

BTS SIO2 slam4	SLAM4 : Application mobile Découverte d'ionic 4 et format JSON.	Antony Santero 2019/2020
---------------------------------	---	------------------------------------

Découverte d'Ionic 4

- Rappel : qu'est-ce c'est que Ionic ?

Ionic est un framework gratuit et Open Source créé en 2013, permettant de développer rapidement des applications mobiles multi-plateforme notamment pour Android, iOS et le web, à partir d'une seule base de code. Ionic 4 est sorti en janvier 2019. Vous connaissez le HTML, CSS et JAVASCRIPT ? , vous pouvez y aller !

Installation de l'environnement :

NodeJS

Pour fonctionner, Ionic a besoin d'utiliser NodeJS. Nous allons vérifier si une version d'Ionic et laquelle est installée, en saisissant dans un invite de commande : `node -v` . Ionic 4 utilise Nodejs version 10. Pour installer nodejs version 10, téléchargez le fichier suivant :

<https://nodejs.org/dist/latest-v10.x/node-v10.17.0-x64.msi>

Ionic 4

Ensuite nous allons installer Ionic 4. Dans l'invite de commande : `npm install -g ionic`

Visual Studio Code

Télécharger et installer la dernière version de Visual Studio Code si nécessaire.

Notre premier programme :

Dans l'invite de commande, allez jusqu'au répertoire « Mes Documents »

Puis saisir : `ionic start MonApplonic blank`

- Sélectionnez le framework angular
- Would like to share : No

Lancez Visual Studio Code

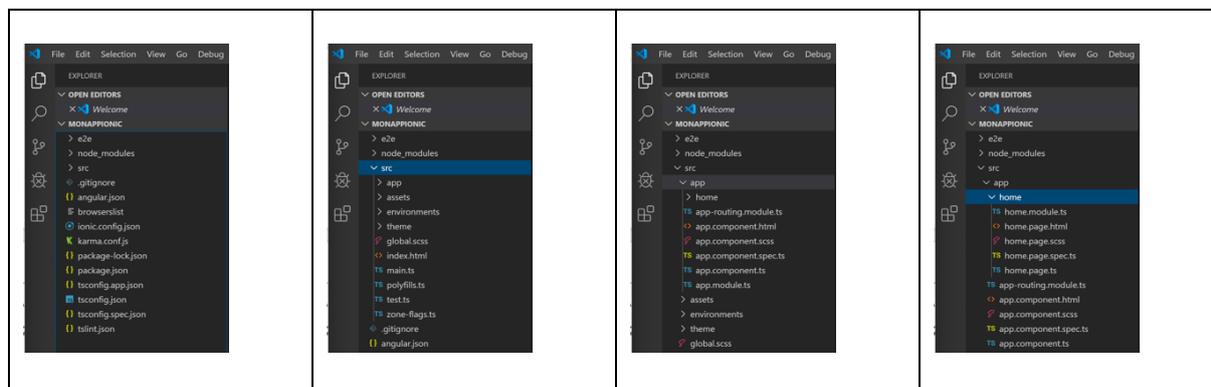
- Cliquez sur `file -> open folder`
- Sélectionnez le dossier « MonApplonic » dans le répertoire « Mes Documents »
- Ouvrir un terminal dans VSC : cliquez sur `Terminal -> New Terminal`
- Vérifiez que le terminal est de type **cmd** et non **powerscript !!!!! IMPORTANT**
- Pour exécuter votre programme, dans le terminal, saisir : `ionic serve`

Un navigateur s'ouvre avec votre nouvelle application !!!

<p>BTS SIO2 slam4</p>	<p>SLAM4 : Application mobile</p> <p>Découverte d'ionic 4 et format JSON.</p>	<p>Antony Santero</p> <p>2019/2020</p>
---	--	---

Utilisation des codes sources

Notre application ne contient, pour l'instant, qu'une seule page. La page home. Les fichiers se trouvent dans : src->app->home



Seulement 3 fichiers nous intéressent :

- Home.page.html ... c'est le fichier de la vue html de notre page
- Home.page.css ... c'est la feuille de style de notre page
- Home.page.ts ... c'est le contrôleur de notre page

La vue html : home.page.html

Ionic dispose de ses propres composants html. Le fichier home.page.html contient donc des balises correspondantes aux composants. Pour une liste complète rendez-vous sur

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

Le style css : home.page.css

Le fichier home.page.css contient des style css3 identiques a ceux que vous connaissez déjà.

Le controleur ts : home.page.ts

Le fichier home.page.ts contient le code source de notre page. Il est en langage javascript. Ce langage est actuellement le plus utilisé dans le développement informatique. C'est un langage identique au langage java. C'est dans celui-ci que nous allons définir les variables qui seront transmises à notre vue home.page.html .

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

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  constructor() {}
}
```

<p>BTS SIO2 slam4</p>	<p align="center">SLAM4 : Application mobile</p> <p align="center">Découverte d'ionic 4 et format JSON.</p>	<p align="right">Antony Santero 2019/2020</p>
---	--	--

Utilisation des variables

- Pour qu'une variable apparaisse dans votre vue html il faut qu'elle soit déclarée dans votre contrôleur.
- Dans le fichier (contrôleur) home.page.ts, déclarez une variable MonText = « bonjour » dans export
- Dans le fichier html, supprimez le contenu situez entre les balises <ion-content>...</ion-content>
- Dans le fichier html, entre les balises <ion-content>, saisissez {{MonText}}

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

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  MonText="Bonjour";

  constructor() {}
}

<ion-header>
  <ion-toolbar>
    <ion-title>
      Ionic Blank
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  {{MonText}}
</ion-content>
```

Utilisation des boutons

- Dans le fichier html, ajoutez un bouton avec les balises <ion-button>Mon bouton</ion-button>
- Dans le contrôleur, ajoutez la fonction **MonClick()**
- Ajoutez l'évènement click a : <ion-button (click)="ClickSurMonBouton()" >Mon...

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

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  MonText="Bonjour";

  constructor() {}

  MonClick(){
    alert("C'est un click!!");
  }
}

<ion-header>
  <ion-toolbar>
    <ion-title>
      Ionic Blank
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <p>{{MonText}}</p>
  <ion-button (click)="MonClick()">Default</ion-button>
</ion-content>
```

BTS SIO2 slam4	SLAM4 : Application mobile Découverte d'ionic 4 et format JSON.	Antony Santero 2019/2020
---------------------------------	---	------------------------------------

Création d'une nouvelle page

Dans l'invite de commande saisissez :

- ionic generate
- type page
- name : Page2

Pour naviguer entre les vues

Pour la navigation, Ionic utilise la librairie « router ». pour l'utiliser il faut :

- Importer la librairie dans home : `import { Router} from '@angular/router';`
- Déclarer la librairie dans constructor : `private router: Router`
- Charger la Page2 : `this.router.navigate(['/page2']);`

```
import { Component } from '@angular/core';
import { Router} from '@angular/router';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  MonText="Bonjour";

  constructor(private router: Router) {}

  MonClick(){
    this.router.navigate(['/page2']);
  }
}
```

Pour passer des paramètres entre 2 vues

Pour échanger des paramètres entre les vues, Ionic utilise la librairie « NavigationExtras ». Pour l'utiliser il faut :

Dans la vue émettrice (home.page.ts) :

- Importer la librairie dans home : `import { NavigationExtras } from '@angular/router';`
- Déclarer les paramètres :


```
let navigationExtras: NavigationExtras = {
  state : {
    param1: « valeur »
  }
};
```
- Charger la Page2 : `this.router.navigate(['/page2'], navigationExtras);`

<p>BTS SIO2 slam4</p>	<p align="center">SLAM4 : Application mobile</p> <p align="center">Découverte d'ionic 4 et format JSON.</p>	<p align="right">Antony Santero 2019/2020</p>
---	--	--

Pour recevoir des paramètres dans une vue, Ionic utilise la librairie « ActivatedRoute ». Pour l'utiliser il faut :

Dans la vue réceptrice (page2.page.ts) :

- Importer la librairie dans home : `import { ActivatedRoute } from '@angular/router'`;
- Déclarer la librairie dans constructor : `private router: Router`
- Déclarer la librairie dans constructor : `private activeRoute : ActivatedRoute`

```
import { Component } from '@angular/core';
import { Router, NavigationExtras } from '@angular/router';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  MonText="Bonjour";

  constructor(private router: Router) {}

  MonClick(){
    let navigationExtras: NavigationExtras = {
      state : {
        param1: this.MonText
      }
    }
    this.router.navigate(['./page2'],navigationExtras);
  }
}
```

```
import { Component, OnInit } from '@angular/core';
import { Router,ActivatedRoute} from '@angular/router';
@Component({
  selector: 'app-page2',
  templateUrl: './page2.page.html',
  styleUrls: ['./page2.page.scss'],
})
export class Page2Page implements OnInit {

  MonText="";

  constructor(private router: Router,
    private activeRoute : ActivatedRoute) {
    this.activeRoute.queryParams.subscribe(params => {
      if (this.router.getCurrentNavigation().extras.state) {
        this.MonText =
          this.router.getCurrentNavigation().extras.state.param1;
      }
    });
  }

  ngOnInit() {
  }
}
```

Utilisation des boucles d'affichage

Pour répéter un affichage, Ionic utilise Angular (***ngFor**):

- Dans le fichier (contrôleur) home.page.ts, déclarez une liste JSON
- Dans le fichier html, utilisez la fonction *ngFor

```
import { Component } from '@angular/core';
import { Router, NavigationExtras } from '@angular/router';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  Maliste = [
    {nom : "Kirya", prenom:"Jean"},
    {nom : "Quiroule", prenom:"Pierre"},
    {nom : "Dutocque", prenom:"Yvan"}
  ]

  constructor(private router: Router) {}
}
```

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Ionic Blank
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
<div *ngFor="let item of Maliste">
  <p>{{item.nom}}</p>
  <p>{{item.prenom}}</p>
</div>

</ion-content>
```

Utilisation des listes

- Dans le fichier html, ajoutez une liste avec les balises `<ion-list>...</ion-list>`
- Une liste est composée de `<ion-item>...</ion-item>` répétés en boucle
- Ajoutez l'évènement click a `<ion-item ... (click)= « MonClick(item) »>`
- Dans le contrôleur, ajoutez la fonction **MonClick(item)**

```
import { Component } from '@angular/core';
import { Router, NavigationExtras } from '@angular/router';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {
  MaListe = [
    {nom : "Kirya", prenom:"Jean"},
    {nom : "Quiroule", prenom:"Pierre"},
    {nom : "Dutocque", prenom:"Yvan"}
  ]

  constructor(private router: Router) {}

  MonClick(item){
    console.log(item);
  }
}
```

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Ionic Blank
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
<ion-list>
  <ion-item *ngFor="let item of MaListe"
    (click)="MonClick(item)">
    <p>{{item.nom}}</p>
    <p>{{item.prenom}}</p>
  </ion-item>
</ion-list>
</ion-content>
```

Utilisation des conditions d'affichage

Pour gérer les conditions d'affichage, Ionic utilise Angular (***ngIf**):

- Dans le fichier (contrôleur) home.page.ts, déclarez la variable de condition dans la liste JSON
- Dans le fichier html, utilisez la fonction ***ngIf**

```
import { Component } from '@angular/core';
import { Router, NavigationExtras } from '@angular/router';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {
  MaListe = [
    {nom : "Kirya", prenom:"Jean",sexe:0},
    {nom : "Tournelle", prenom:"Marie",sexe:1},
    {nom : "Dutocque", prenom:"Yvan",sexe:0}
  ]

  constructor(private router: Router) {}

  MonClick(item){
    console.log(item);
  }
}
```

```
<ion-content>
<ion-list>
  <ion-item *ngFor="let item of MaListe"
    (click)="MonClick(item)">
    <ion-avatar *ngIf="item.sexe==0" slot="start">
      
    </ion-avatar>
    <ion-avatar *ngIf="item.sexe==1" slot="start">
      
    </ion-avatar>
    <ion-label>
      <p>{{item.nom}}</p>
      <p>{{item.prenom}}</p>
    </ion-label>
  </ion-item>
</ion-list>
</ion-content>
```

<p>BTS SIO2 slam4</p>	<p align="center">SLAM4 : Application mobile</p> <p align="center">Découverte d'ionic 4 et format JSON.</p>	<p align="right">Antony Santero 2019/2020</p>
---	--	--

Utilisation des formulaires

Tout comme en HTML, Ionic utilise des formulaires en méthode POST

- Dans le fichier html, créez votre formulaire avec vos paramètres
- Dans le fichier (contrôleur) home.page.ts, créez une fonction pour exploiter ces paramètres

```

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

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  user={};

  constructor() {}

  logForm() {
    console.log(this.user)
  }
}

```

```

<ion-content>
  <form (ngSubmit)="logForm()">
    <ion-item>
      <ion-label>Login</ion-label>
      <ion-input type="text" [(ngModel)]="user.login" name="login">
    </ion-item>
    <ion-item>
      <ion-label>Password</ion-label>
      <ion-input type="text" [(ngModel)]="user.mdp" name="mdp">
    </ion-item>
    <button ion-button type="submit" block>Valider</button>
  </form>
</ion-content>

```

Utilisation des api et requêtes HTTP

Pour communiquer avec le monde extérieur, Ionic utilise la librairie **http**. Avant de pouvoir être utilisée, elle doit être déclarée dans le fichier **app.modules.ts**

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
import { HttpClientModule } from '@angular/common/http';

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule, HttpClientModule],
  providers: [
    StatusBar,
    SplashScreen,
    { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
  ],
  bootstrap: [AppComponent]
})
export class AppModule {}

```

BTS SIO2 slam4	SLAM4 : Application mobile Découverte d'ionic 4 et format JSON.	Antony Santero 2019/2020
---------------------------------	---	------------------------------------

Utilisation des requêtes HTTP GET

Pour utiliser google chrome, vous devez installer le plug-ins **CORS** dans google chrome. **!!! IMPORTANT**

- Utilisez la fonction **http.get** :


```

      this.http.get("URL")
        .subscribe(results => {
          /* utilisation de results */
        });
      
```
- Subscribe** attend l'arrivée asynchrone des données

```

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

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  ListeArret:any;
  constructor(private http: HttpClient) {

    this.http.get("http://open.tan.fr/ewp/arrets_json/47,21661/-1,556754")
      .subscribe(results => {
        this.ListeArret=results;
      });
  }
}

```

```

<ion-content>
<ion-list>
  <ion-item *ngFor="let item of ListeArret">
    <p>{{item.libelle}}</p>
    <p>{{item.distance}}</p>
  </ion-item>
</ion-list>
</ion-content>

```

Utilisation des requêtes HTTP POST

Pour utiliser google chrome, vous devez installer le plug-ins **CORS** dans google chrome. **!!! IMPORTANT**

- Utilisez la fonction **http.post** :


```

      this.http.post("URL", { JSON }).subscribe(results => {
          /* utilisation de results */
        });
      
```
- Subscribe** attend l'arrivée asynchrone des données

```

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

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {

  user={login:"",mdp:""};
  constructor(private http: HttpClient) {}

  logForm(){
    this.http.post("http://127.0.0.1/login",this.user)
      .subscribe(results => {
        /*...*/
      });
  }
}

```

```

<ion-content>
<form (ngSubmit)="logForm()">
  <ion-item>
    <ion-label>Login</ion-label>
    <ion-input type="text" [(ngModel)]="user.login" name="login">
  </ion-input>
  </ion-item>
  <ion-item>
    <ion-label>Password</ion-label>
    <ion-input type="text" [(ngModel)]="user.mdp" name="mdp">
  </ion-input>
  </ion-item>
  <button ion-button type="submit" block>Valider</button>
</form>
</ion-content>

```