**WORKSHOP 1: GETTING STARTED**

**PART 1: CREATE AND BUILD FIRST UP [1h]**

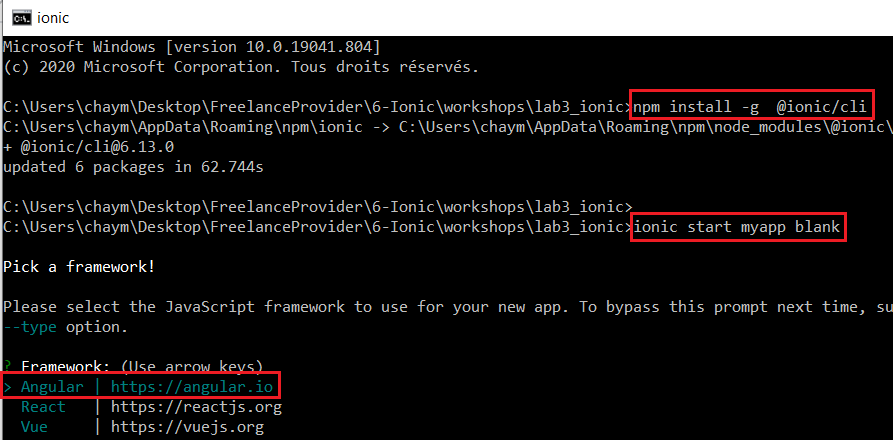
To begin, start by installing IONIC (we suppose that you have installed node so you have npm)

npm install -g @ionic/cli

After successfully installed Node.js and Ionic CLI, you can generate a new project using your terminal or command prompt.

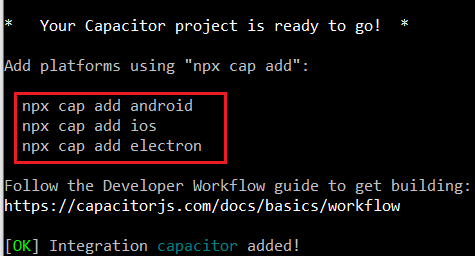
Enter the following command to generate a project:

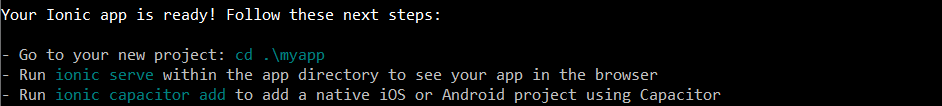
ionic start myapp blank



That's it, now wait for the CLI to generate a project with name myapp or any name you choose ,based on the blank template which has basic scaffolding for an Ionic project with one example Home page .



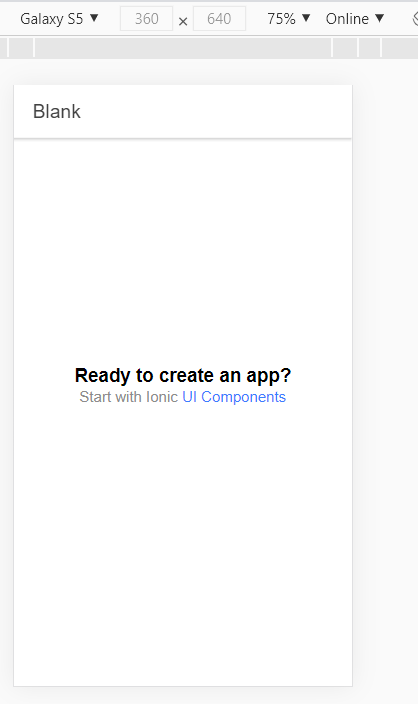




After finishing the initial setup, you should be able to enter to your project directory and serve you app .

cd myapp

ionic serve



**Exercise 1 : [15min]**

Try to create others projects using different template: tabs, sidemenu

**PART 2: LAYOUT [2h]**

* Link 1: https://ionicframework.com/docs/layout/structure
* Link 2 : <https://ionicframework.com/docs/components>

Ionic Framework provides several different layouts that can be used to structure an app. From single page layouts, to split pane views and modals.

## [Header and Footer Layout](https://ionicframework.com/docs/layout/structure#header-and-footer-layout)

### [Header](https://ionicframework.com/docs/layout/structure#header)

The most simple layout available consists of a [header](https://ionicframework.com/docs/api/header) and [content](https://ionicframework.com/docs/api/content). Most pages in an app generally have both of these, but a header is not required in order to use content.



<ion-app>

<ion-header>

<ion-toolbar>

<ion-title>Header</ion-title>

</ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

<h1>Main Content</h1>

</ion-content>

</ion-app>

### [Header and Footer](https://ionicframework.com/docs/layout/structure#header-and-footer)

These can also be combined on one page to have a toolbar above and below the content.

  <ion-header>

    <ion-toolbar>

      <ion-title>Header</ion-title>

    </ion-toolbar>

  </ion-header>

  <ion-content class="ion-padding">

    <h1>Main Content</h1>

  </ion-content>

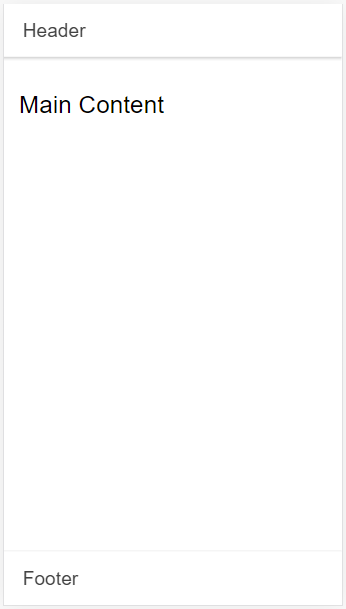
  <ion-footer>

    <ion-toolbar>

      <ion-title>Footer</ion-title>

    </ion-toolbar>

  </ion-footer>



List of items

<ion-content class="ion-padding">

    <h1>Main Content</h1>

    <ion-list>

      <ion-item>

        <ion-label>Pokémon Yellow</ion-label>

      </ion-item>

      <ion-item>

        <ion-label>Mega Man X</ion-label>

      </ion-item>

      <ion-item>

        <ion-label>The Legend of Zelda</ion-label>

      </ion-item>

      <ion-item>

        <ion-label>Pac-Man</ion-label>

      </ion-item>

      <ion-item>

        <ion-label>Super Mario World</ion-label>

      </ion-item>

    </ion-list>

  </ion-content>

List of inputs

<!-- List of Input Items -->

<ion-list>

  <ion-item>

    <ion-label>Input</ion-label>

    <ion-input></ion-input>

  </ion-item>

  <ion-item>

    <ion-label>Toggle</ion-label>

    <ion-toggle slot="end"></ion-toggle>

  </ion-item>

  <ion-item>

    <ion-label>Radio</ion-label>

    <ion-radio slot="end"></ion-radio>

  </ion-item>

  <ion-item>

    <ion-label>Checkbox</ion-label>

    <ion-checkbox slot="start"></ion-checkbox>

  </ion-item>

</ion-list>

# UI Components

<https://ionicframework.com/docs/components>

# ion-button

# ion-badge

# ion-content

# ion-card

# ion-alert

**Exercise 2: [20 min]**

1-Create new sidemenu application then add some ionic component like card button…

2-Add new page (About us) and button to navigate between.

**PART 3: CONSUME WEB API [2h]**

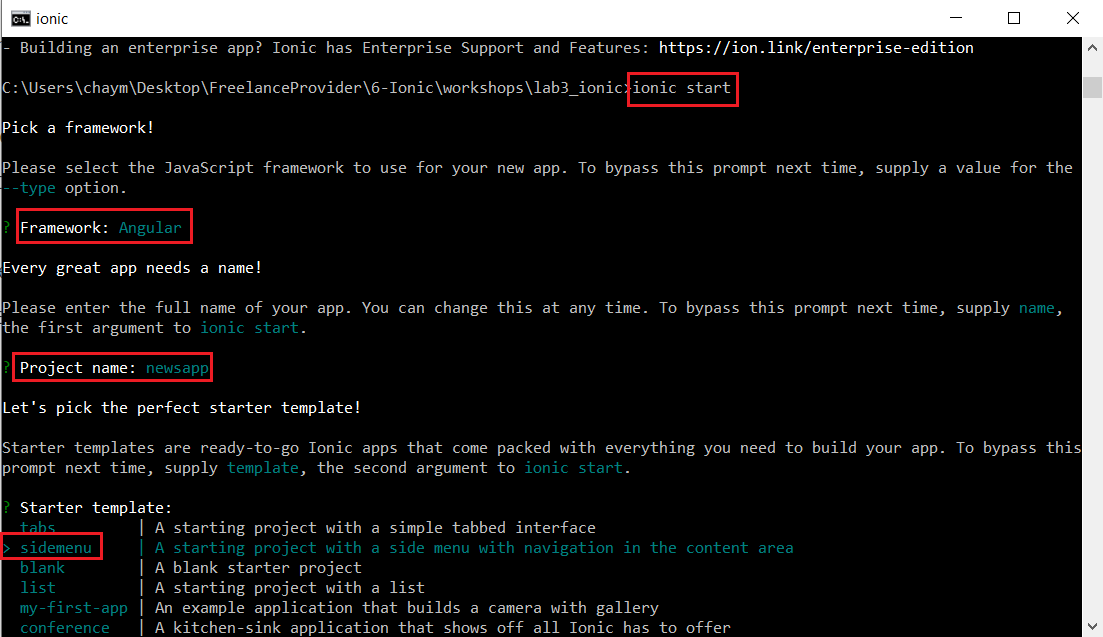
We’ll be using Ionic with Angular, which is based on TypeScript, so you need to be familiar with the basic concepts of TypeScript.

Next, you can generate a project based on Angular by running the following command:

ionic start

The CLI will interactively ask you for the necessary information about your project, such as the **name** (Enter **newsapp** or whatever name you prefer) and the starter template (choose **sidemenu** which will give you a starting project with a side menu and navigation of the box).

Next press **Enter** to instruct the CLI to start generating the files and installing the dependencies from npm.



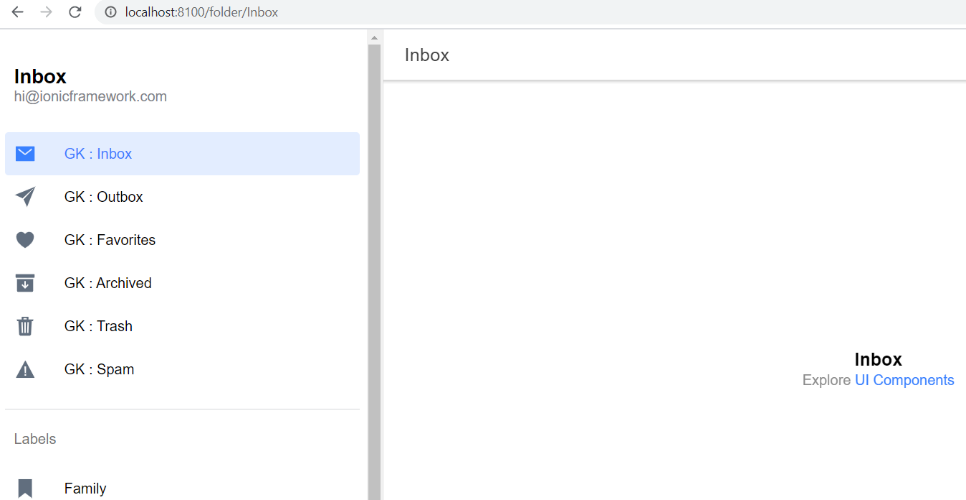
*Appflow is a continuous integration and deployment platform for Ionic developers. Appflow helps developers continuously build and ship their iOS, Android, and web apps faster than ever. You can find more information about Appflow from the*[*official docs*](https://ionicframework.com/docs/appflow/)*.*

Next, you can navigate to your project’s root folder and run the following command to start a live-reload development server:

cd ./newsapp

ionic serve

Your application will be available from the http://localhost:8100 address.



Open the src/app/app-routing.module.ts file

import { NgModule } from '@angular/core';

import { PreloadAllModules, RouterModule, Routes } from '@angular/router';

const routes: Routes = [

  {

    path: '',

    redirectTo: 'folder/Inbox',

    pathMatch: 'full'

  },

  {

    path: 'folder/:id',

    loadChildren: () => import('./folder/folder.module').then( m => m.FolderPageModule)

];

@NgModule({

  imports: [

    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })

  ],

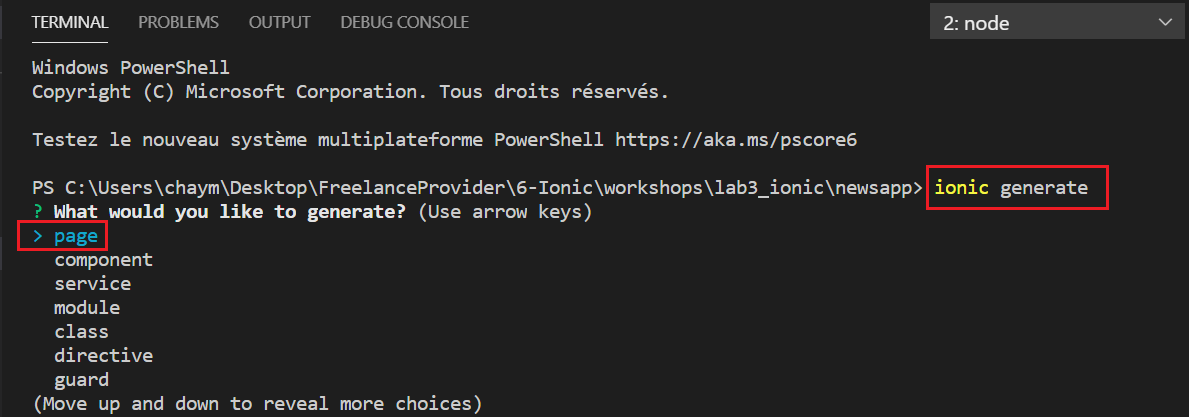
  exports: [RouterModule]

})

export class AppRoutingModule {}

Now, let’s create an about page for our application. In your terminal, run the following command:

ionic generate page about



Let’s add a link to the about page in the side menu.

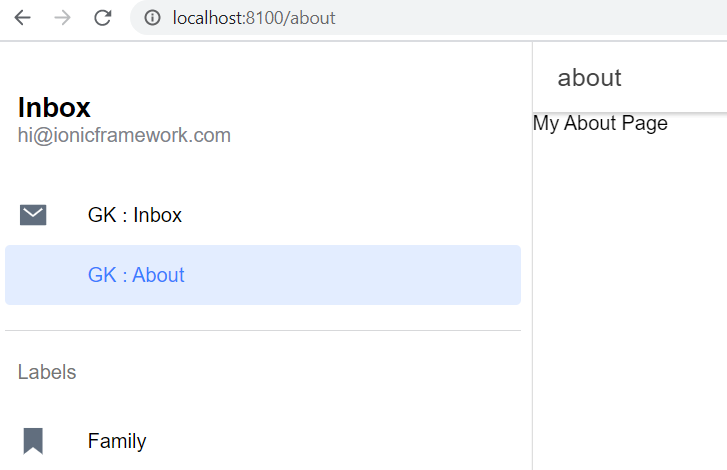
{

      title: 'About',

      url: '/about',

      icon: 'help-circle-outline'

    }



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

@Component({

  selector: 'app-root',

  templateUrl: 'app.component.html',

  styleUrls: ['app.component.scss'],

})

export class AppComponent {

  public appPages = [

    { title: 'Inbox', url: '/folder/Inbox', icon: 'mail' },

    {

      title: 'About',

      url: '/about',

      icon: 'help-circle-outline'

    }

    /\*{ title: 'Outbox', url: '/folder/Outbox', icon: 'paper-plane' },

    { title: 'Favorites', url: '/folder/Favorites', icon: 'heart' },

    { title: 'Archived', url: '/folder/Archived', icon: 'archive' },

    { title: 'Trash', url: '/folder/Trash', icon: 'trash' },

    { title: 'Spam', url: '/folder/Spam', icon: 'warning' },\*/

  ];

  public labels = ['Family', 'Friends', 'Notes', 'Work', 'Travel', 'Reminders'];

  constructor() {}

}

Next, open the src/app/about/about.page.html and add a menu icon to the toolbar of the page, which allows users to open the side menu:

<ion-header>

  <ion-toolbar>

    <ion-title>About</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content>

My About Page

</ion-content>

Open the src/app/about/about.page.html and add a primary color to the menu toolbar and a dark color to the content section:

<ion-header>

  <ion-toolbar color="primary">

    <ion-buttons slot="start">

      <ion-menu-button></ion-menu-button>

    </ion-buttons>

    <ion-title>

      About

    </ion-title>

  </ion-toolbar>

</ion-header>

<ion-content color="dark" padding>

<p>

  This is a news app built with Ionic 4 and the <a href="https://newsapi.org/">News API</a>

</p>

</ion-content>

# Ionic 4 Angular

# Next, let’s theme the homepage. Open the src/app/folder/folder.page.html file and replace its contents with the following:

<ion-header>

  <ion-toolbar color="primary">

    <ion-buttons slot="start">

      <ion-menu-button></ion-menu-button>

    </ion-buttons>

    <ion-title>

      Home

    </ion-title>

  </ion-toolbar>

</ion-header>

<ion-content color="primary">

    <ion-card>

      <ion-card-header>

        <ion-card-subtitle>Welcome to our News App</ion-card-subtitle>

      </ion-card-header>

      <ion-card-content>

        <p>

          Enjoy the latest news from TechCrunch.

        </p>

        <ion-spinner \*ngIf="!articles"  name="dots"></ion-spinner>

      </ion-card-content>

    </ion-card>

</ion-content>

ion-card{

  --background: #021b46;

  --color: #fff;

}

# Next, open the src/app/ folder/folder.page.scss file and add:

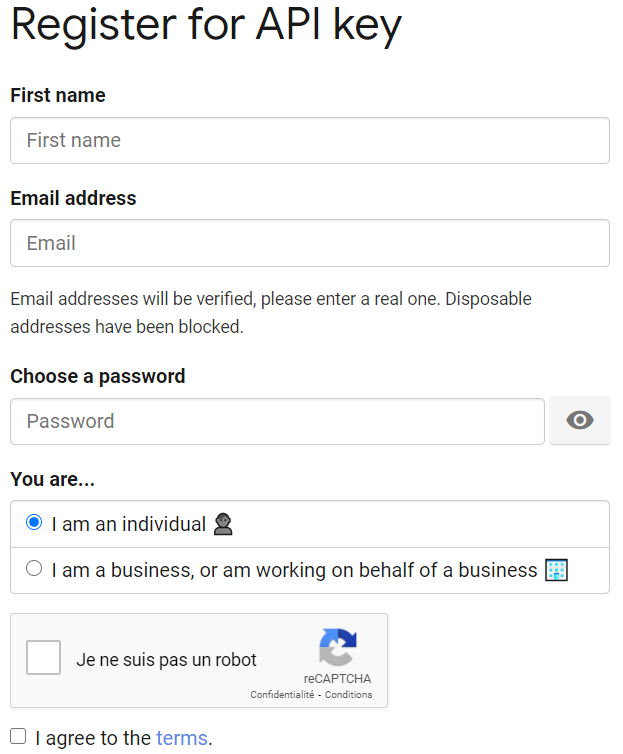
# Ionic 4 Angular TechCrunch

## **Getting News Data**

Let’s now see how you can retrieve news data from the third-party news API available from [NewsAPI.org/](https://newsapi.org/), which offers a free plan for open source and development projects.

You first need to head [here](https://newsapi.org/register) :

<https://newsapi.org/register>



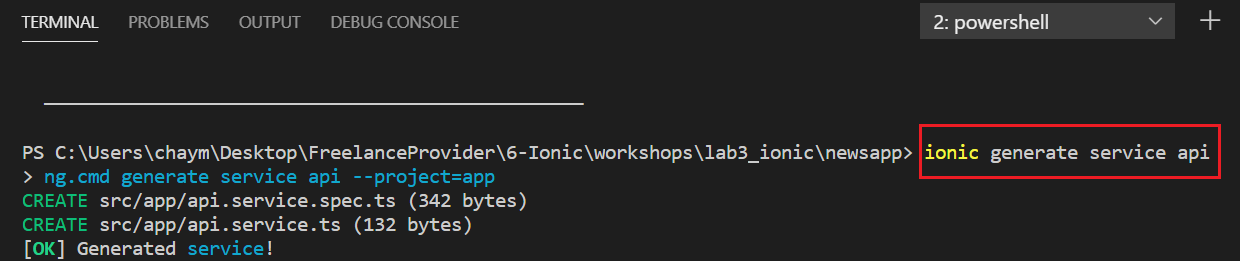
 to register for an API key:

# 

## **Adding a Service**

Next, let’s create a service that will take care of getting data from the news API. In your terminal, run the following command:

ionic generate service api



Next, open the src/app/app.module.ts file then import HttpClientModule and add it to the imports array:

import { HttpClientModule } from '@angular/common/http';

And

@NgModule({

  declarations: [AppComponent],

  entryComponents: [],

  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule, HttpClientModule],

  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],

  bootstrap: [AppComponent],

})

Next, open the src/app/api.service.ts file and inject HttpClient via the service constructor:

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

@Injectable({

  providedIn: 'root'

})

export class ApiService {

  constructor(private httpClient: HttpClient) { }

}

Next, define an API\_KEY variable which will hold your API key from the News API:

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

@Injectable({

  providedIn: 'root'

})

export class ApiService {

  API\_KEY = 'My\_Secret\_Key';

  constructor(private httpClient: HttpClient) { }

  getNews(){

    return this.httpClient.get(`https://newsapi.org/v2/top-headlines?sources=techcrunch&apiKey=${this.API\_KEY}`);

  }

}

# Open the src/app/folder/folder.page.ts file and import, then inject, ApiService via the component constructor:

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

import { ActivatedRoute } from '@angular/router';

import { ApiService } from '../api.service';

@Component({

  selector: 'app-folder',

  templateUrl: './folder.page.html',

  styleUrls: ['./folder.page.scss'],

})

export class FolderPage implements OnInit {

  public folder: string;

  constructor(private apiService: ApiService, private activatedRoute: ActivatedRoute) { }

  ngOnInit() {

   // this.folder = this.activatedRoute.snapshot.paramMap.get('id');

  }

}

# Next, add an articles variable that will hold the retrieved news:

 articles;

# Add an ionViewDidEnter() method, where you call the getNews() method of ApiService to retrieve the news:

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

import { ActivatedRoute } from '@angular/router';

import { ApiService } from '../api.service';

@Component({

  selector: 'app-folder',

  templateUrl: './folder.page.html',

  styleUrls: ['./folder.page.scss'],

})

export class FolderPage implements OnInit {

  //public folder: string;

  articles;

  constructor(private apiService: ApiService) { }

  ngOnInit() {

   // this.folder = this.activatedRoute.snapshot.paramMap.get('id');

    this.ionViewDidEnter();

  }

  ionViewDidEnter(){

    this.apiService.getNews().subscribe((data)=>{

      console.log(data);

      this.articles = data['articles'];

    });

  }

}

Finally, let’s iterate through the articles variable and display the news on our homepage.

Again, open the src/app/folder/folder.page.html file and add the following code:

<ion-header>

  <ion-toolbar color="primary">

    <ion-buttons slot="start">

      <ion-menu-button></ion-menu-button>

    </ion-buttons>

    <ion-title>

      Home

    </ion-title>

  </ion-toolbar>

</ion-header>

<ion-content color="primary">

  <!-- if no articles -->

    <ion-card>

      <ion-card-header>

        <ion-card-subtitle>Welcome to our News App</ion-card-subtitle>

      </ion-card-header>

      <ion-card-content>

        <p>

          Enjoy the latest news from TechCrunch.

        </p>

        <ion-spinner \*ngIf="!articles"  name="dots"></ion-spinner>

      </ion-card-content>

    </ion-card>

    <!-- if articles -->

    <ion-card \*ngFor="let article of articles">

      <ion-img src="{{article.urlToImage}}"></ion-img>

      <ion-card-header>

        <ion-card-subtitle>{{article.title}}</ion-card-subtitle>

      </ion-card-header>

      <ion-card-content>

        <p>

          {{article.description}}

        </p>

        <ion-button ion-button fill="outline" href="{{article.url}}" large>Read full article</ion-button>

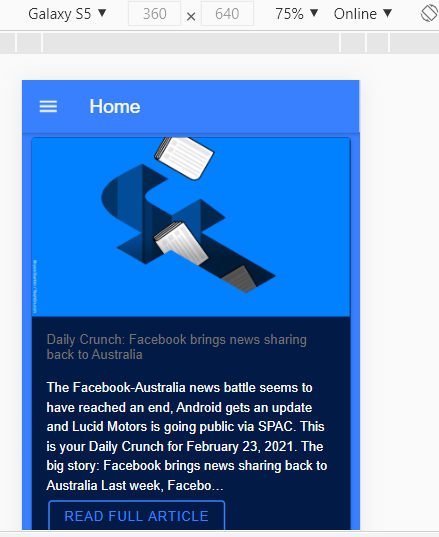
      </ion-card-content>

    </ion-card>

</ion-content>

We simply use the ngFor directive to loop through the articles variable and display the image, title, description and URL of each article inside a card component.

This is a screenshot of the result:



# 

**Exercise [1h]: IONIC-NODE EXPRESS MONGO STACK**

Create new Ionic App(sidemenu type) and display data from NodeExpressJS backend app (liste des équipes)