

## Angular5.0 バージョンアップ対応手順書

「Angular2 によるモダン Web 開発 TypeScript を使った基本プログラミング」(以下、本書と記載します)の対象である Angular2 は、2017 年 11 月 1 日にメジャーバージョンアップを行い Angular5.0 になりました。Angular5.0 への対応をまとめたのがこの手順書です。

### Angular のバージョンアップと本書の対応

Angular は、半年間隔でメジャーバージョンアップを予定しています(表 1)。これからは、Angular4 ではなく一般的なソフトウェアと同じ様に「Angular」と呼ばれるようになります。本書は、メジャーバージョンアップに継続して対応し、手順書とサンプルアプリの完成版ファイルを公開する予定です。

表 1. リリース予定表

時期	バージョン	後方互換
2016 年 9 月 14 日	2.0.0	なし
2017 年 3 月 23 日	4.0.0	2.x
2017 年 11 月 1 日	5.0.0	4.x
2018 年 3-4 月(予定)	6.0.0	5.x
2018 年 9-10 月(予定)	7.0.0	6.x

\*注) バージョン 3 は欠番です。

### Angular5.0 の互換性

表 1 に記載のとおり、Angular5.0 は後方互換があります。つまり Angular4.x で作成したアプリは Angular5.0 で基本的に動作可能です。しかし、Angular5.0 で非推奨となった API や Angular2 または 4 では Production 用ビルドで不具合が発生する JavaScript ライブラリがありますので、変更が発生します。

### 今回対応する開発環境

Angular	5.0.0
Angular CLI	1.5.0

Bootstrap	3.3.7
ngx-bootstrap	2.0.0-beta.8
Intl	1.2.5
WebStorm	2017.2
Google Chrome	62.0
Node.js	8.5.0
git for Windows	2.15.0

### 開発環境バージョンアップの影響

1 章	変更なし
2 章	変更なし
3 章	ソフトウェアのバージョン指定 新規プロジェクトの作成手順
4 章	テストプログラムのインストール
5 章	新規プロジェクトの作成手順 Bootstrap CSS の適用場所 マニフェストファイルの設定
6 章	新規プロジェクトの作成手順 追加モジュールのバージョン Bootstrap CSS の適用場所 追加モジュールの登録記述
付録	変更なし

### 3 章の変更

3.2 ダウンロード URL やインストールコマンドが変更になります。

3.2.2-①Node.js ダウンロードサイト
<a href="https://nodejs.org/dist/v8.5.0/">https://nodejs.org/dist/v8.5.0/</a>

3.2.3-①Git for Windows ダウンロードサイト
<a href="https://github.com/git-for-windows/git/releases/tag/v2.15.0.windows.1">https://github.com/git-for-windows/git/releases/tag/v2.15.0.windows.1</a>

### 3.2.4-① Angular CLI のインストールコマンド

```
npm install @angular/cli@1.5.0
```

パスの自動設定が行われますので、3.2.4-⑤⑥の手順は不要です。

#### 3.3.1.1 新規プロジェクト作成手順の変更

¥User¥user01(ユーザーフォルダ)は環境で異なりますので、利用する環境に合わせて読み替えてください。

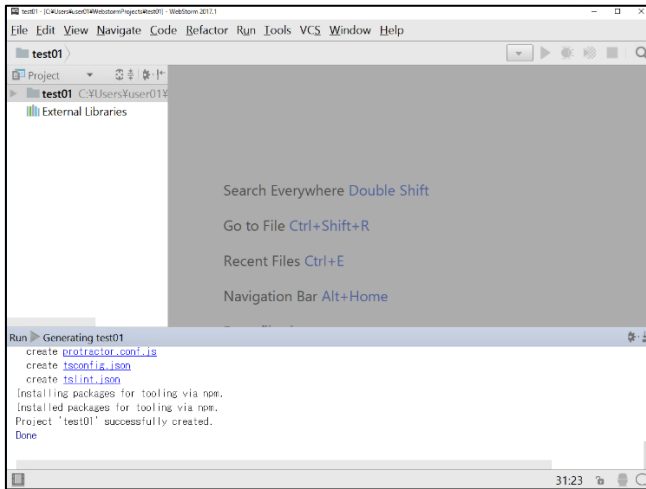
本書では、WebStorm のプロジェクト作成ウィザードを使いプロジェクト作成していますが、新規プロジェクトをダウンロードする方法に変更します。開発環境のバージョンアップの影響を受けず、確実な動作が期待できます。

①～⑥の手順を、下記と差し替えます

1. 本書ダウンロードサイトから完成版ファイルをダウンロードします。
2. zip ファイルを展開します。完成版フォルダ内に自己展開ファイルが 4 個生成します。
  - ① test01.exe 生成済のプロジェクトのひな型
  - ② testProgram.exe 4 章 テストプログラム
  - ③ calcApp.exe 5 章 複利計算アプリ
  - ④ cloudApp.exe 6 章 海外ツアー検索アプリ
3. test01.exe を C:¥Users¥user01¥フォルダにコピーします。
4. C:¥Users¥user01¥test01.exe をダブルクリックして自己展開します。
5. test01 プロジェクトフォルダが生成します。
6. WebStorm を起動します。スタート画面中央の「Open」をクリックします。



7. スタート画面が開かずにプロジェクト画面が開いた場合は、WebStorm のメニューから [File]>[Close Project] を選択してスタート画面へ戻り、画面中央の「Open」をクリックします。



8. ファイル選択ダイアログが表示されます。下記フォルダを選択します。

`C:\Users\user01\test01`

9. 以降の手順は、書籍 3.3.1-⑥以降を参照してください。

新規プロジェクト作成は、本書のように WebStorm のプロジェクト作成ウィザード使うこともできます。この方法は、Angular CLI のファイル構造変更等により動作しなくなることがありますので確実にプロジェクト作成できるダウンロードファイルの利用を推奨します。

## 4 章の変更

### 4.3.2.2 テストプログラムのインストール手順変更

1. 3 章の変更の「3.3.1.1 新規プロジェクト作成手順の変更」を参照して、testProgram.exe を自己展開後、プロジェクトを WebStorm に読み込みます。
2. 以降の手順は、本書 4.3.2.2-⑨以降を参照してください。

## 5 章の変更

### 5.2.1.1 新規プロジェクトの作成手順変更

- 3 章の変更の「3.3.1.1 新規プロジェクト作成手順の変更」を参照して、calcApp.exe を自

已展開後、プロジェクトを WebStorm に読み込みます。

#### 5.2.1.2 追加ライブラリのインストール手順変更

この手順は、ダウンロードしたプロジェクトファイルでは実施済ですので不要です。

#### 5.2.1.3 angular-cli.json の設定を styles.css に変更

この手順は、ダウンロードしたプロジェクトファイルでは実施済ですので不要です。

なお、angular-cli.json は仕様変更が頻繁にあり、設定が無効になることがありますので CSS の設定場所を src フォルダの styles.css ファイルに変更します。コンポーネントごとに指定する CSS はコンポーネント内でのみ適応可能ですが、ここに設定するとアプリ全体に適用できます。

ここでは Bootstrap の CSS ファイルを指定します。

設定内容の確認は以下の手順で行います。

1. WebStorm ストームのプロジェクトエクスプローラで src フォルダ内の styles.css ファイルをダブルクリックします。
2. CSS ファイルの内容は以下のとおりです。

---

```
/* bootstrap css のインポート*/
```

```
@import url("../node_modules/bootstrap/dist/css/bootstrap.min.css");
```

---

#### 5.4.3.1 キャッシュ対象ファイルの確認内容変更

ビルド出力されるファイルが本書記載の内容から 2 個追加され、名称も一部変更になりました。

リスト 5-13 index.html の確認

```
//この行は中央付近に表示されます
<link rel="icon" type="image/x-icon" href="favicon.ico">

//以下の記述は、下から 2 行目に 1 行で表示されます
<script type="text/javascript" src="inline.bundle.js"></script>
<script type="text/javascript" src="polyfills.bundle.js"></script>
<script type="text/javascript" src="styles.bundle.js"></script>
```

```
<script type="text/javascript" src="vendor.bundle.js"></script>  
<script type="text/javascript" src="main.bundle.js"></script>
```

#### 5.4.3.2 マニフェストファイルの作成内容変更

キャッシュ対象となる出力ファイルが変更になったため、マニフェストファイルの内容を変更します

リスト 5-14 calcApp.appcache の変更

```
CACHE MANIFEST  
#ver 1.2.0  
CACHE:  
./favicon.ico  
../inline.bundle.js  
../styles.bundle.js  
../main.bundle.js  
../vendor.bundle.js  
../polyfills.bundle.js
```

## 6 章の変更

#### 6.2.2.1 新規プロジェクトの作成手順変更

3 章の変更の「3.3.1.1 新規プロジェクト作成手順の変更」を参照して、cloudApp.exe を自己展開後、プロジェクトを WebStorm に読み込みます。

#### 6.2.2.2 追加ライブラリのバージョン変更

この手順は、ダウンロードしたプロジェクトファイルでは実施済ですので不要です。

#### 5.2.1.3 angular-cli.json の設定を styles.css に変更

この手順は、ダウンロードしたプロジェクトファイルでは実施済ですので不要です。

なお、angular-cli.json は仕様変更が頻繁にあり、設定が無効になることがありますので CSS の設定場所を src フォルダの styles.css ファイルに変更します。コンポーネントごと

に指定する CSS はコンポーネント内でのみ適応可能ですが、ここに設定するとアプリ全体に適用できます。

ここでは、Bootstrap の CSS ファイルを指定します。

設定内容の確認は以下の手順で行います。

1. WebStorm ストームのプロジェクトエクスプローラで src フォルダ内の styles.css ファイルをダブルクリックします。
2. CSS ファイルの内容は以下のとおりです。

---

```
/* bootstrap css のインポート*/
@import url("../node_modules/bootstrap/dist/css/bootstrap.min.css");
```

---

#### 6.4.1.4 共通 CSS 内容変更

bootstrap css を 5.2.1.3 で共通 CSS(styles.css)に bootstrap の記述が追加済ですので、リスト 6-9 は以下になります。

```
/* bootstrap css のインポート*/
@import url("../node_modules/bootstrap/dist/css/bootstrap.min.css");

/*見やすさのために横幅を制限*/
html {
  max-width: 1000px;
  margin-right:auto;
  margin-left:auto;
  background-color: #dae3f3;
}
/*ヘッダーバーとの重なるの調整*/
body{
  padding-Top:50px;
  background-color: #dae3f3;
  font-family: sans-serif;
}

/*スライドインデックスアイコンを上部に表示*/
ol.carousel-indicators{
```

```

top:10px !important;
}
/*スライドインデックスアイコンの枠線を黒*/
ol.carousel-indicators li{
  border-color: black !important;
}

```

#### 6.1.4.5 外部モジュール登録内容変更

ダウンロードしたプロジェクトファイルでは反映済です。

1)ng2-Bootstrap(v1.1.14)を Angular5.0.0 対応のバージョン(v2.0.0-beta.8)へ変更

2)@angular/http/Json クラスが非推奨になったため、@angular/common/http/HttpClient の json メソッドへ変更(http.service.ts ファイル)

※修正箇所には網掛けをしています。

app.module.ts ファイル

```

import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { AppComponent } from './app.component';
import { HttpClientModule, HttpClientModule } from '@angular/common/http';
import { HttpService } from './http.service';
import { CarouselModule, CollapseModule, ModalModule } from 'ngx-bootstrap';
import { DetailComponent } from './detail.component';

@NgModule({
  declarations: [
    AppComponent,
    DetailComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    HttpClientModule,

```



```

    ModalModule.forRoot(),
    CarouselModule.forRoot(),
    CollapseModule.forRoot()
  ],
  providers: [HttpService],
  bootstrap: [AppComponent]
})
export class AppModule {}

```

http.service.ts ファイル

```

//-----
// 通信処理
//-----
import {Injectable} from '@angular/core';
import {HttpParams, JsonpClientBackend, HttpClient} from '@angular/common/http';
import {Observable} from 'rxjs/Observable';
import 'rxjs/add/operator/map';

export interface TourData {
  error: string;
  data: string;
}

@Injectable()
export class HttpService {

  //Web API URL
  WEB_API_URL = 'https://webservice.recruit.co.jp/ab-road/tour/v1/';
  //API キー
  API_KEY = '■入手した API キー■';
  //取得件数
  DEFAULT_SIZE = '30';
  //取得の順番(人気順:5)
  SORT_RANKING = '5';
  //JSONP コールバック関数名(Angular 固有値)

```

```
CALLBACK = 'JSONP_CALLBACK';

constructor(private jsonp: JsonpClientBackend,
             private http: HttpClient) {
}

//クラウドからツアー情報取得
getTourData(areaCode: string): Observable<TourData> {
  //接続設定
  let config = this.setParam(areaCode);
  //データ取得
  return this.reqData(config);
}

//通信設定値作成
setParam(areaCode: string): HttpParams {
  //Url パラメータオブジェクト作成
  return (new HttpParams())
    .set('key', this.API_KEY)
    .set('area', areaCode)
    .set('order', this.SORT_RANKING)
    .set('count', this.DEFAULT_SIZE)
    .set('format', 'jsonp')
    .set('callback', this.CALLBACK);
}

//HTTP リクエストとレスポンス処理
reqData(config): Observable<TourData> {
  let url = this.WEB_API_URL + '?' + config.toString();
  return this.http.jsonp<{ results: any }>(url, this.CALLBACK)
    .map(res => {
      let tourData;
      if (res.results.error) {
        //Web API リクエスト失敗
        let err = res.results.error[0];

```

```
    tourData = {
      error: err.code,
      data: err.message
    };
  } else {
    //Web API リクエスト成功
    let dataObj = res.results.tour;
    tourData = {
      error: null,
      data: dataObj,
    };
  }
  console.dir(tourData);
  return tourData;
}
);
}
```