----- 201907071939
```

```
target_path ----- ../ng8-bootstrap4-initial_201907071939_Initial
```

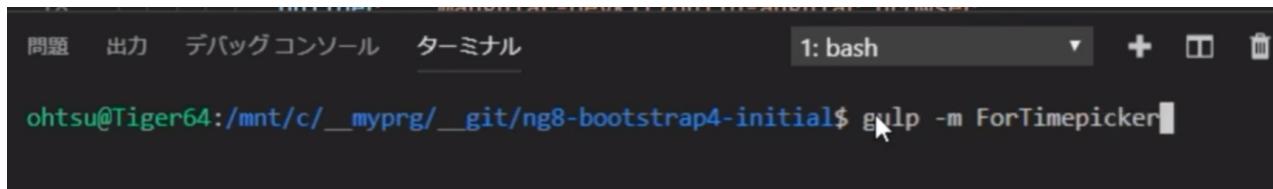
```
message ----- Initial
```

```
[19:39:23] Finished 'default' after 282 ms
```

次に、Timepickerテスト用のプロジェクトも作成しておきます。

```
gulp -m ForTimepicker
```

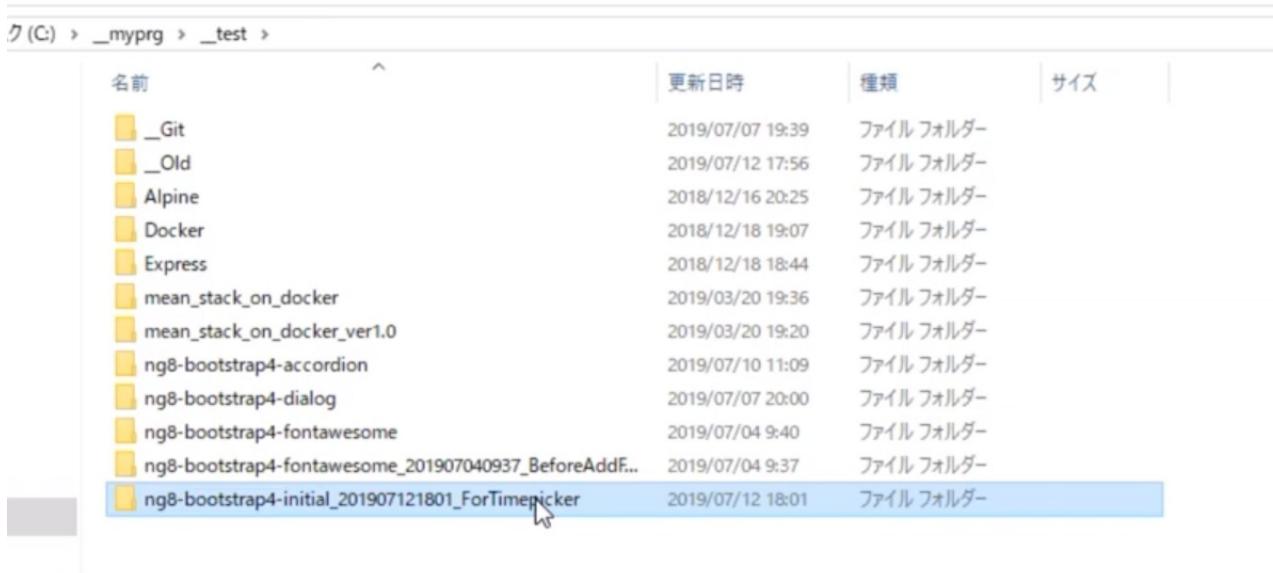
と入力します。



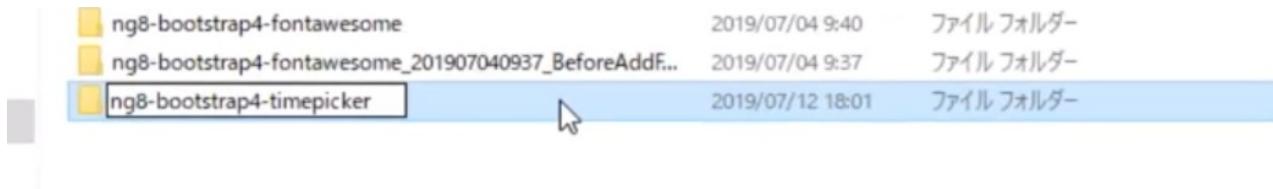
```
問題 出力 デバッグコンソール ターミナル 1: bash + ■
```

```
ohtsu@Tiger64:/mnt/c/_myprg/_git/ng8-bootstrap4-initial$ gulp -m ForTimepicker
```

次にファイルエクスプローラーを開き、作成したTimepickerテスト用のプロジェクトをディレクトリごとコピーし、適当なディレクトリ内にペーストします。



そして、ディレクトリ名をng8-bootstrap4-timepickerに変更しておきます。



次に、コマンド・プロンプトを開き、このディレクトリに移動します。

そして

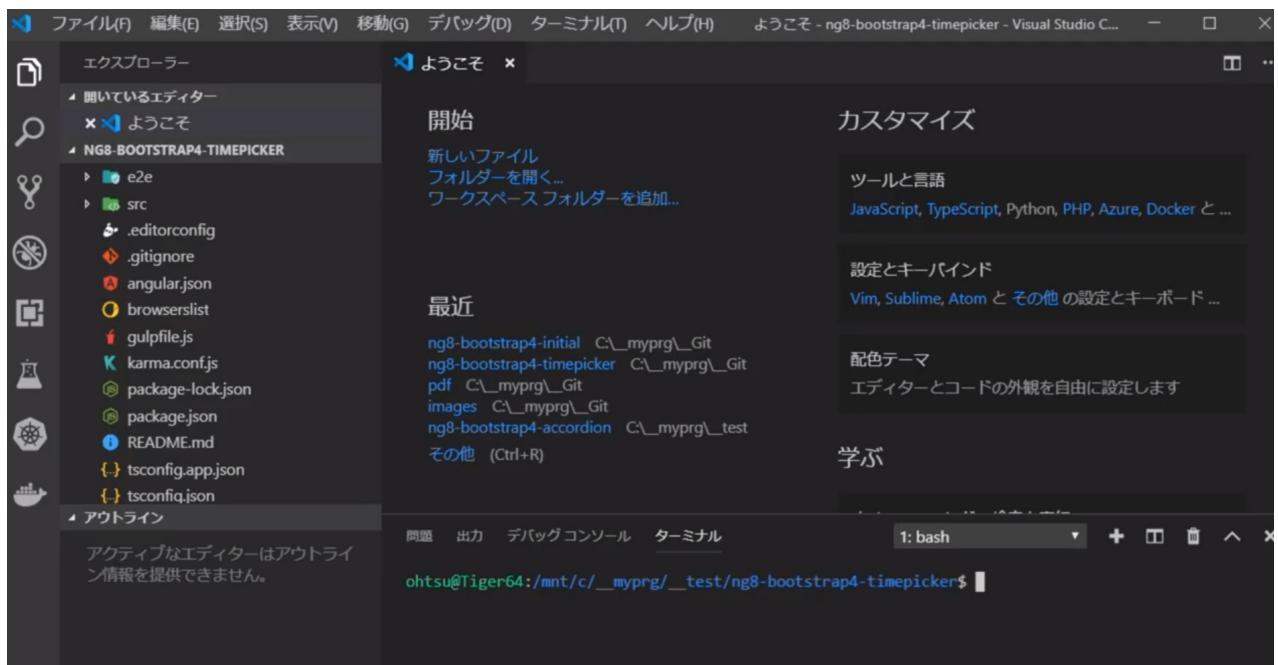
```
code .
```

と入力して、VSCodeを開きます。

```
$ cd ng8-bootstrap4-timepicker  
/ng8-bootstrap4-timepicker$ code .
```

VSCodiumが起動しましたら、

ctrl+@キーを押し、ターミナル・ウィンドウを開きます。



## プロジェクト名の変更

次に、プロジェクトの内容を変更していきます。

まず、プロジェクト名を変更します。

*package.json*ファイルを開くと、プロジェクト名が`ng8-bootstrap4-initial`になっているのがわかります。これを`ng8-bootstrap4-timepicker`に変更します。

左の検索アイコンをクリックし、プロジェクト全体のソースを対象として置換を行います。

The screenshot shows a terminal window with the following content:

```
ng8-bootstrap4-initial Aa Abi *
ng8-bootstrap4-timepicker F5
7 個のファイルに 16 件の結果 ...
  ↳ angular.json 8
    "ng8-bootstrap4-initialNg8-bootstrap...
    "outputPath": "dist/ng8-bootstrap4-in...
    "browserTarget": "ng8-bootstrap4-init...
    "browserTarget": "ng8-bootstrap4-init...
    "browserTarget": "ng8-bootstrap4-init...
    "devServerTarget": "ng8-bootstrap4-in...
    "devServerTarget": "ng8-bootstrap4-in...
    "defaultProject": "ng8-bootstrap4-init...
  ↳ app.e2e-spec.ts 1
    getTitleText().toEqual('Welcome to ng...
  ↳ karma.conf.js 1
    join(_dirname, './coverage/ng8-boots...
  ↳ package-lock.json 1
    "name": "ng8-bootstrap4-initialNg8-b...
  ↳ package.json 1
```

package.json

```
1  {
2    "name": "ng8-bootstrap4-initial",
3    "version": "0.0.0",
4    "scripts": {
5      "ng": "ng",
6      "start": "ng serve",
7      "build": "ng build",
8      "test": "ng test",
9      "lint": "ng lint",
10     "e2e": "ng e2e"
11   },
12   "private": true,
13   "dependencies": {
14     "@angular/animations": "~8.0.0",
15     "@angular/common": "~8.0.0",
16     "@angular/compiler": "~8.0.0",
17     "@angular/core": "~8.0.0",
18     "@angular/forms": "~8.0.0"
19   }
20 }
```

問題 出力 デバッグコンソール ターミナル 1: bash

```
ohtsu@Tiger64:/mnt/c/_myprg/_test/ng8-bootstrap4-timepicker$
```

次に、パスカルケース、すなわち-ではなく大文字で区切っているものも存在しますので、`Ng8Bootstrap4Initial`を`Ng8Bootstrap4Timepicker`に変更しておきます。

The screenshot shows a terminal window with the following content:

```
Ng8Bootstrap4Initial Aa Abi *
Ng8Bootstrap4Timepicker F5
2 個のファイルに 2 件の結果 ...
  ↳ README.md 1
    # Ng8Bootstrap4InitialNg8Bootstrap4...
  ↳ index.html src 1
    <title>Ng8Bootstrap4InitialNg8Boots...
```

次に

```
npm i
```

と入力し、再度ライブラリをインストールします。

```
問題 22 出力 デバッグコンソール ターミナル 1: bash +  
ohtsu@Tiger64:/mnt/c/_myprg/_test/ng8-bootstrap4-timepicker$ npm i  
[██████████] | diffTrees: sill install generateActionsToTake
```

このプロセスには若干時間を要します。

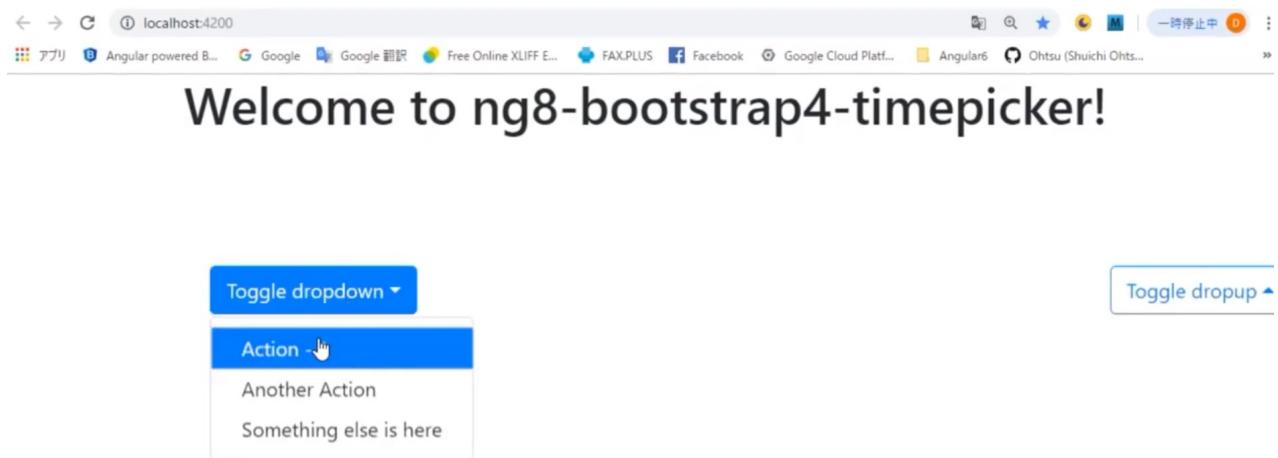
また、プロジェクトが起動できるかもチェックします。

```
ng s -o
```

と入力します。

```
問題 6 出力 デバッグコンソール ターミナル 1: bash ▾  
ohtsu@Tiger64:/mnt/c/_myprg/_test/ng8-bootstrap4-timepicker$ ng s -o  
10% building 3/3 modules 0 active
```

ブラウザが自動的に開き、ドロップダウン及びドロップアップが表示されます。



OKです。

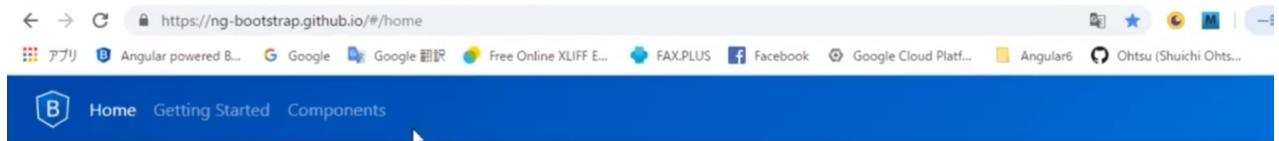
## Timepickerを追加

次に`ng-bootstrap`のサイトを開き、Timepickerのサンプル・プログラムを参照します。

そのURLは、

<https://ng-bootstrap.github.io/#/home>

です。



## Bootstrap widgets

### The angular way

Angular widgets built from the ground up using only Bootstrap 4 CSS with APIs designed for the Angular ecosystem

No dependencies on 3rd party JavaScript.

[Demo](#)

[Installation](#)

Currently at v5.0.0

ここで、[Demo](#)ボタンをクリックし、左側のメニューから[Timepicker](#)をクリックします。

そして[Global configuration of timepickers](#)を表示します。  
この設定を取り込むことにします。

Popover  
Progressbar  
Rating  
Table  
Tabset  
**Timepicker**  
Toast  
Tooltip  
Typeahead  
  
Positioning

### 🔗 Global configuration of timepickers

[Code](#)

[StackBlitz](#)

This timepicker uses customized default values.

13 : 30 : 00

まず、右上の[Code](#)ボタンをクリックし、そのソースを表示します。

まずHTML文をコピーします。

# Global configuration of timepickers

[Code](#)[StackBlitz](#)

timepicker-config.html

timepicker-config.ts

```
<p>This timepicker uses customized default values.</p>
<ngb-timepicker [(ngModel)]="time"></ngb-timepicker>
```

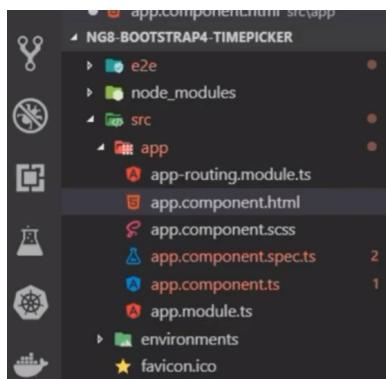
This timepicker uses customized default values.

13 : 30 : 00

VSCODEに戻り、src/app内のapp.component.htmlファイルを開きます。

コピーしたHTMLをここに貼り付けます。

また、プルダウン及びプルアップ関連の部分をコメントアウトします。



```
23 <div class="col text-right">
24   <div ngbDropdown placement="top-right" class="d-inline-block">
25     <button class="btn btn-outline-primary" id="dropdownBasic2" ngbDropdownToggle>
26       <div ngbDropdownMenu aria-labelledby="dropdownBasic2">
27         <button ngbDropdownItem>Action - 1</button>
28         <button ngbDropdownItem>Another Action</button>
29         <button ngbDropdownItem>Something else is here</button>
30       </div>
31     </div>
32   </div>
33 </div>
34
35 <p>This timepicker uses customized default values.</p>
36
37 <ngb-timepicker [(ngModel)]="time"></ngb-timepicker>
38 <router-outlet></router-outlet>
```

# Global configuration of timepickers

</> Code

StackBlitz

timepicker-config.html

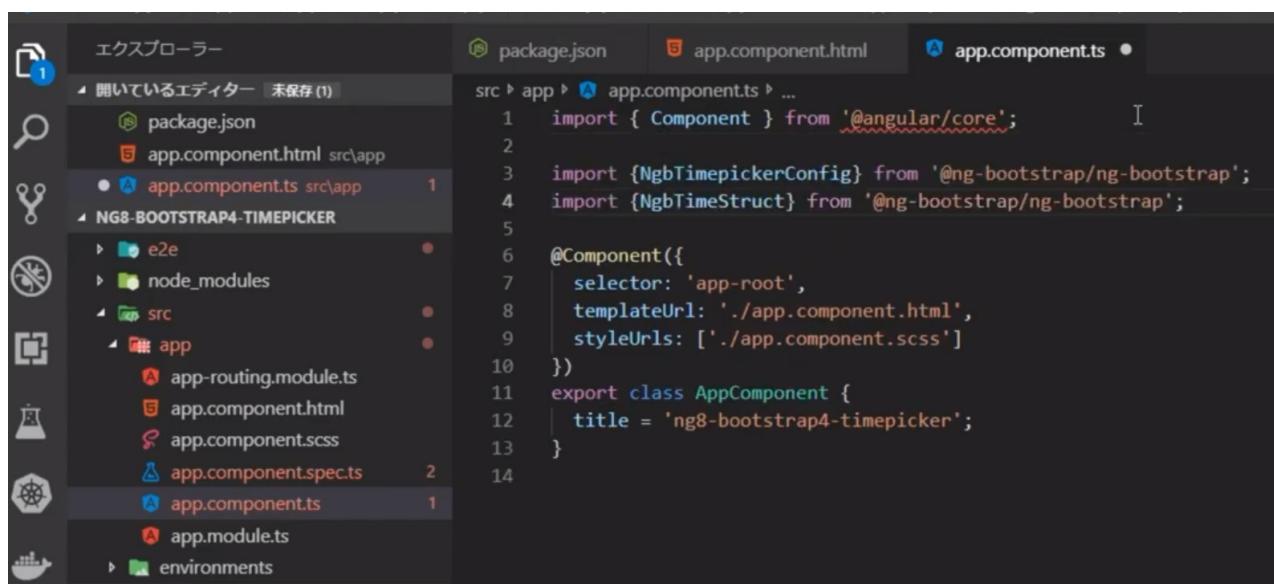
timepicker-config.ts

```
import {Component} from '@angular/core';
import {NgbTimepickerConfig} from '@ng-bootstrap/ng-bootstrap';
import {NgbTimeStruct} from '@ng-bootstrap/ng-bootstrap';

@Component({
  selector: 'ngbd-timepicker-config',
  templateUrl: './timepicker-config.html',
  providers: [NgbTimepickerConfig] // add NgbTimepickerConfig to the component
})
export class NgbdTimepickerConfig {
  time: NgbTimeStruct = {hour: 13, minute: 30, second: 0};
```

ブラウザに戻り、TypeScriptのファイルを開きます。

まず追加が必要なimport部分をピーします。



The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure with files like package.json, app.component.html, app.component.ts, app-routing.module.ts, app.component.spec.ts, app.module.ts, and environments.
- Code Editor:** Displays the content of app.component.ts. The first few lines of code are:

```
1 import { Component } from '@angular/core';
2
3 import {NgbTimepickerConfig} from '@ng-bootstrap/ng-bootstrap';
4 import {NgbTimeStruct} from '@ng-bootstrap/ng-bootstrap';
```
- Top Bar:** Shows tabs for package.json, app.component.html, and app.component.ts. The app.component.ts tab is active.

VSCodeに戻り、src/app内のapp.component.tsファイルを開きます。

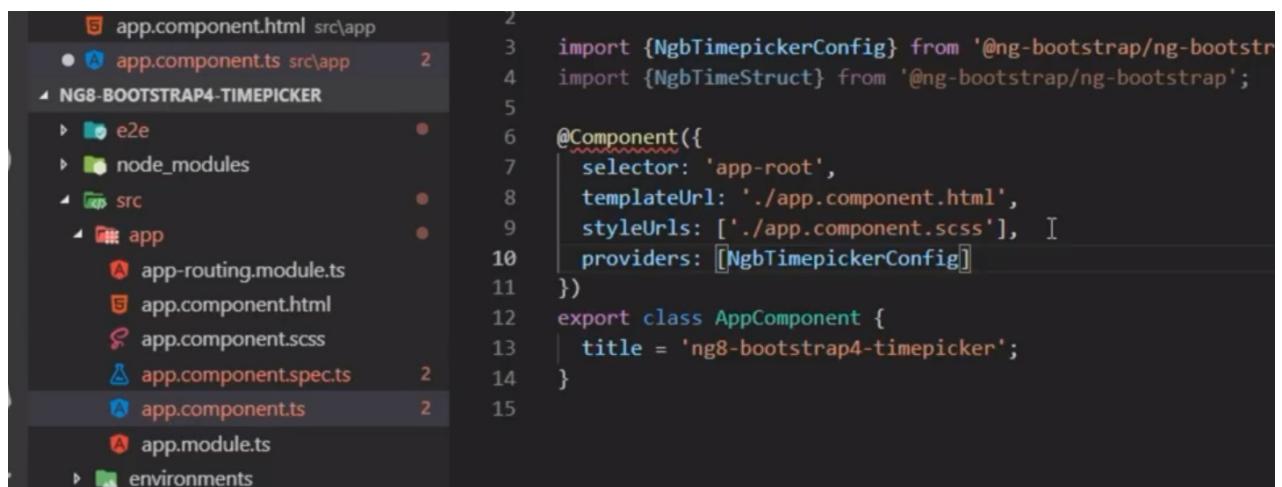
```
timepicker-config.ts
```

```
import {Component} from '@angular/core';
import {NgbTimepickerConfig} from '@ng-bootstrap/ng-bootstrap';
import {NgbTimeStruct} from '@ng-bootstrap/ng-bootstrap';

@Component({
  selector: 'ngbd-timepicker-config',
  templateUrl: './timepicker-config.html',
  providers: [NgbTimepickerConfig] // add NgbTimepickerConfig to the component's provider list
})
export class NgbdTimepickerConfig {
  time: NgbTimeStruct = {hour: 13, minute: 30, second: 0};

  constructor(config: NgbTimepickerConfig) {
    // customize default values of ratings used by this component tree
    config.seconds = true;
    config.spinners = false;
  }
}
```

コピーしたimport部分をここに貼り付けます。



```
app.component.html src\app
● app.component.ts src\app 2
NG8-BOOTSTRAP4-TIMEPICKER
  e2e
  node_modules
  src
    app
      app-routing.module.ts
      app.component.html
      app.component.scss
      app.component.spec.ts 2
      app.component.ts 2
      app.module.ts
    environments
```

```
import {NgbTimepickerConfig} from '@ng-bootstrap/ng-bootstrap';
import {NgbTimeStruct} from '@ng-bootstrap/ng-bootstrap';
5
6 @Component({
7   selector: 'app-root',
8   templateUrl: './app.component.html',
9   styleUrls: ['./app.component.scss'],
10  providers: [NgbTimepickerConfig]
11 })
12 export class AppComponent {
13   title = 'ng8-bootstrap4-timepicker';
14 }
```

ブラウザに戻り、providers部分をコピーします。

```
@Component({
  selector: 'ngbd-timepicker-config',
  templateUrl: './timepicker-config.html',
  providers: [NgbTimepickerConfig] // add NgbTimepickerConfig to the component's provider list
})
export class NgbdTimepickerConfig {
  time: NgbTimeStruct = {hour: 13, minute: 30, second: 0};
```

VSCodeに戻り、*app.component.ts*ファイルを開き、providers部分をここに貼り付けます。

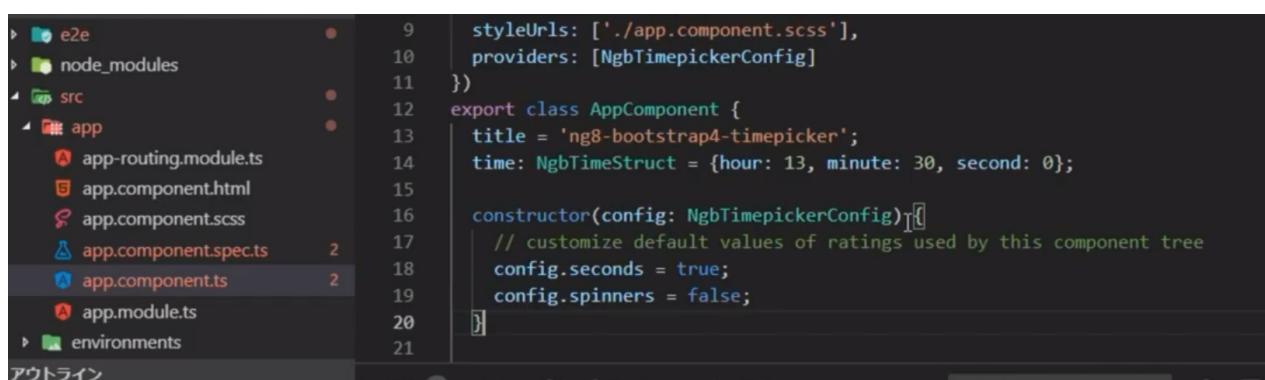
```
5
6  @Component({
7    selector: 'app-root',
8    templateUrl: './app.component.html',
9    styleUrls: ['./app.component.scss'],
10   providers: [NgbTimepickerConfig]
11 })
```

ブラウザに戻り、クラス内の主要部分をコピーします。

```
@Component({
  selector: 'ngbd-timepicker-config',
  templateUrl: './timepicker-config.html',
  providers: [NgbTimepickerConfig] // add NgbTimepickerConfig to the component's provider list
})
export class NgbdTimepickerConfig {
  time: NgbTimeStruct = {hour: 13, minute: 30, second: 0};

  constructor(config: NgbTimepickerConfig) {
    // customize default values of ratings used by this component tree
    config.seconds = true;
    config.spinners = false;
  }
}
```

VSCodeに戻り、*app.component.ts*ファイルを開き、ここに貼り付けます。



The screenshot shows a code editor with the following file structure on the left:

- e2e
- node\_modules
- src
  - app
    - app-routing.module.ts
    - app.component.html
    - app.component.scss
    - app.component.spec.ts
    - app.component.ts
  - environments

The right side displays the code for `AppComponent`:

```
styleUrls: ['./app.component.scss'],
providers: [NgbTimepickerConfig]
})
export class AppComponent {
  title = 'ng8-bootstrap4-timepicker';
  time: NgbTimeStruct = {hour: 13, minute: 30, second: 0};

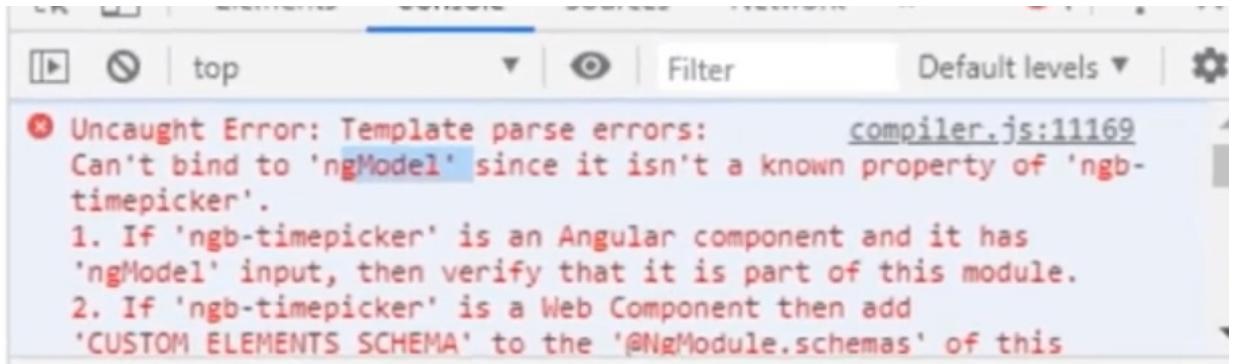
  constructor(config: NgbTimepickerConfig) {
    // customize default values of ratings used by this component tree
    config.seconds = true;
    config.spinners = false;
  }
}
```

## Timepickerの表示

ここで、ブラウザに戻り、<http://localhost:4200>を開きます。

しかしTimepickerは表示されません。エラーが発生しているようです。

Ctrl+Shift+i キーでコンソール情報を表示します。ngModelが原因のようです。ngModelを利用するためには、FormsModuleをインポートしておく必要があります。



## FormsModuleのインポート

VSCODEに戻り、app.module.tsを開き、FormsModuleをインポートします。

```
src > app > A app.module.ts > ...
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { FormsModule } from '@angular/forms';
5
6 import { AppRoutingModule } from './app-routing.module';
7
8 import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
9
10 import { AppComponent } from './app.component';
11
12 @NgModule({
13   declarations: [
14     AppComponent
15   ],
16   imports: [
17     BrowserModule,
```

再度、ブラウザに戻りますと、エラーがなくなり、Timepickerが表示されました。

This timepicker uses customized default values.

13 : 30 : 00

Angular is running in the development mode. Call enableProdMode() to enable the production mode.

OKです。

## Spinnerの表示

しかし、このままですると、数値入力が面倒なので、Spinnerを表示することにします。

VSCodeに戻り、*app.component.ts*ファイルを開き、

```
config.spinners = true
```

と変更します。

```
12  export class AppComponent {
13    title = 'ng8-bootstrap4-timepicker';
14    time: NgbTimeStruct = {hour: 13, minute: 30, second: 0};
15
16    constructor(config: NgbTimepickerConfig) {
17      // customize default values of ratings
18      config.seconds = true;
19      config.spinners = true;
20    }
21
22 }
```

VS Code のコード補完メニューで、config.spinners = true; の下に表示される候補肢（true, truncate, truncateSync）を確認できます。

ブラウザに戻り、表示を確認すると、Spinnerが表示されるようになりました。

# Welcome to ng8-bootstrap4-timepicker!

This timepicker uses customized default values.



OKです。

## 秒の非表示

次に、秒までは表示する必要はないので、秒の欄を非表示にします。

VSCodeに戻り、*app.component.ts*ファイルを開き、

```
config.seconds = false
```

に変更します。

```
16  constructor(config: NgbTimepickerConfig) {  
17    // customize default values of ratings us  
18    config.seconds = false;  
19    config.spinners = true; false  
20  }  
21
```

ブラウザに戻り、表示を確認すると、秒の欄が非表示になりました。

# Welcome to ng8-bootstrap4-timepicker!

This timepicker uses customized default values.



OKです。

## AM/PMの表示

次にAM/PMの区別をした表示にしてみます。

VSCodeに戻り、*app.component.ts*ファイルを開き、

```
config.meridian = true
```

を追加します。

```
15
16  constructor(config: NgbTimepickerConfig) {
17    // customize default values of ratings used by this component tree
18    config.meridian = true;           I
19    config.seconds = false;          true
20    config.spiners = true;          truncate
21  }                                truncateSync
```

ブラウザに戻り、表示を確認すると、AM/PMのボタンが追加されました。

# Welcome to ng8-bootstrap4-timepicker!

This timepicker uses customized default values.



## 取得した時刻データの表示

しかし、このままだと時刻データが本当に取得できているのかがわかりません。そこで取得時刻データを画面上に表示することにします。

VSCodeに戻り、*app.component.html*ファイルを開き、取得時刻データを表示するように変更します。

具体的には、

```
<pre>Selected time: {{time | json}} </pre>
```

という一行を追加します。

```
10
11  <p>This timepicker uses customized default values.</p>
12
13  <ngb-timepicker [(ngModel)]="time"></ngb-timepicker> [
14    <pre>Selected time: {{time | json}}</pre>
15  <router-outlet></router-outlet>
16
```

ブラウザに戻り、表示を確認すると、取得時刻データが追加されました。

## Welcome to ng8-bootstrap4-timepicker!

This timepicker uses customized default values.



```
Selected time: {
  "hour": 13,
  "minute": 30,
  "second": 0
}
```

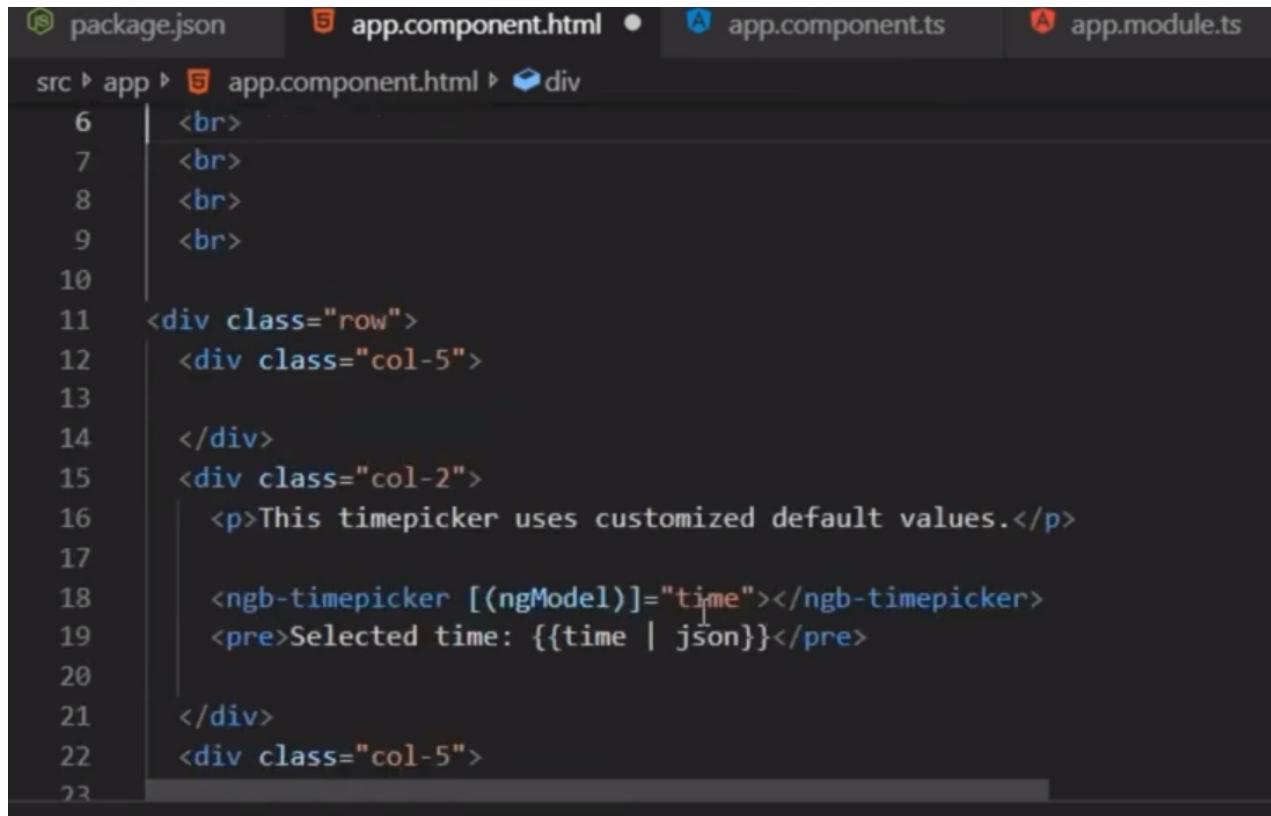
OKです。

## 表示スタイルの修正

しかし、このままでするとTimepickerが左寄せになっており、取得時刻データとのバランスがよくないので、そこを修正していきます。

まず、一行を5対2対5の比率で区切り、中央に表示するようにしてみます。

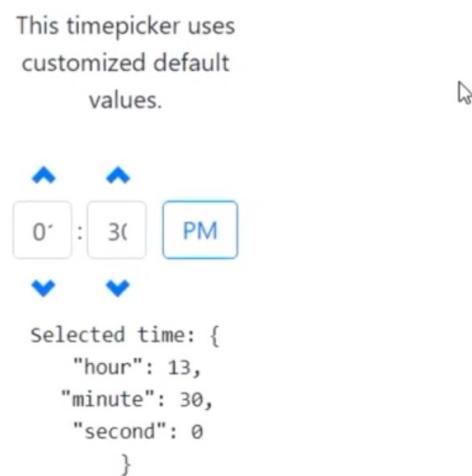
VSCodeに戻り、*app.component.html*ファイルを開き、divタグを追加していきます。



```
package.json app.component.html app.component.ts app.module.ts
src > app > app.component.html > div
6   <br>
7   <br>
8   <br>
9   <br>
10
11  <div class="row">
12    <div class="col-5">
13
14    </div>
15    <div class="col-2">
16      <p>This timepicker uses customized default values.</p>
17
18      <ngb-timepicker [(ngModel)]="time"></ngb-timepicker>
19      <pre>Selected time: {{time | json}}</pre>
20
21    </div>
22    <div class="col-5">
```

ブラウザに戻り、表示を確認しますと、数値データの表示が欠けています。

## Welcome to ng8-bootstrap4-timepicker!



そこで中央の表示幅を広げるために、一行を4対4対4の比率で区切ることにします。

VSCodeに戻り、*app.component.html*ファイルを開き、比率を変更します。

```
11  <div class="row">
12    <div class="col-4">
13
14    </div>
15    <div class="col-4">
16      <p>This timepicker uses customized default values.</p>
17
18      <ngb-timepicker [(ngModel)]="time"></ngb-timepicker>
19      <pre>Selected time: {{time | json}}</pre>
20
21    </div>
22    <div class="col-4">
23
24    </div>
25
26  </div>
```

ブラウザに戻り、表示を確認します。

## Welcome to ng8-bootstrap4-timepicker!

This timepicker uses customized default  
values.



OKです。

## Reference

- "Angular8におけるBootstrap4環境の設定方法(Youtube)",  
<https://youtu.be/gF xuSi4IZjM>

- "How to set up Bootstrap 4 environment in Angular 8",  
<https://youtu.be/NoSkIVv5o8>
- "Angular8におけるBootstrap4環境の設定方法(Qiita)",  
<https://qiita.com/SOhtsu/items/62253bb2309c9a4809fd>
- "Angular 7 Bootstrap 4 UIs: ng-bootstrap and ngx-bootstrap [Part 1]",  
<https://www.techiediaries.com/angular-bootstrap-ui/>
- "Angular5, Angular6, Angular7 Custom Library: Step-by-step guide",  
<https://www.udemy.com/angular5-custom-library-the-definitive-step-by-step-guide/>
- "Angular5, Angular6, Angular7用 カスタムライブラリの作成: 完全ステップ・バイ・ステップ・ガイド",  
<https://www.udemy.com/angular5-1/>