

# README

May 23, 2016

## Contents

<b>1</b>	<b>Config</b>	<b>2</b>
1.1	: package.json . . . . .	2
1.2	: tsconfig.json . . . . .	3
1.3	: typings.json . . . . .	4
1.4	: systemjs.config.js . . . . .	4
<b>2</b>	<b>App</b>	<b>5</b>
2.1	: app/main.ts . . . . .	5
2.2	: app/app.component.ts . . . . .	6
2.3	: app/in-memory-data.service.ts . . . . .	7
2.4	Dashboard . . . . .	8
2.4.1	: app/dashboard.component.ts . . . . .	8
2.5	Pages . . . . .	9
2.5.1	: app/page.ts . . . . .	9
2.5.2	: app/pages.component.ts . . . . .	9
2.5.3	: app/page-detail.component.ts . . . . .	11
2.5.4	: app/page.service.ts . . . . .	12
<b>3</b>	<b>Template</b>	<b>15</b>
3.1	: index.html . . . . .	15
3.2	Dashboard . . . . .	16
3.2.1	: app/dashboard.component.html . . . . .	16
3.3	Pages . . . . .	16
3.3.1	: app/pages.component.html . . . . .	16
3.3.2	: app/page-detail.component.html . . . . .	17

<b>4</b>	<b>Styles</b>	<b>17</b>
4.1	: styles.css . . . . .	18
4.2	: app/app.component.css . . . . .	19
4.3	Dashboard . . . . .	19
4.3.1	: app/dashboard.component.css . . . . .	19
4.4	Pages . . . . .	21
4.4.1	: app/pages.component.css . . . . .	21
4.4.2	: app/page-detail.component.css . . . . .	22
<b>5</b>	<b>Dev</b>	<b>23</b>
5.1	Mon May 23 21:42:18 CDT 2016 . . . . .	23
5.2	Sun May 22 14:10:19 CDT 2016 . . . . .	23
5.3	Sat May 21 22:28:33 CDT 2016 . . . . .	24
5.4	Tutorials . . . . .	25
<b>Prairie Hill Learning Center</b>		
NAV => 5 / 4 / Index / 3 / 2.4		

## 1 Config

### 1.1 : package.json

package.json

```
{
  "name": "phill-angular2",
  "version": "1.0.0",
  "scripts": {
    "start": "tsc && concurrently \"npm run tsc:w\" \"npm run lite\" ",
    "lite": "lite-server",
    "postinstall": "typings install",
    "tsc": "tsc",
    "tsc:w": "tsc -w",
    "typings": "typings"
  },
  "license": "ISC",
  "dependencies": {
    "@angular/common": "2.0.0-rc.1",
    "@angular/compiler": "2.0.0-rc.1",
    "@angular/core": "2.0.0-rc.1",
    "@angular/http": "2.0.0-rc.1",
```

```

    "@angular/platform-browser": "2.0.0-rc.1",
    "@angular/platform-browser-dynamic": "2.0.0-rc.1",
    "@angular/router": "2.0.0-rc.1",
    "@angular/router-deprecated": "2.0.0-rc.1",
    "@angular/upgrade": "2.0.0-rc.1",
    "systemjs": "0.19.27",
    "core-js": "^2.4.0",
    "reflect-metadata": "^0.1.3",
    "rxjs": "5.0.0-beta.6",
    "zone.js": "^0.6.12",
    "angular2-in-memory-web-api": "0.0.10",
    "bootstrap": "^3.3.6"
  },
  "devDependencies": {
    "concurrently": "^2.0.0",
    "lite-server": "^2.2.0",
    "typescript": "^1.8.10",
    "typings": "^0.8.1"
  }
}

```

## 1.2 : tsconfig.json

tsconfig.json

```

{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "moduleResolution": "node",
    "sourceMap": true,
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "removeComments": false,
    "noImplicitAny": false
  },
  "exclude": [
    "node_modules",
    "typings/main",
    "typings/main.d.ts"
  ]
}

```

```
]
}
```

### 1.3 : typings.json

typings.json

```
{
  "ambientDependencies": {
    "core-js": "registry:dt/core-js#0.0.0+20160317120654",
    "jasmine": "registry:dt/jasmine#2.2.0+20160412134438",
    "node": "registry:dt/node#4.0.0+20160509154515"
  }
}
```

### 1.4 : systemjs.config.js

systemjs.config.js

```
/**
 * System configuration for Angular 2 samples
 * Adjust as necessary for your application needs.
 */
(function(global) {
  // map tells the System loader where to look for things
  var map = {
    'app': 'app', // 'dist',
    '@angular': 'node_modules/@angular',
    'angular2-in-memory-web-api': 'node_modules/angular2-in-memory-web-api',
    'rxjs': 'node_modules/rxjs'
  };
  // packages tells the System loader how to load when no filename and/or no extension
  var packages = {
    'app': { main: 'main.js', defaultExtension: 'js' },
    'rxjs': { defaultExtension: 'js' },
    'angular2-in-memory-web-api': { main: 'index.js', defaultExtension: 'js' }
  };
  var ngPackageNames = [
    'common',
    'compiler',
    'core',

```

```

    'http',
    'platform-browser',
    'platform-browser-dynamic',
    'router',
    'router-deprecated',
    'upgrade',
  ];
  // Add package entries for angular packages
  ngPackageNames.forEach(function(pkgName) {
    packages['@angular/' + pkgName] = { main: pkgName + '.umd.js', defaultExtension: 'js'
  });
  var config = {
    map: map,
    packages: packages
  }
  System.config(config);
})(this);

```

## 2 App

NAV => Index / 5 / 4 / 3 / 2.4

### 2.1 : app/main.ts

app/main.ts

```

// TEMP: Imports for loading & configuring the in-memory web api
import { provide } from '@angular/core';
import { XHRBackend } from '@angular/http';

import { InMemoryBackendService, SEED_DATA } from 'angular2-in-memory-web-api';
import { InMemoryDataService } from './in-memory-data.service';

import { bootstrap }    from '@angular/platform-browser-dynamic';
import { HTTP_PROVIDERS } from '@angular/http';

import { AppComponent } from './app.component';

bootstrap(AppComponent, [
  HTTP_PROVIDERS,

```

```

    provide(XHRBackend, {
      useClass: InMemoryBackendService }), // in-mem server
    provide(SEED_DATA, {
      useClass: InMemoryDataService }) // in-mem server data
  ]);

```

## 2.2 : app/app.component.ts

app/app.component.ts

```

import { Component } from '@angular/core';
import { RouteConfig, ROUTER_DIRECTIVES, ROUTER_PROVIDERS } from '@angular/router-depre

import { DashboardComponent } from './dashboard.component';
import { PagesComponent } from './pages.component';
import { PageDetailComponent } from './page-detail.component';
import { PageService } from './page.service';

@Component({
  selector: 'my-app',

  template: `
    <h1>{{title}}</h1>
    <nav>

      <a [routerLink]="['Dashboard']">Dashboard</a>
      <a [routerLink]="['Pages']">Pages</a>
    </nav>
    <router-outlet></router-outlet>
  `,
  styleUrls: ['app/app.component.css'],
  directives: [ROUTER_DIRECTIVES],
  providers: [
    ROUTER_PROVIDERS,
    PageService
  ]
})

@RouteConfig([
  {

```

```

        path: '/dashboard',
        name: 'Dashboard',
        component: DashboardComponent,
        useAsDefault: true
    },
    {
        path: '/detail/:id',
        name: 'PageDetail',
        component: PageDetailComponent
    },
    {
        path: '/pages',
        name: 'Pages',
        component: PagesComponent
    }
]

export class AppComponent {
    title = 'Prairie Hill Learning Center';
}

<a [routerLink]="['Home']">Home</a>
<a [routerLink]="['About']">About</a>
<a [routerLink]="['Programs']">Programs</a>
<a [routerLink]="['Tours']">Tours</a>
<a [routerLink]="['Staff']">Staff</a>
<a [routerLink]="['Calendar']">Calendar</a>
<a [routerLink]="['Employment']">Employment</a>
<a [routerLink]="['Donate']">Donate</a>
<a [routerLink]="['Contact']">Contact</a>
<a [routerLink]="['Events']">Events</a>

```

## 2.3 : app/in-memory-data.service.ts

app/in-memory-data.service.ts

```

export class InMemoryDataService {
    createDb() {
        let pages = [
            { "id": 1, "title": "Home" },

```

```

        { "id": 2, "title": "About"      },
        { "id": 3, "title": "Programs"   },
        { "id": 4, "title": "Tours"      },
        { "id": 5, "title": "Staff"       },
        { "id": 6, "title": "Calendar"    },
        { "id": 7, "title": "Employment" },
        { "id": 8, "title": "Donate"      },
        { "id": 9, "title": "Contact"     },
        { "id": 10, "title": "Events"     }
    ];
    return {pages};
  }
}

```

## 2.4 Dashboard

NAV => ?? / ?? / ?? => ?? / ??

### 2.4.1 : app/dashboard.component.ts

```

app/dashboard.component.ts

import { Component, OnInit } from '@angular/core';

import { Page } from '../page';
import { PageService } from '../page.service';

import { Router } from '@angular/router-deprecated';

@Component({
  selector: 'my-dashboard',
  templateUrl: 'app/dashboard.component.html',
  styleUrls: ['app/dashboard.component.css']
})

export class DashboardComponent implements OnInit {

  pages: Page[] = [];

  constructor(
    private router: Router,

```



```

        private pageService: PageService) {
    }

    ngOnInit() {
        this.pageService.getPages()
            .then(pages => this.pages = pages.slice(1,5));
    }

    gotoDetail(page: Page){
        let link = ['PageDetail', { id: page.id }];
        this.router.navigate(link);
    }
}

```

## 2.5 Pages

TEMP => ?? / ?? / ?? SERV => ?? COMP => ?? / ?? / ?? STYL => ?? /

### 2.5.1 : app/page.ts

app/page.ts

```

export class Page {
    id: number;
    title: string;
}

```

### 2.5.2 : app/pages.component.ts

app/pages.component.ts

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router-deprecated';

import { Page } from './page';
import { PageService } from './page.service';
import { PageDetailComponent } from './page-detail.component';

@Component({
    selector: 'my-pages',

```

```

        templateUrl: 'app/pages.component.html',
        styleUrls: ['app/pages.component.css'],
        directives: [PageDetailComponent]
    })

export class PagesComponent implements OnInit {
    pages: Page[];
    selectedPage: Page;
    addingPage = false;
    error: any;

    constructor(
        private _router: Router,
        private _pageService: PageService) { }

    getPages() {
        this._pageService
            .getPages()
            .then(pages => this.pages = pages)
            .catch(error => this.error = error); //TODO: Display error message
    }

    addPage() {
        this.addingPage = true;
        this.selectedPage = null;
    }

    close(savedPage: Page) {
        this.addingPage = false;
        if (savedPage) { this.getPages(); }
    }

    delete(page: Page, event: any) {
        event.stopPropagation();
        this._pageService
            .delete(page)
            .then(res => {
                this.pages = this.pages.filter(h => h !== page);
                if (this.selectedPage === page) {
                    this.selectedPage = null; }
            })
    }

```

```

        })
        .catch(error => this.error = error); // TODO: Display error message
    }

    ngOnInit() {
        this.getPages();
    }

    onSelect(page: Page) {
        this.selectedPage = page;
        this.addingPage = false;
    }

    gotoDetail() {
        this._router.navigate(['PageDetail', {
            id: this.selectedPage.id }]);
    }
}

```

### 2.5.3 : app/page-detail.component.ts

app/page-detail.component.ts

```

import { Component, Input, Output, OnInit, EventEmitter } from '@angular/core';
import { RouteParams } from '@angular/router-deprecated';

import { Page } from '../page';
import { PageService } from '../page.service';

@Component({
    selector: 'my-page-detail',
    templateUrl: 'app/page-detail.component.html',
    styleUrls: ['app/page-detail.component.css']
})

export class PageDetailComponent implements OnInit {
    @Input() page: Page;
    @Output() close = new EventEmitter();
    error: any;
    navigated = false; // true if navigated here
}

```

```

    constructor(
        private _pageService: PageService,
        private _routeParams: RouteParams) {
    }

    ngOnInit() {
        if (this._routeParams.get('id') !== null) {
            let id = +this._routeParams.get('id');
            this.navigated = true;
            this._pageService.getPage(id)
                .then(page => this.page = page);
        } else {
            this.navigated = false;
            this.page = new Page();
        }
    }

    save() {
        this._pageService
            .save(this.page)
            .then(page => {
                this.page = page; // saved page, w/ id if new
                this.goBack(page);
            })
            .catch(error => this.error = error); // TODO: Display error message
    }

    goBack(savedPage: Page = null) {
        this.close.emit(savedPage);
        if (this.navigated) {
            window.history.back();
        }
    }
}

```

#### 2.5.4 : app/page.service.ts

app/page.service.ts

```

import { Injectable } from '@angular/core';
import { Http, Headers } from '@angular/http';

import 'rxjs/add/operator/toPromise';

import { Page } from '../page';

@Injectable()
export class PageService {

    private pagesUrl = 'app/pages'; // URL to web api

    constructor(private http: Http) { }

    // CREATE new Page
    private post(page: Page): Promise<Page> {
        let headers = new Headers({
            'Content-Type': 'application/json'
        });

        return this.http
            .post(this.pagesUrl,
                JSON.stringify(page),
                {headers: headers})
            .toPromise()
            .then(res => res.json().data)
            .catch(this.handleError);
    }

    // READ existing Page(s)
    getPages(): Promise<Page[]> {
        return this.http.get(this.pagesUrl)
            .toPromise()
            .then(response => response.json().data)
            .catch(this.handleError);
    }

    getPage(id: number) {
        return this.getPages()
            .then(pages => pages.filter(page => page.id === id)[0]);
    }
}

```

```

// UPDATE existing Page
private put(page: Page) {
    let headers = new Headers();
    headers.append('Content-Type',
        'application/json');

    let url = `${this.pagesUrl}/${page.id}`;

    return this.http
        .put(url, JSON.stringify(page),
            {headers: headers})
        .toPromise()
        .then(() => page)
        .catch(this.handleError);
}

// DESTROY existing Page
delete(page: Page) {
    let headers = new Headers();
    headers.append('Content-Type',
        'application/json');

    let url = `${this.pagesUrl}/${page.id}`;

    return this.http
        .delete(url, headers)
        .toPromise()
        .catch(this.handleError);
}

// SAVE combination of _post and _put methods
save(page: Page): Promise<Page> {
    if (page.id) {
        return this.put(page);
    }
    return this.post(page);
}

// handle ERRORS
private handleError(error: any) {

```

```

        console.error('An error occurred', error);
        return Promise.reject(error.message || error);
    }
}

```

### 3 Template

APP => 2 / SERV => ?? / COMP => ?? / ?? TEMP => ??

#### 3.1 : index.html

index.html

```

<html>
  <head>
    <base href="/">

    <title>Prairie Hill Learning Center</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--<link rel="stylesheet" href="css/pure-release-0.6.0/pure-min.css">-->
    <link rel="stylesheet" href="styles.css">
    <link href="//fonts.googleapis.com/css?family=Lobster|Roboto:400,100,100italic,700">

    <!-- 1. Load libraries -->
    <!-- Polyfill(s) for older browsers -->
    <script src="node_modules/core-js/client/shim.min.js"></script>
    <script src="node_modules/zone.js/dist/zone.js"></script>
    <script src="node_modules/reflect-metadata/Reflect.js"></script>
    <script src="node_modules/systemjs/dist/system.src.js"></script>

    <!--<script src="https://www.gstatic.com/firebasejs/3.0.0/firebase.js"></script>-->

    <!-- 2. Configure SystemJS -->
    <script src="systemjs.config.js"></script>
    <script>
      System.import('app').catch(function(err){ console.error(err); });
    </script>
  </head>
  <!-- 3. Display the application -->

```

```

<body>
  <div id="header" styleName="pure-g">
    <phill-header>...</phill-header>
  </div>
  <div id="main" styleName="pure-g">
    <my-app>Loading...</my-app>
  </div>
  <div id="footer" styleName="pure-g">
    <phill-footer>...</phill-footer>
  </div>
</body>
</html>

```

## 3.2 Dashboard

NAV => ?? / ?? /

### 3.2.1 : app/dashboard.component.html

app/dashboard.component.html

```

<h3>Prairie Hill Pages (Spaces)</h3>
<div class="grid grid-pad">
  <div *ngFor="let page of pages"
    (click)="gotoDetail(page)" class="col-1-4">
    <div class="module page">
      <h4>{{page.title}}</h4>
    </div>
  </div>
</div>

```

## 3.3 Pages

### 3.3.1 : app/pages.component.html

app/pages.component.html

```

<h2>My Pages</h2>
<ul class="pages">
  <li *ngFor="let page of pages"
    (click)="onSelect(page)"
    [class.selected]="page === selectedPage">

```



```

    <span class="page-element">
      <span class="badge">{{page.id}}</span> {{page.title}}
    </span>
    <button class="delete-button"
      (click)="delete(page, $event)">Delete</button>
  </li>
</ul>

<button (click)="addPage()">Add New Page</button>
<div *ngIf="addingPage">
  <my-page-detail (close)="close($event)"></my-page-detail>
</div>

<div *ngIf="selectedPage">
  <h2>
    {{selectedPage.title | uppercase}} is your current page
  </h2>
  <button (click)="gotoDetail()">View Details</button>
</div>

```

### 3.3.2 : app/page-detail.component.html

app/page-detail.component.html

```

<div *ngIf="page">
  <h2>{{page.title}}</h2>
  <div>
    <label>id: </label>{{page.id}}
  </div>
  <div>
    <label>title: </label>
    <input [(ngModel)]="page.title" placeholder="title"/>
  </div>
  <button (click)="goBack()">Back</button>
  <button (click)="save()">Save</button>
</div>

```

## 4 Styles

5 / 2

## 4.1 : styles.css

```
styles.css

h1 {
  color: #369;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 250%;
}
h2 {
  color: #444;
  font-family: Arial, Helvetica, sans-serif;
  font-weight: lighter;
}
body {
  margin: 2em;
}
body, input[text], button {
  color: #888;
  font-family: Cambria, Georgia;
}
button {
  font-family: Arial;
  background-color: #eee;
  border: none;
  padding: 5px 10px;
  border-radius: 4px;
  cursor: pointer;
  cursor: hand;
}
button:hover {
  background-color: #cfd8dc;
}
button:disabled {
  background-color: #eee;
  color: #aaa;
  cursor: auto;
}
/* everywhere else */
* {
  font-family: Arial, Helvetica, sans-serif;
```

```
}
```

NAV => ?? / ?? / ??

## 4.2 : app/app.component.css

app/app.component.css

```
h1 {
  font-size: 1.2em;
  color: #999;
  margin-bottom: 0;
}
h2 {
  font-size: 2em;
  margin-top: 0;
  padding-top: 0;
}
nav a {
  padding: 5px 10px;
  text-decoration: none;
  margin-top: 10px;
  display: inline-block;
  background-color: #eee;
  border-radius: 4px;
}
nav a:visited, a:link {
  color: #607D8B;
}
nav a:hover {
  color: #039be5;
  background-color: #CFD8DC;
}
nav a.router-link-active {
  color: #039be5;
}
```

## 4.3 Dashboard

### 4.3.1 : app/dashboard.component.css

app/dashboard.component.css

```

[class*='col-'] {
    float: left;
}
*, *:after, *:before {
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
}
h3 {
    text-align: center; margin-bottom: 0;
}
[class*='col-'] {
    padding-right: 20px;
    padding-bottom: 20px;
}
[class*='col-']:last-of-type {
    padding-right: 0;
}
.grid {
    margin: 0;
}
.col-1-4 {
    width: 25%;
}
.module {
    padding: 20px;
    text-align: center;
    color: #eee;
    max-height: 120px;
    min-width: 120px;
    background-color: #607D8B;
    border-radius: 2px;
}
h4 {
    position: relative;
}
.module:hover {
    background-color: #EEE;
    cursor: pointer;
    color: #607d8b;
}

```

```

}
.grid-pad {
  padding: 10px 0;
}
.grid-pad > [class*='col-']:last-of-type {
  padding-right: 20px;
}
@media (max-width: 600px) {
  .module {
    font-size: 10px;
    max-height: 75px; }
}
@media (max-width: 1024px) {
  .grid {
    margin: 0;
  }
  .module {
    min-width: 60px;
  }
}
}

```

## 4.4 Pages

### 4.4.1 : app/pages.component.css

app/pages.component.css

```

.selected {
  background-color: #CFD8DC !important;
  color: white;
}
.pages {
  margin: 0 0 2em 0;
  list-style-type: none;
  padding: 0;
  width: 15em;
}
.pages li {
  cursor: pointer;
  position: relative;
  left: 0;
}

```

```

        background-color: #EEE;
        margin: .5em;
        padding: .3em 0;
        height: 1.6em;
        border-radius: 4px;
    }
    .pages li.selected:hover {
        background-color: #BBD8DC !important;
        color: white;
    }
    .pages li:hover {
        color: #607D8B;
        background-color: #DDD;
        left: .1em;
    }
    .pages .text {
        position: relative;
        top: -3px;
    }
    .pages .badge {
        display: inline-block;
        font-size: small;
        color: white;
        padding: 0.8em 0.7em 0 0.7em;
        background-color: #607D8B;
        line-height: 1em;
        position: relative;
        left: -1px;
        top: -4px;
        height: 1.8em;
        margin-right: .8em;
        border-radius: 4px 0 0 4px;
    }

```

#### 4.4.2 : app/page-detail.component.css

app/page-detail.component.css

```

label {
    display: inline-block;

```

```

    width: 3em;
    margin: .5em 0;
    color: #607D8B;
    font-weight: bold;
  }
  input {
    height: 2em;
    font-size: 1em;
    padding-left: .4em;
  }
  button {
    margin-top: 20px;
    font-family: Arial;
    background-color: #eee;
    border: none;
    padding: 5px 10px;
    border-radius: 4px;
    cursor: pointer; cursor: hand;
  }
  button:hover {
    background-color: #cfd8dc;
  }
  button:disabled {
    background-color: #eee;
    color: #ccc;
    cursor: auto;
  }

```

## 5 Dev

### 5.1 Mon May 23 21:42:18 CDT 2016

### 5.2 Sun May 22 14:10:19 CDT 2016

<https://angular.io/docs/ts/latest/tutorial/toh-pt5.html>

Routing

- ☐ turn 2 into an application shell that only handles navigation
- ☐ relocate *Pages* concerns within the current `app.component.js` to a separate `PagesComponent`

- ☐ add routing
- ☐ create a new `DashboardComponent`
- ☐ tie the *Dashboard* into the navigation structure

### 5.3 Sat May 21 22:28:33 CDT 2016

<https://angular.io/docs/js/latest/quickstart.html>

Angular2 is written with TypeScript(ES6). This is the future.

<https://angular.io/docs/ts/latest/quickstart.html>

1. Create the app's project folder and define package dependencies and special project setup

- (a) Create the project folder

You are in it.

- (b) Add package definition and configuration files

1

?? ?? ?? ??

- (c) Install packages

`npm install`

- ☐ npm WARN

– ☒ optional

Skipping failed optional dependency /chokidar/fsevents:

<https://github.com/paulmillr/chokidar/issues/425>

"It's just a warning, not an error. You can safely ignore it. Fsevents is an optional dependency and is used on only on OSX." –nono

– ☒ notsup

Not compatible with your operating system or architecture: fsevents@1.0.0

– ☐ phill-angular2@1.0.0 No repository field.

2. Create the app's Angular root component

---

1 / 2



---

app/??

Structure of every component:

---

- One or more `import` statements to reference the things we need.
- A `@Component` decorator that tells Angular what template to use and how to create the component.
- A `component class` that controls the appearance and behavior of a view through its template.

3. Add ??, identifying the root component to Angular

2

app/??

4. Add ??, the web page that hosts the application

3

5. Build and run the app

`npm start`

## 5.4 Tutorials

<https://angular.io/docs/ts/latest/tutorial/toh-pt3.html>

??

??

2