



# Tutorial des ateliers des Journées du WebMaster Universitaire

Djerba, du 10 au 13 Mai 2017

## Intervenants

- **M. Béchir Hamidi** (bechir.hamidi@gmail.com)
- **M. Mejdi BLAGHGI** (m.blaghi@gmail.com)
- **Mme Mouna KORT** (kort\_mouna@yahoo.fr)
- **Mlle Ratiba BEN AZZOUN** (ratiba.ben.azzoun@gmail.com)

## Atelier 1 : Mise en place de l'environnement de travail

### Objectifs 🎯

- Installer et configurer les outils de développement des différentes interfaces de l'application
- Mettre en place la plateforme de test

### Étapes 📌

1. Installation de node.js
2. Installation des frameworks Cordova et Ionic
3. Installation de la plateforme d'application, Android SDK
4. Création d'un projet avec CLI ionic
5. Exploration de l'arborescence d'un projet ionic

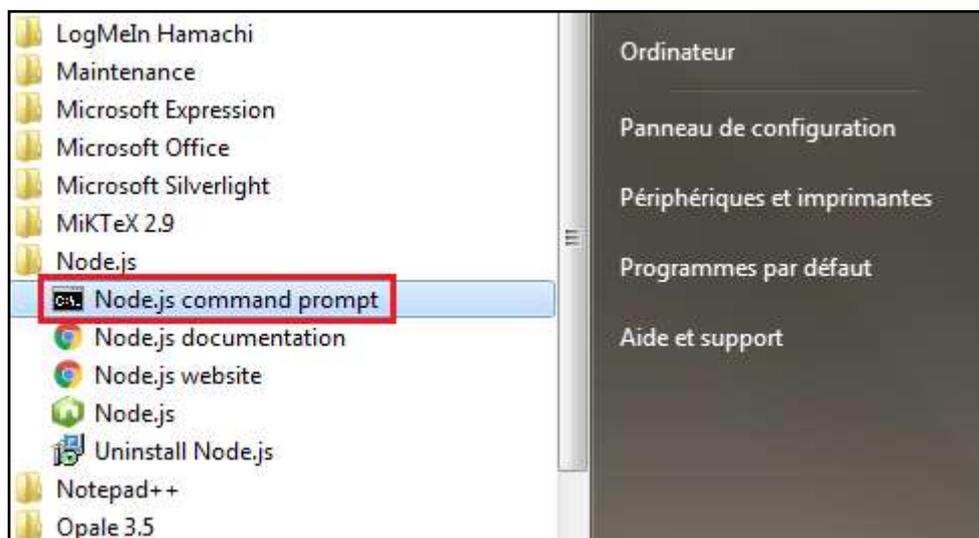
### Manipulation ✂

#### Installation de node.js

- a. Accéder au site <https://nodejs.org> et télécharger la version **v7.9.0 LTS** de node.js, 32bits ou 64bits selon votre PC.

*node.js* : c'est une plateforme logicielle libre et événementielle en JavaScript. Autrement, c'est un environnement d'assez bas niveau permettant d'exécuter du JavaScript non plus dans le navigateur web mais sur le serveur. Donc c'est un langage côté serveur.

- b. Vérifier que nodejs est ajouté dans **la variable d'environnement PATH**
- c. À partir de votre menu « Programme » accéder à « Node.js », une invite de commandes apparaît. Lancer cette invite en tant qu'Administrateur.



- d. Vous pouvez taper les commandes « `node -v` » et « `npm -v` », afin de s'assurer de la bonne installation du **node** et **npm**.

```
C:\Users\DGI>npm -v
3.10.10

C:\Users\DGI>node -v
v6.10.3
```

npm : c'est le gestionnaire officiel des paquets de Node.js. Il fait partie de l'environnement et est automatiquement installé par défaut.

### Installation des frameworks Cordova et Ionic

- a. Exécuter les commandes « `npm install -g cordova` » puis « `npm install -g ionic` », afin d'installer **Cordova** et **Ionic**. Assurez-vous qu'une connexion internet existe. Ceci va prendre quelques minutes !!

ionic : c'est un framework de développement qui permet de créer des applications hybrides en HTML5, CSS, Javascript. Il est basé sur des frameworks/technologies telles qu'**AngularJS** et **Apache Cordova**.

cordova: c'est un framework open-source. Il permet de créer des applications pour différentes plateformes. C'est un pont de développement qui permet d'encapsuler du code client Web (Ionic/AngularJS) dans une application native telle qu'Android, IOS, ... . Il permet par ailleurs d'accéder aux fonctionnalités natives des appareils, telle que la caméra, la géolocalisation, ...

- b. Vous pouvez taper les commandes « `cordova -v` » et « `ionic -v` », afin de s'assurer de la bonne installation de **Cordova** et **Ionic**.

```
C:\Users\DGI>cordova -v
7.0.0

C:\Users\DGI>ionic -v
2.2.3
```

### Installation de la plateforme Android SDK

À fin de développer une application pour appareils Android, il est indispensable de disposer des outils Android tel que le **SDK Android**. Si l'on souhaite cibler l'application pour d'autres systèmes d'exploitation, il est là aussi indispensable de disposer du SDK de la plateforme visée. Ces outils serviront à la compilation et à la construction de l'application.

- a. Commençons par installer Android studio à partir de ce site, selon votre système d'exploitation.
- b. Installer le JDK

- c. Une fois l'environnement prêt, nous pouvons générer les fichiers nécessaires à la plateforme désirée, en tapant la commande « `ionic platform add android` » ou « `ionic platform add ios` »
- d. Vous pouvez tester votre application soit sur un émulateur, soit sur un périphérique réel. Il est toujours préférable de tester sur de vrais smartphones et tablettes, notamment à cause de la lenteur et l'imperfection des émulateurs fournis.
- e. Ainsi pour émuler vous pouvez utiliser les commandes : « `ionic built android` » ou « `ionic built ios` » selon votre plateforme.
- f. La commande « `ionic run android` » lance la version debug de notre application sur le premier périphérique trouvé, et si aucun n'est trouvé sur un émulateur.

### Création d'un projet avec CLI Ionic

La CLI (Command Line Interface) contient un certain nombre de commandes cruciales pour le développement Ionic. Les commandes les plus utilisées sont décrites par ce tableau.

Commande	Description
info	Affiche des informations utiles sur l'environnement Ionic et les dépendances de votre système (version Ionic, Cordova, node,...)
start	Crée un nouveau projet Ionic à base d'un <b>modèle</b> et dans le chemin spécifié
serve	Démarre le serveur de développement local pour tester vos applications sur le navigateur local. Cette commande surveille les changements dans les fichiers et actualise automatiquement l'affichage dans le navigateur
build	Crée une application pour une plate-forme spécifique ( iOS ou Android)
emulate	Émule un projet Ionic sur un simulateur ou un émulateur
Run	Exécute un projet Ionic sur un périphérique connecté

La syntaxe de la commande start est : « `ionic start nomApp [template] [path]` ».

[template] : Détermine le modèle du projet. Ses valeurs possibles sont :

- blank : crée un projet vide avec une seule page
- tabs : crée un projet contenant 3 onglets (par défaut)
- sidemenu: crée un projet à base de menu extensible sur l'un des côtés droit ou gauche
- maps : crée un projet pour intégrer une carte de géolocalisation
- super : crée un projet contenant 14 pages différentes
- tutorial: crée un projet pour concevoir un tutoriel

[path] : Détermine le chemin du répertoire du projet, par défaut le répertoire courant.

- a. Taper la commande « `ionic start -list` » fin d'afficher les templates installées sous votre système.

```

C:\Users\DGI>ionic start --list
Downloading Starter Templates - http://code.ionicframework.com/content/starter-templates.json
Ionic Starter templates

blank ..... A blank starter project for Ionic
complex-list ..... A complex list starter template
maps ..... An Ionic starter project using Google Maps and a side menu
salesforce ..... A starter project for Ionic and Salesforce
sidemenu ..... A starting project for Ionic using a side menu with navigation in the content area
tabs ..... A starting project for Ionic using a simple tabbed interface

```

- b. Se placer dans le répertoire de l'application à développer
- c. Exécuter la commande « `ionic start AppJWMU sidemenu --v1` » qui permettra de créer la structure d'un projet Ionic à partir d'une template de menu extensible. Ceci prend quelques minutes !

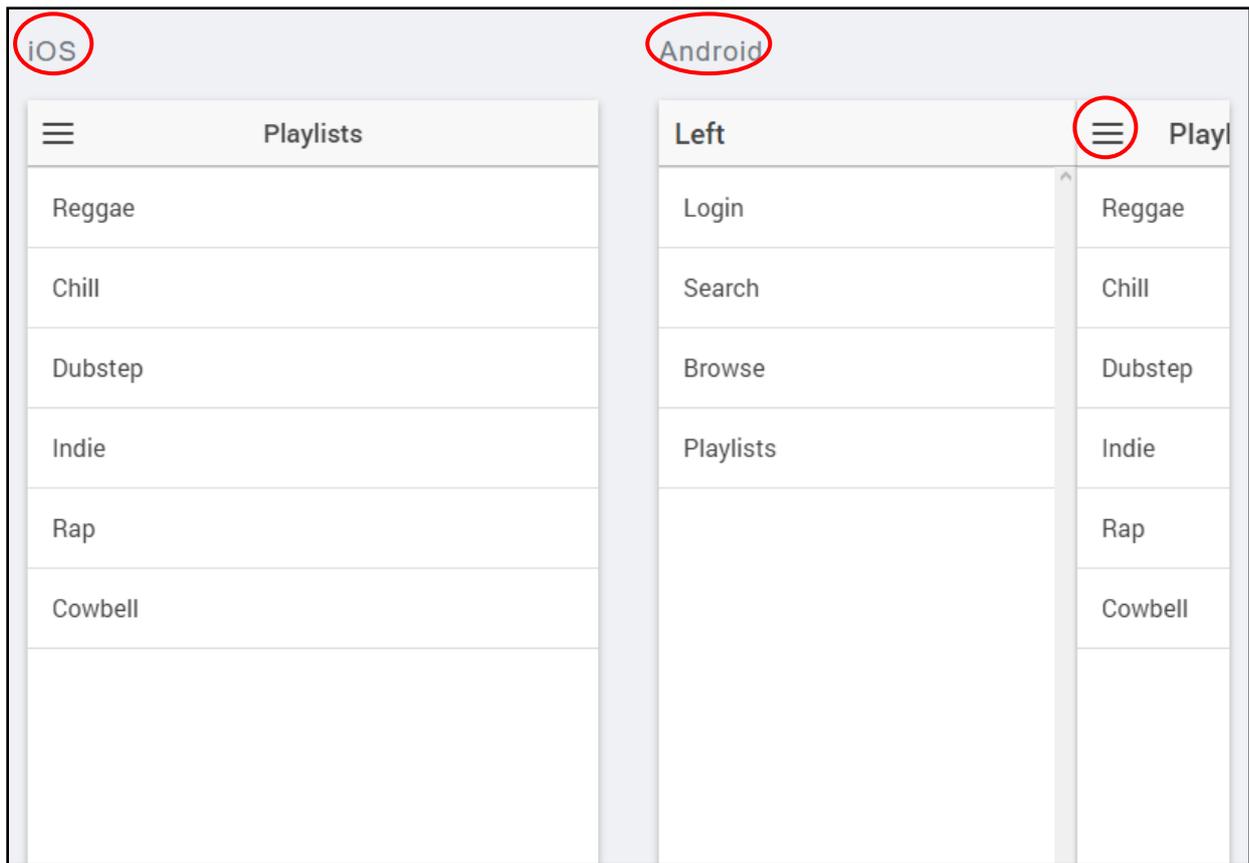
```

C:\Users\DGI>ionic start AppJWMU sidemenu
*****
Dependency warning - for the CLI to run correctly,
it is highly recommended to install/upgrade the following:

Please install your Cordova CLI to version >=4.2.0 'npm install -g cordova'
*****
Creating an Ionic 2.x app in C:\Users\DGI\AppData\Local\Temp\ionic2-app-base based on the sidemenu template
*
Downloading: https://github.com/driftyco/ionic2-app-base/archive/master.zip
Downloading: https://github.com/driftyco/ionic2-starter-sidemenu/archive/master.zip
Installing npm packages (may take a minute or two)...
? ? ? ? Your Ionic app is ready to go! ? ? ? ?
Some helpful tips:
Run your app in the browser (great for initial development):
  ionic serve
Run on a device or simulator:
  ionic run ios[android,browser]
Share your app with testers, and test on device easily with the Ionic View companion app:
  http://view.ionic.io
Create a free Ionic account to share and test apps with Ionic View?
(Y/n): n
C:\Users\DGI>

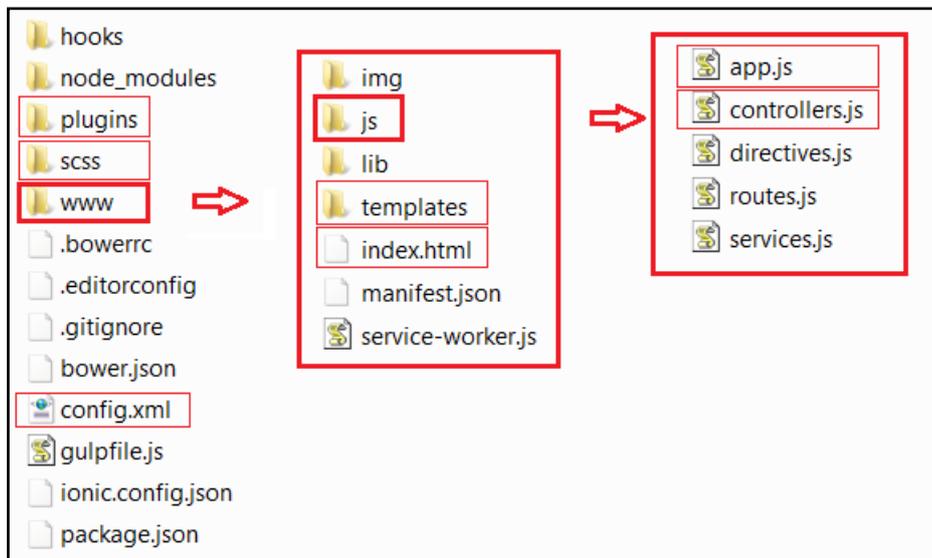
```

- d. Taper « `cd nomApp` » pour accéder au répertoire de l'application.
- e. Taper la commande « `ionic serve` » pour tester les interfaces de l'application dans le navigateur local. La commande « `ionic serve --lab` » affiche sur le navigateur deux aperçus, un pour iOS et un autre pour Android. À titre d'exemple on pourra avoir l'aperçu suivant.



## Exploration de l'arborescence d'un projet ionic

Une fois créé, le répertoire de votre projet à la structure suivante.



### Description des répertoires et fichiers

#### ○ 1<sup>er</sup> niveau

- **plugins** : contient les plugins cordova utilisés lors de développement de votre application (géolocalisation, accès à la caméra...)

- **scss** : permet d'utiliser Sass (Syntactically Awesome Style Sheets) comme préprocesseurs CSS pour générer les styles de vos applications
- **config.xml** : permet de définir certaines propriétés de l'application (name, description, author,...)
- **www** : contient tous les fichiers et les répertoires qui définissent votre application

#### ○ 2<sup>ème</sup> niveau

- **index.html** : c'est le point d'entrée de l'application. Dans ce fichier on y importe les librairies Ionic et Cordova, nos propres script JavaScript et on y définit les styles CSS.
- **templates** : ou views, ce répertoire contient toutes les pages HTML de votre application (onglets, menu, ...)
- **js** : ce répertoire contient les fichiers JavaScript de l'application.

#### ○ 3<sup>ème</sup> niveau

- **apps.js** : c'est le fichier JS de configuration et de gestion de l'application.
- **controllers.js** : Les contrôleurs sont des fonctions JavaScript qui relient les pages html (Vues) avec leurs données (Modèle) tout en ajoutant une logique à ces pages.

## Atelier 2 : Création d'un projet avec Ionic Creator

### Objectifs

- S'initier avec l'éditeur en ligne Ionic Creator
- Créer les premières pages de l'application mobile Hybride "AppJWMU"

### Étapes

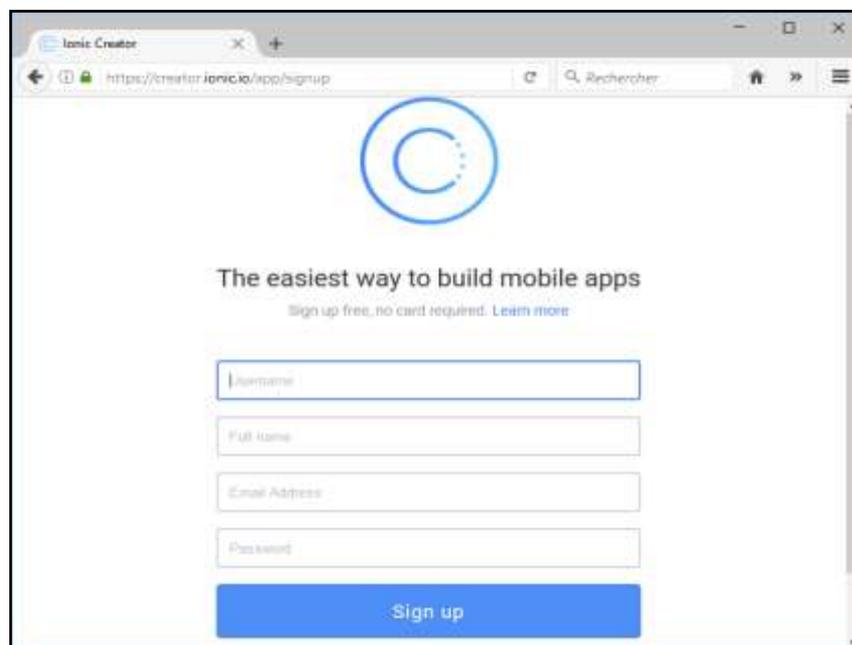
1. Inscription sur le site de Ionic Creator
2. Se familiariser avec l'éditeur Ionic Creator
3. Création des pages du projet AppJWMU
4. Exportation et intégration du projet
5. Edition des pages

### Manipulation

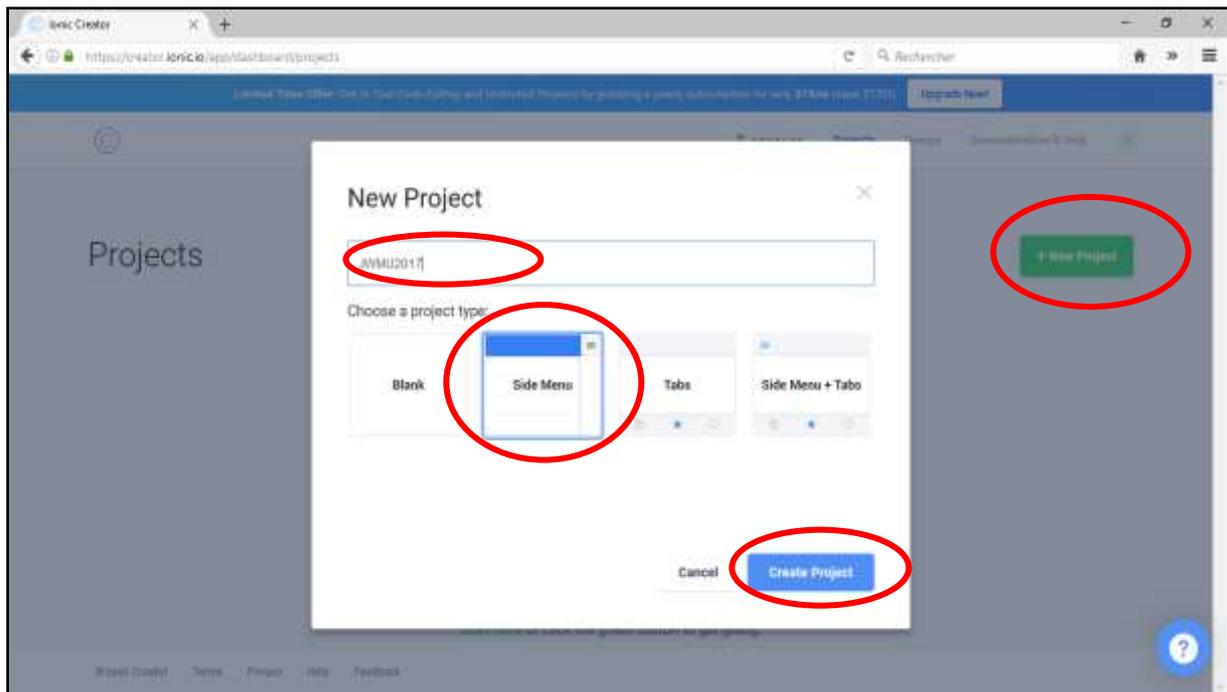
#### 1) Inscription sur le site de Ionic Creator

- a. Créer un compte sur le site <https://creator.ionic.io/>

*Ionic Creator* : est un assistant à la création d'applications Ionic, permettant de créer des interfaces graphiques via de simples "drag and drop" puis de créer votre projet Ionic grâce à l'identifiant de votre projet "creator". Ce service est cependant encore en version Alpha.

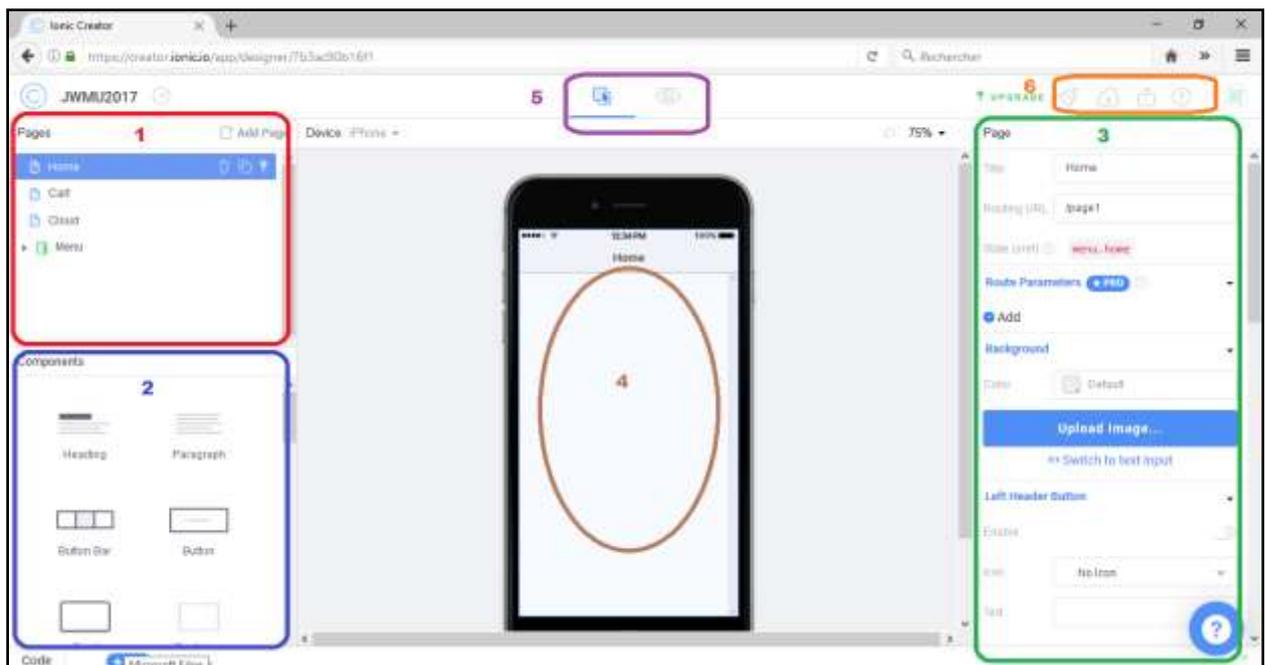


- b. Une fois inscrit, il faut créer un nouveau projet Ionic et choisir sa template. Pour notre application, créer un projet intitulé **AppJWMU** avec une template de type **side Menu**.



## 2) Se familiariser avec Ionic Creator

L'interface principale d'Ionic Creator est décrite comme suit:

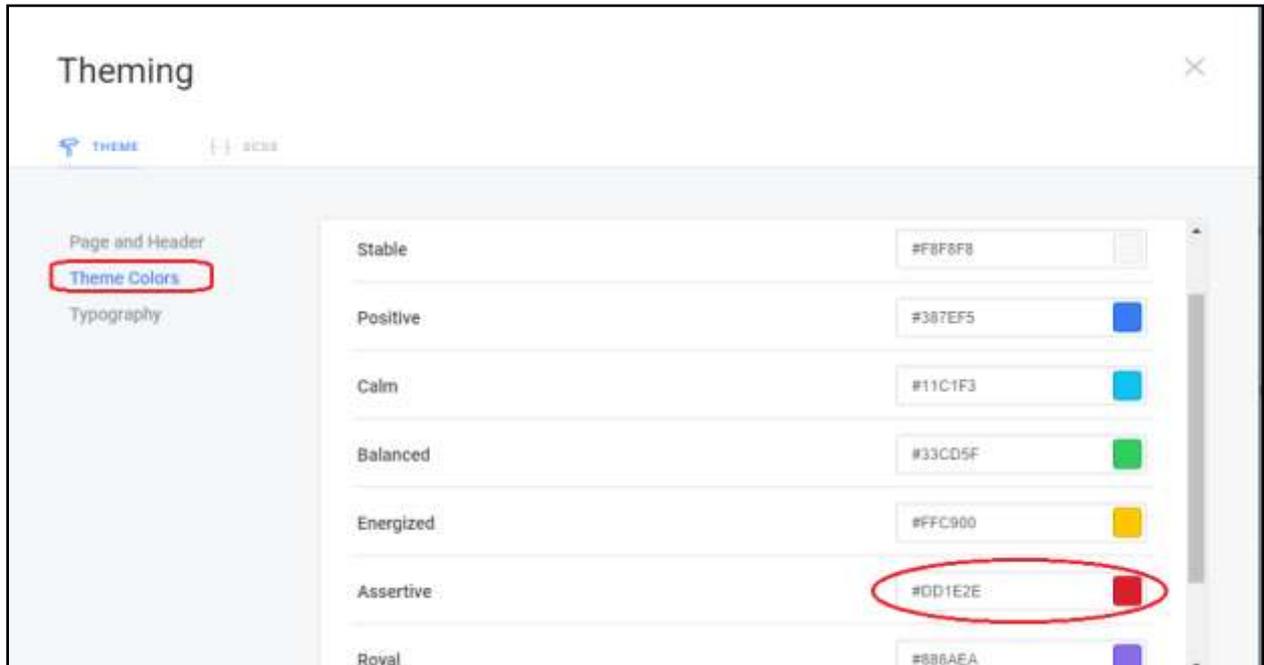


- 1: Arborescence du projet (ensemble des pages HTML créés avec leurs composants)
- 2: Les différents outils d'insertion ionic
- 3: Bloc de paramétrage de chaque page et de chaque composant (name, title, lien, icon, route, style, etc)
- 4: Zone d'affichage
- 5: Modes d'affichage du projet (mode Edition et mode Visualisation)
- 6: Fonctionnalités de style, d'export et de partage du projet créé

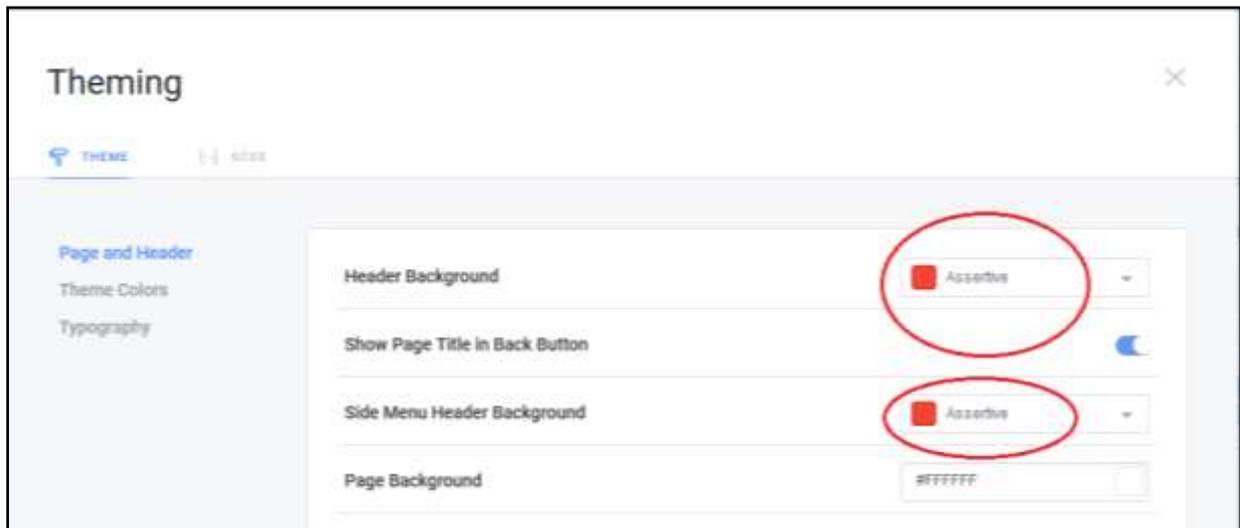
### 3) Création des pages du projet APPJWMU

a. Commencer par changer le thème de l'application : choisir une entête et un pied de page de couleur rouge (code: #DD1E2E)

b. Cliquer sur cette icône  (en haut à droite), accéder à **THEME--> Theme Colors** et changer le code de "assertive" avec le code hexadécimal #DD1E2E.

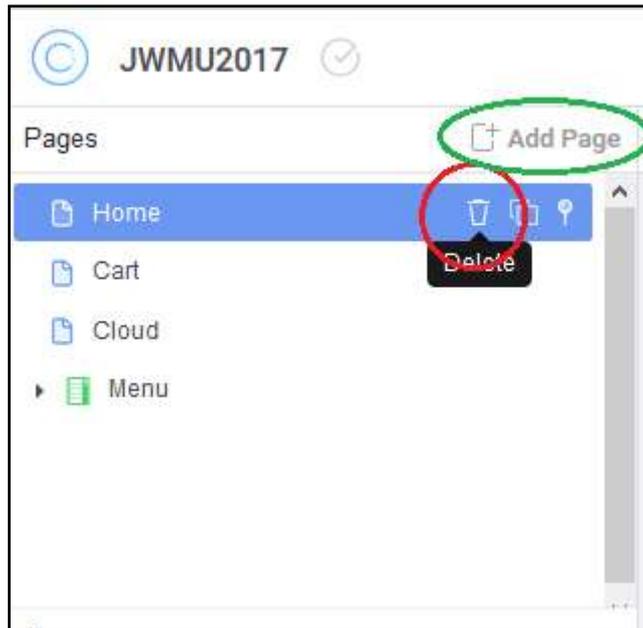


c. Cliquer sur l'onglet **Page and Header** et sélectionner la couleur "assertive" pour l'arrière plan de l'entête.



Le type de template choisi pour ce projet (sidemenu) ne fournit pas un pied de page.

d. Supprimer toutes les pages créés par défaut dans le projet



- e. Créer les nouvelles pages intitulées Accueil, A propos, Actualités, Programme, Organismes, Inscription, Galerie de photos et Contact.
- f. Nommer chaque page créée dans la zone à droite en donnant le même nom pour **Title** et **Routing URL**.



- g. Dans l'entête de chaque page, ajouter une icone à droite.
- h. Marquer la page d'Accueil comme page par défaut pour l'application.

Les noms des icones utilisées:

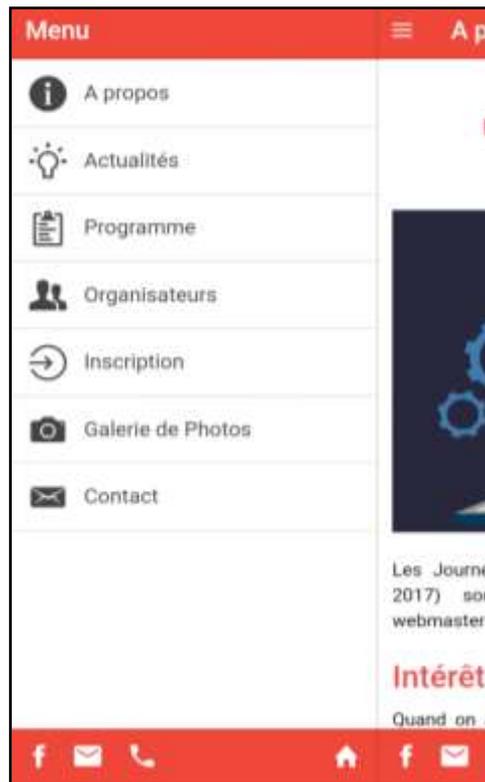
**A propos:** ion-information-circled

**Actualités:** ion-android-bulb

**Programme:** ion-clipboard

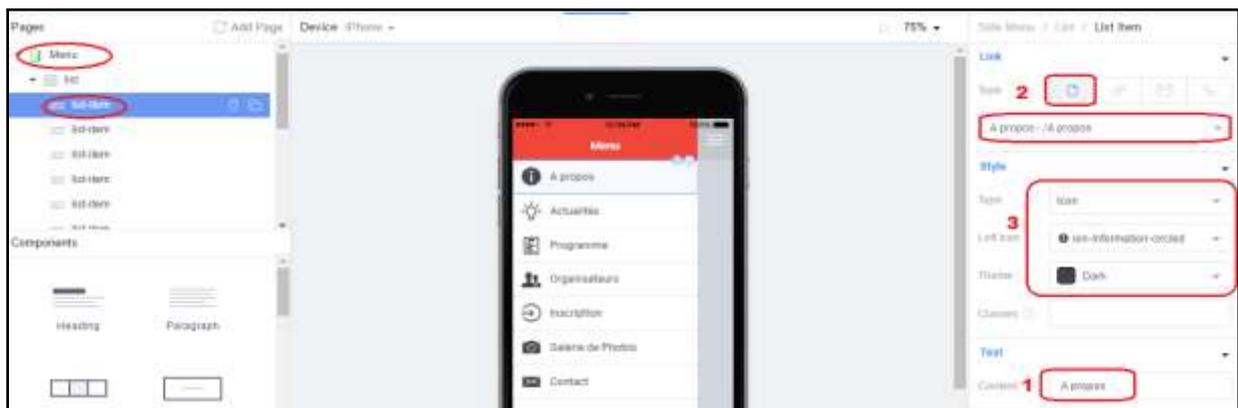
**Organismes:** ion-person-stalker

## Page Menu



a. Revenir à la page Menu pour créer les items comme suit:

- Sélectionner le premier item dans la liste de menu et le paramétrer en changeant son contenu (1), lui associer la page correspondante(2) parmi les pages créées et lui donner une icône convenable (3)
- faire le même travail pour tous les items de Menu.

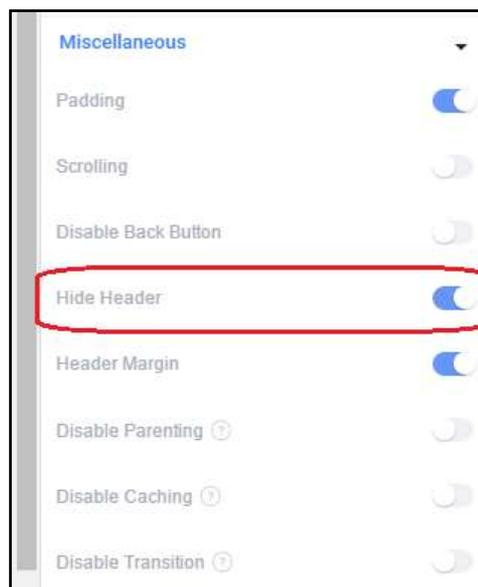


b. Utiliser le bouton du mode Visualisation  pour simuler le fonctionnement de l'application sur un device.

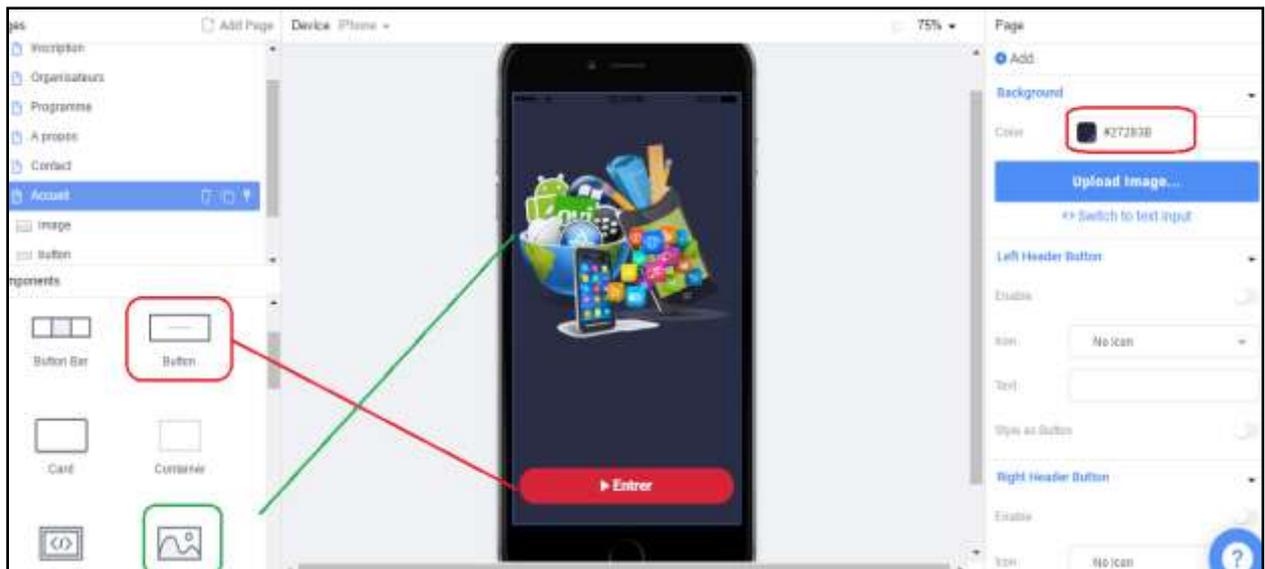
## Page d'Accueil



- a. la page d'accueil est la première page de notre application qui ne contient pas d'entête ni du pied. Pour-cela, il faut désactiver le Header dans la partie inférieure du bloc de paramétrage.



- b. Ensuite, changer la couleur d'arrière plan et insérer deux composants : une image et un bouton.



- c. Uploader l'image appropriée lui donner une classe nommée **imgA** (classe CSS qu'on va définir dans le **theming**)
- d. Accéder à theming -->SCSS et copier le bout de code suivant:

```
.imgA {
  display: block;
  width:100%;
  height:100%;
  margin-left:auto;
  margin-right:auto;
  margin-bottom:100px;
  margin-top:50px
}
```

- e. Paramétrer le bouton **Entrer** comme suit:

Page / Button

Link

Type    

A propos - /A propos

Text

Entrer

Size 20 px

Align    

Weight 600 - Semi B  

Color  Default

Style

Width Block

Type Default

Theme  Assertive

Size Standard

Radius  50 px

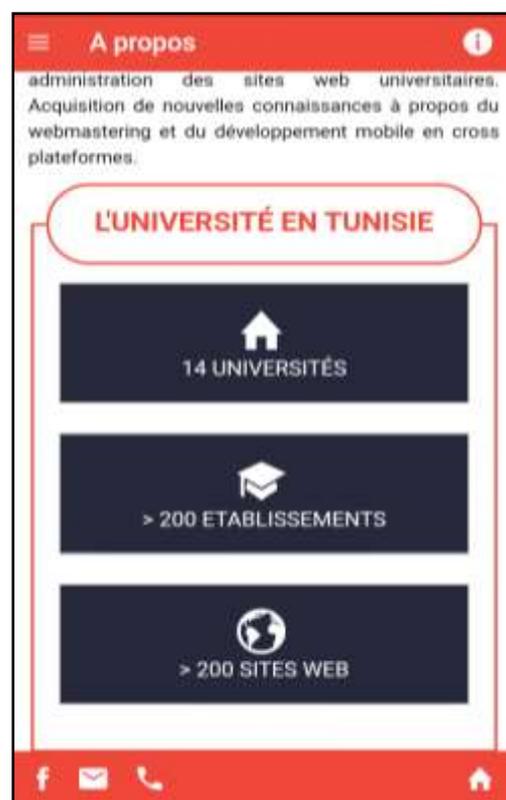
Classes ?

Icon

ion-ios-play

Position   

**Page A propos**



a. Insérer les différents composants de la page A propos comme suit:



Consulter le site <http://www.isetib.rnu.tn/jwmu2017/> pour compléter le contenu de la page.

b. Le slider est composé de 3 slides avec des images et modifier le style CSS comme suit:

```
.slider {
  width:100%;
  height:50%;
  background-color:#27283B;}
```

c. la partie inférieure de la page **A propos** sera programmée en HTML ultérieurement.



## Page Programme



a. Ajouter les composants de la page comme suit :



b. Donner la classe **item-text-wrap** pour chaque élément pour afficher le texte complet.



## Page Organisateurs



- a. Ajouter les différents composants de type **thumbnail** qui doivent être des liens hypertextes vers les sites web des organisateurs (utiliser le bouton  pour insérer un lien)

ANSI : <https://www.ansi.tn/>

ISET DJERBA : <http://www.isetjb.rnu.tn>

ADST: <http://www.isetjb.rnu.tn/fr/adst/presentation-adst.html>

CCK : <http://www.cck.rnu.tn>

IRESA : <http://www.iresa.agrinet.tn>



## Page Inscription

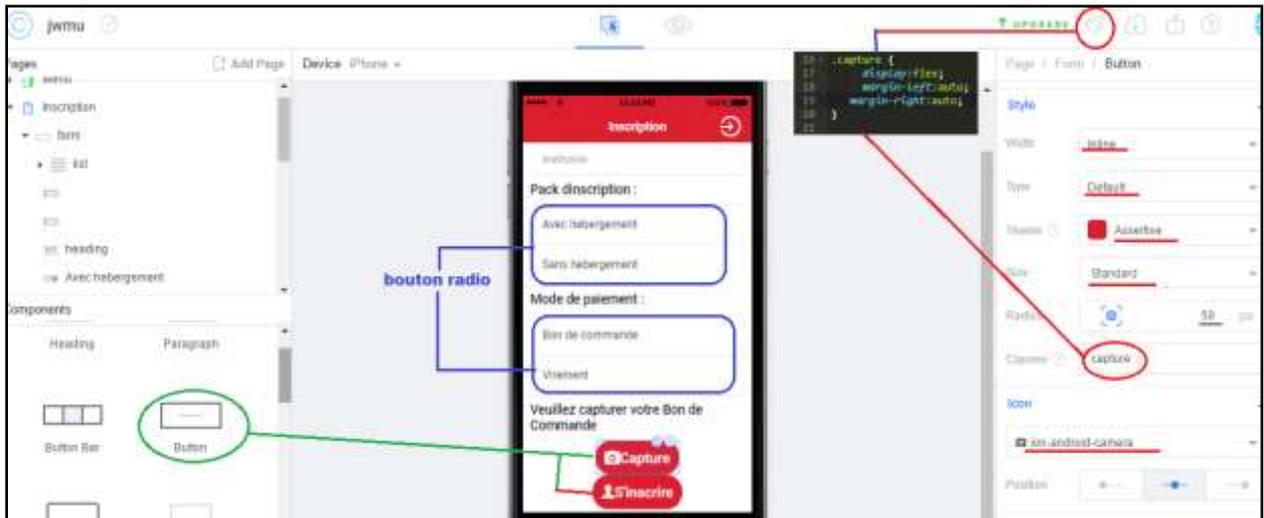
The screenshot shows a mobile application interface for registration. At the top, there is a red header with a hamburger menu icon, the title "Inscription", and a refresh icon. Below the header, there is a form with several input fields: "Email", "Nom", "Prenom", "Telephone", "Profession", and "Institution". Underneath these fields, there are two sections: "Pack d'inscription :" and "Mode de paiement :". The "Pack d'inscription" section has two radio button options: "Avec hebergement" and "Sans hebergement". The "Mode de paiement" section has a radio button option: "Bon de commande". At the bottom of the screen, there is a red navigation bar with icons for Facebook, email, phone, and home.

The screenshot shows a mobile application interface for payment capture. At the top, there is a red header with a hamburger menu icon, the title "Inscription", and a refresh icon. Below the header, there is a form with two input fields: "Bon de commande" and "Virement". Below these fields, there is a text prompt: "Veillez capturer votre bon de commande". Underneath the prompt, there is a red button with a camera icon and the text "Capture". Below the button, there is a large gray square placeholder with the text "300x300". At the bottom of the screen, there is a red navigation bar with icons for Facebook, email, phone, and home, and a red button with a person icon and the text "S'inscrire".

a. La page Inscription est composée de:

- un composant de type **form** qui englobe tous les éléments.
- plusieurs composants de type **input** (pour chaque élément, donner son type et un placeholder)
- 2 groupes de boutons radio ( **Pack d'inscription** et **Mode de Paiement**), donner le même nom de catégorie pour chaque groupe



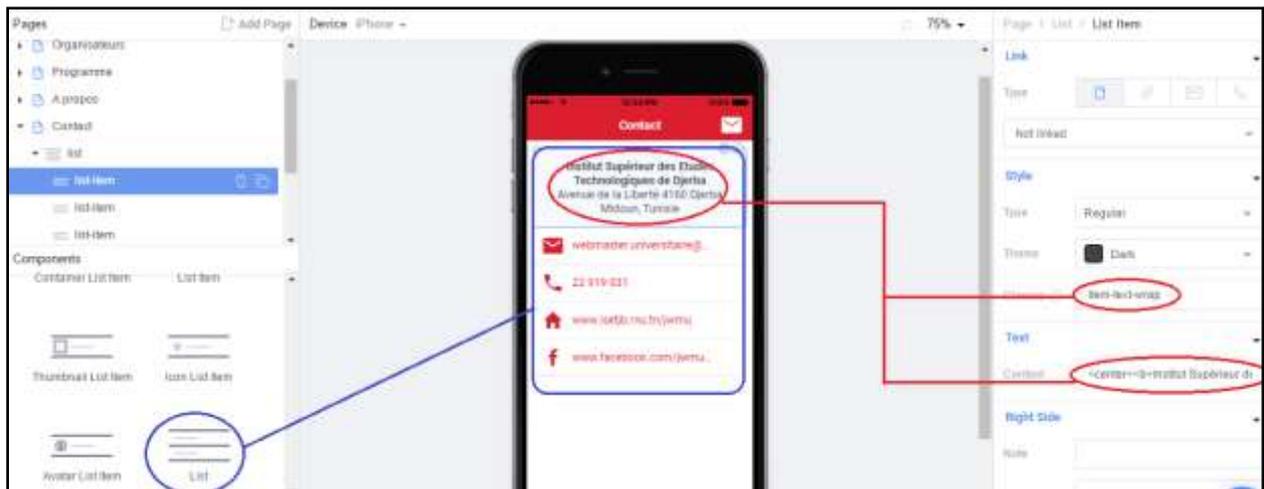


Les actions des deux boutons Capture et S'inscrire (Click) seront développées dans les prochains ateliers.

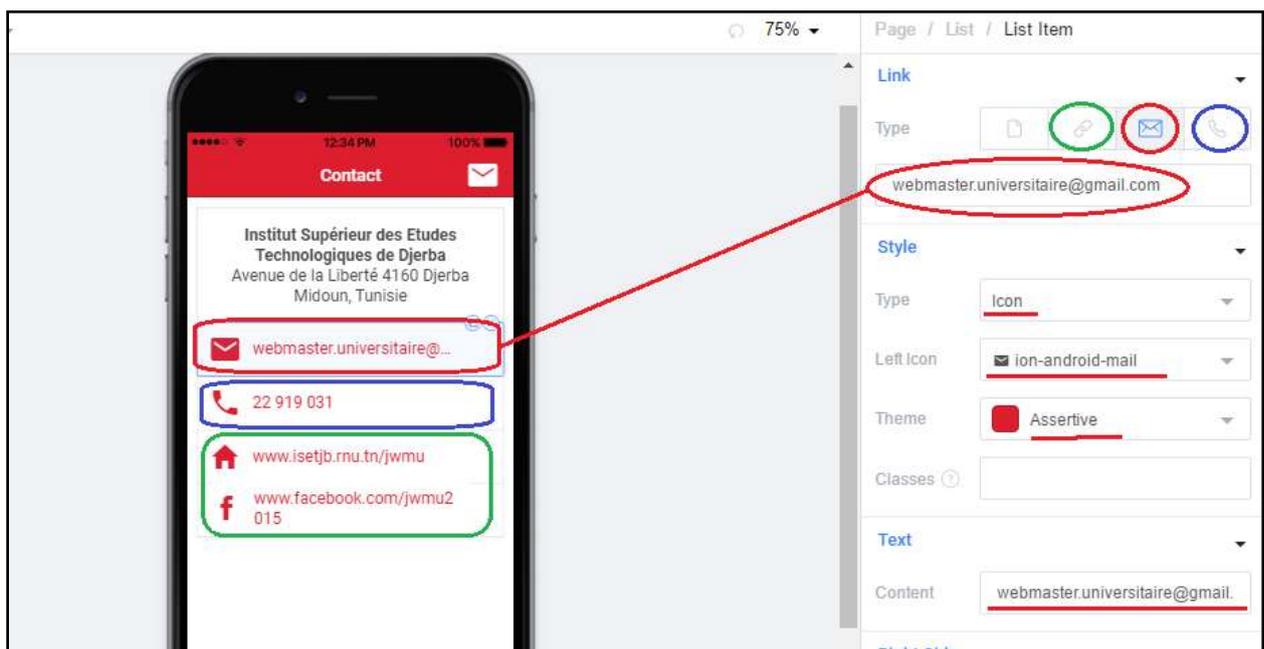
### Page Contact



- a. la page Contact est composée d'une liste d'items:
  - le premier item (nom et adresse de ISET de Djerba) possède la classe **item-text-wrap**. Pour le style gras et le retour à la ligne, insérer les balises **<b>** et **<br>** dans le contenu du texte



- le mail est un élément lien qui pointe sur l'adresse électronique du séminaire (webmaster.universitaire@gmail.com)
- le téléphone est un élément lien qui permet d'appeler le numéro précisé.
- le site web et la page facebook sont des liens hypertextes vers les pages suivantes : <http://www.isetjb.rnu.tn/jwmu2017> et <http://www.facebook.com/jwmu2015>



Une partie de géolocalisation sera détaillée ultérieurement dans un prochain atelier.

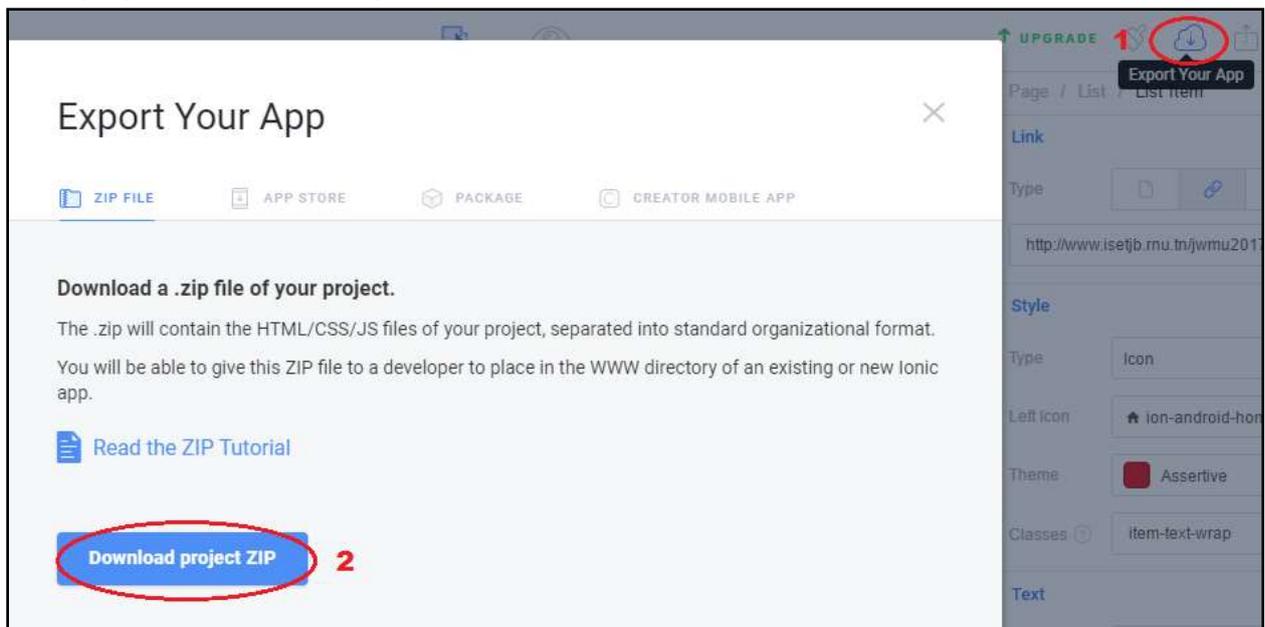
### **Page Galerie de Photos et Actualités**

Cette partie sera décrite ultérieurement dans l'atelier "Accès à une base de donnée MySQL. Le contenu des pages Galerie de Photos et Actualités s'affiche via une connexion à la base de données.

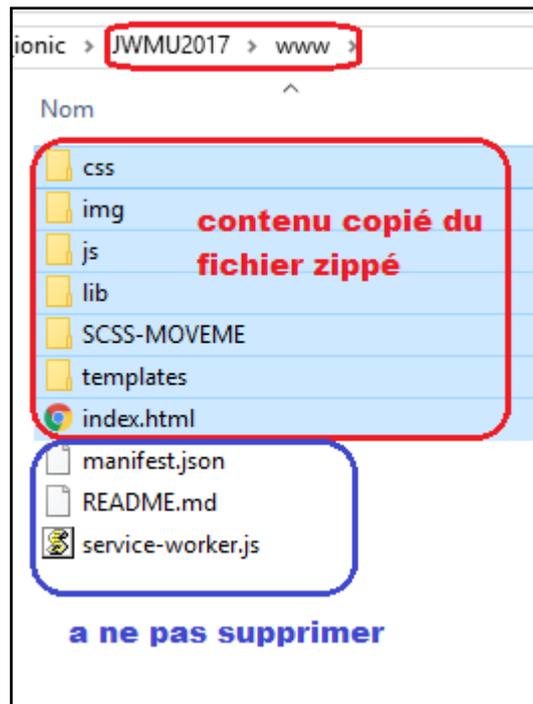


#### 4) Exportation et intégration du projet

- a. Après la création de toutes les pages, exporter le projet **AppJWMU** pour l'intégrer au projet crée avec **node.js**.



- b. Extraire le fichier compressé et copier son contenu dans le répertoire **www** du projet initial.



c. Exécuter le projet sous **Node.js** en utilisant les commandes suivantes:

```
$ cd AppJWMU //accéder au projet AppJWMU
$ ionic platform add android //ajouter la plateforme android ou ios
$ ionic serve -l //tester le projet dans le navigateur local
$ ionic emulate android //exécuter le projet sur un emulateur
$ionic run android //exécuter le projet sur un device
```

Aller dans paramètres --> A propos de telephone-->Numéro de version et en cliquer 5 ou 7 fois.Pour tester l'application sur un smartphone , il faut activer le mode développeur

## 5) Edition des pages

Ajouter tout d'abord le pied de page suivant :



Pour cela, ajouter le code suivant après la balise `</ion-nav-view>` dans chaque page HTML sauf **Accueil.html**:

```
//création d'un footer
<ion-footer-bar class="bar bar-assertive">
// création des boutons dynamiques
<button class="button button-icon icon ion-social-facebook" href-
inappbrowser="http://www.facebook.com/jwmu2015"></button>
<button class="button button-icon icon ion-android-mail" href-
inappbrowser="mailto:webmaster.universitaire@gmail.com"></button>
<button class="button button-icon icon ion-android-call" href-
inappbrowser="tel:22919031"></button>
```

```
<button class="button button-icon icon ion-home" ui-sref="accueil"></button>
</ion-footer-bar>
```

Pour activer les liens mailto: et tel: sur le device, il faut ajouter le plugin inappbrowser avec la commande suivante : « \$cordova plugin add cordova-plugin-inappbrowser »

### Page Accueil

Si vous voulez rediriger la page d'**accueil** automatiquement à la page **A propos** sans cliquer sur le bouton Entrer, accéder au fichier **www/js/controllers.js** et ajouter la fonction suivante dans **AccueilCtrl** comme suit:

```
.controller('accueilCtrl', ['$scope', '$state', '$stateParams',
function ($scope, $state, $stateParams) {
// Redirection automatique
setTimeout(function(){
    $state.go("menu.apropos")
}, 3000);
}])
```

### Page A propos

- Pour rendre le défilement de Slider automatique avec une vitesse de 2sec, ajouter l'option **'autoplay':1000** après **options="{loop: true}** dans la balise **<ion-slides>**.
- Pour adapter les images avec la taille du Slider, changer la valeur de l'attribut **background-size:80%**; de la balise **<ion-slide-page>**
- Ajouter les composants suivants :



Pour cela, insérer le code suivant avant la balise **</ion-content>**:

```
<fieldset style="border:2px solid #EF473A">
  <legend style="font-weight:bold;border:2px solid
#EF473A;color:#EF473A;text-align:center;border-radius:40px;">L'UNIVERSITÉ EN
TUNISIE</style></legend>
  <div class="row responsive-sm">
```

```

    <div class="col">
      <ion-item class="item-icon icon ion-home light text-center"
style="background-color:#27283B;">
        <br>14 UNIVERSITÉS</ion-item>
      </div>
    <div class="col">
      <ion-item class="item-icon icon ion-university light text-center"
style="background-color:#27283B;">
        <br>> 200 ETABLISSEMENTS</ion-item>
      </div>
      <div class="col">
        <ion-item class="item-icon icon ion-earth light text-center"
style="background-color:#27283B;">
          <br>> 200 SITES WEB</ion-item>
        </div>
      </div></fieldset>

```

d. Pour augmenter la taille des icones, accéder au fichier [www/css/ionic.app.css](http://www/css/ionic.app.css) et ajouter ce code:

```

.row .col .item-icon::before {
font-size: 40px;}

```

### **Page Programme**

Pour insérer des puces devant le contenu de chaque atelier thématique, ajouter la balise `<i class="icon ion-android-arrow-dropright"></i>`

- ▶Durée : 12 heures.
- ▶Nombre de formateurs : 6 formateurs de l'ISSET de Djerba.
- ▶Mode : Ateliers en parallèle.
- ▶Nombre de participants par salle : 20 participants.

## Atelier 3 : Utilisation des plugins Cordova

### Objectifs

Se familiariser avec ngcordova

Ngcordova : une collection de plus de 70 extensions 'plugin' **AngularJS** qui facilite la création, le test et le déploiement d'applications mobiles **Cordova** avec **AngularJS**.



Toutes les fonctionnalités de base sont couvertes et sont disponibles sur **Github**. <https://github.com/>

### 1) Téléchargement

Le téléchargement de Ngcordova se fait avec le gestionnaire de paquets Bower :

Bower propose aux développeurs un gestionnaire de paquets Web JavaScript et CSS. Développé par Twitter, il permet de gérer de manière simple l'installation des paquets ainsi que leurs dépendances au sein des différents projets menés.

Deux pré-requis sont nécessaires au bon fonctionnement de **Bower** :

- **Node.js** (et son manager **NPM**, Node Packaged Modules) le site officiel Node.js
- **Git** via <http://git-scm.com/downloads>.

Commandes : "npm install -g bower" et "bower install ngCordova -save"

Une fois **ngCordova** installé, injecter ses sources dans les applications **AngularJS**, en insérant son nom dans le fichier **index.html** et la dépendance dans fichier **app.js** comme indiqué ci-dessous.

#### **index.html**

```
<script src="lib/ngCordova/dist/ng-cordova.js"></script>  
<script src="cordova.js"></script>
```

#### **app.js**

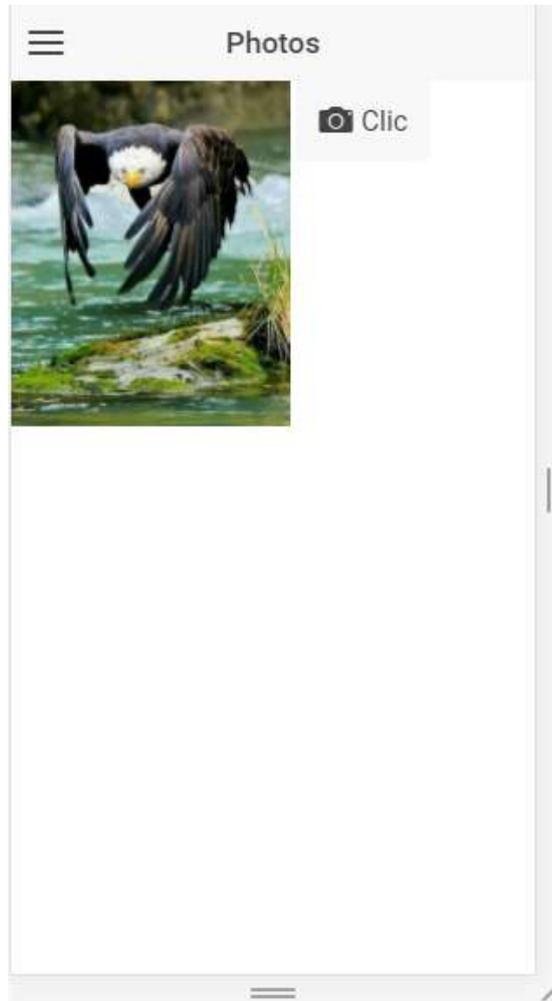
```
angular.module('myApp', ['ngCordova'])
```

## 2) Exemples d'utilisation des 'plugin' ngcordova

### a. Atelier utilisation de la camera

#### Objectifs 🎯

Créer une application permettant d'utiliser la camera du mobile, de capturer une photo et de l'affiche dans la page d'accueil.



#### Étapes 🏗️

- 1 – Création un nouveau projet de type **sidemenu**
- 2 – Ajout du script **ngCordova**.
- 3- Ajout du plugin **cordova-plugin-camera**.
- 4 – Modification des fichiers **Index.html**, **app.ps** et **controllers.jsp** pour pouvoir utiliser le camera.

#### Manipulation

Etape 1 création d'un nouveau projet

```
ionicstartProjetAPI sidemenu
```

- a. Accéder au dossier nouveau projet

```
cd ProjetAPI
```

- b. Ajouter le script ngCordova.

```
bower install ngCordova --save
```

- c. Ajouter le plugin camera dans le projet :

```
ionic plugin add cordova-plugin-camera
```

- d. Modification des fichiers

#### index.html

Pour ajouter l'import de la librairie **ngCordova**, ajouter le code ci-dessous avant le code d'importation de **cordova**

```
<script src="lib/ngCordova/dist/ng-cordova.min.js"></script>
```

#### app.js

Ajouter **ngCordova** dans la ligne qui commence par **angular.module** de la façon suivante

```
.angular.module('camera', ['ionic', 'camera.controllers', 'ngCordova'])
```

#### controllers.js

Ajouter à ce fichier le script contrôleur de la camera suivant.

```
controller('PhotosCtrl', function($scope, $cordovaCamera) {
  $scope.takePicture = function() {
    var options = {
      quality: 75, // Qualité de l'image sauvee, valeur entre 0 et 100
      destinationType: Camera.DestinationType.DATA_URL,
      sourceType: Camera.PictureSourceType.CAMERA,
      allowEdit: true,
      encodingType: Camera.EncodingType.JPEG, // Format d'encodage : JPEG ou
      PNG
      targetWidth: 300, // Largeur de l'image en pixel
      targetHeight: 300, // Hauteur de l'image en pixel
      saveToPhotoAlbum: false // Enregistrer l'image dans l'album photo
      du device
    };
    $cordovaCamera.getPicture(options).then(function(imageData) {
```

```

        $scope.imgURI = "data:image/jpeg;base64," + imageData;
    }, function(err) {
        console.log(err);
    });
};
});

```

### **photos.html** :

Créer le fichier **photos.html** dans le dossier **templates**. Ajouter le code suivant :

```

<ion-view view-title="Photos">
<ion-content>


<button class="button icon-left ion-camera" ng-
click="takePicture()">Clic</button>
</ion-content>
</ion-view>

```

### **menu.html** (dossier templates)

Ajoutez le code suivant à l'intérieur de balise<ion-list></ion-list>:

```

<ion-item menu-close href="#/app/photos">
    Photos
</ion-item>

```

### **fichier app.js**

Ajouterle code suivant à l'intérieur de la fonction **config(function(\$stateProvider, \$urlRouterProvider)**

```

.state('app.photos', {
    url: '/photos',
    views: {
        'menuContent': {
            templateUrl: 'templates/photos.html',
            controller: 'PhotosCtrl'
        }
    }
})

```

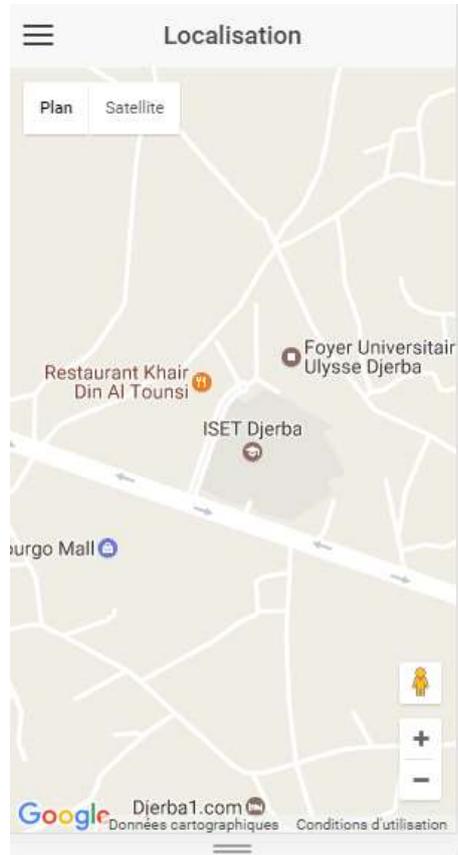
En fin lancer le projet

```
ionic serve
```

### a. Utilisation de plugin géolocalisation

#### Objectifs 🎯

Créer une application permettant d'afficher votre localisation en utilisant les APIs coogle Maps.



#### Étapes 🛠️

- 1 – Activer l’API coogle Maps
- 2 – Obtenir une clé de cette API
- 3- Créer l’application.

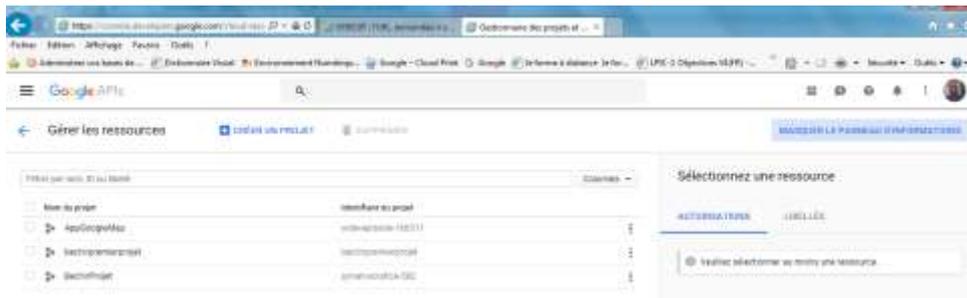
#### Manipulation :

##### **Etape 1 : activer l’API coogle Maps**

Pour créer la géolocalisation, on a besoin d'utiliser des API coogle Maps. Pour ce fait, il faut accéder au site [Google Developer Console](https://console.cloud.google.com/) pour activer cette API et obtenir une clé.

Les étapes à suivre sont :

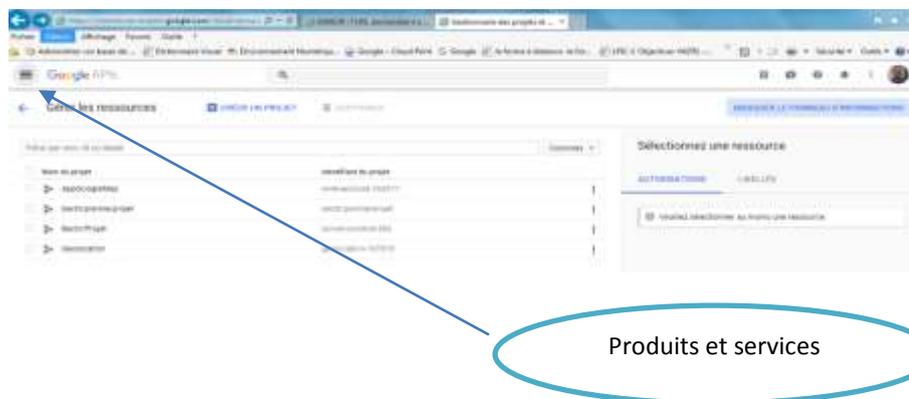
- a. Se rendre sur [Google Developer Console](https://console.cloud.google.com/), et se connecter avec le compte Google. (voir figure ci-dessous)



b. Créer un nouveau projet

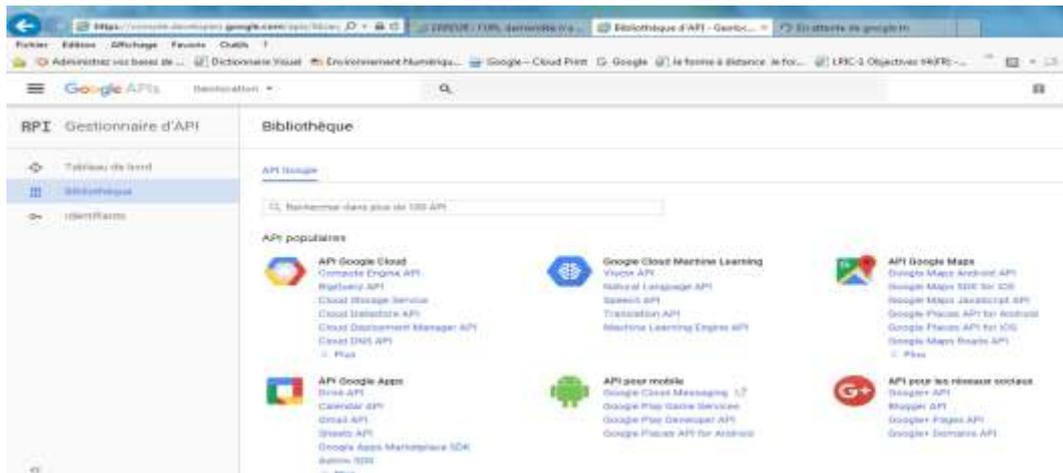


c. Cliquez sur « Produits et services »



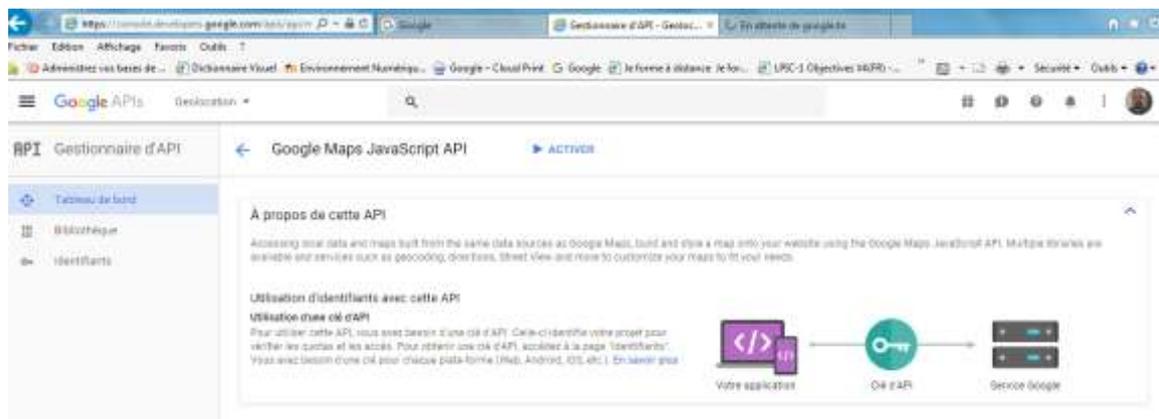
d. A partir du gestionnaire APIsélectionner le menuBibliothèque





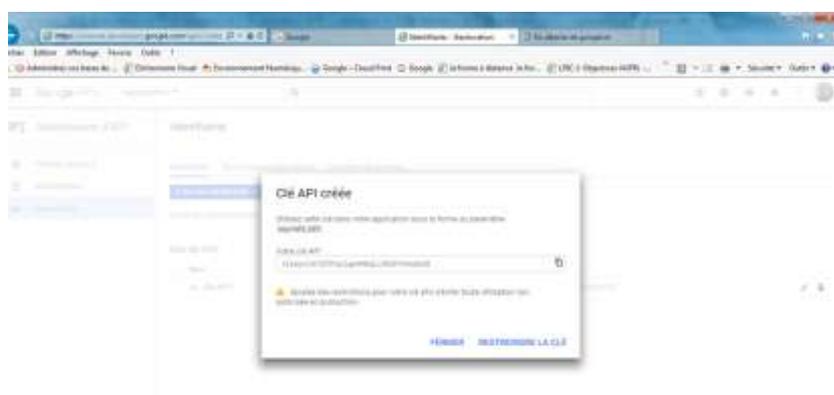
e. A partir de la Bibliothèque, activerles APIs

- Google Maps Javascript API
- [Google Maps Android API](#)



## Etape 2 : Obtenir une clé de l'API

a. A partir du menu identifiants, Créezune clé de l'API



b. Enregistrercette clé dans un fichier texte pour pouvoir l'utiliser ultérieurement.

### Etape 3- création de l'application.

c. Dans cette étape, ajouter au projet ProjetAPI une fonction permettant d'afficher notre carte.

1- Ajout du plugin **geolocalisation** :

```
ionic plugin add cordova-plugin-geolocation
```

2- Modification des fichiers

#### index.html

Ajouter dans l'entête du fichier

```
<meta http-equiv="Content-Security-Policy" content="default-src *;
script-src 'self' 'unsafe-inline' 'unsafe-eval' *; style-src 'self'
'unsafe-inline' *">
```

Ajouter le script suivant

```
<script src="https://maps-api-
ssl.google.com/maps/api/js?libraries=places"></script>
```

#### Localisation.html :

Créer le fichier **Localisation.html** dans le dossier **templates** le nommer **localisation.html** et ajoutez le code suivant :

```
<ion-view view-title="Localisation">
<ion-content>
<ion-content >
<div id="map" data-tap-disabled="true"></div>
</ion-content>
</ion-view>
```

#### menu.html (dossier templates)

Ajoutez le code suivant à l'intérieur de <ion-list></ion-list>:

```
<ion-item menu-close href="#/app/localisation">
    Localisation
</ion-item>
```

#### app.js

```
Ajoutez le code suivant à l'intérieur de la fonction config(function($stateProvider,
$urlRouterProvider)
state('app.localisation', {
    url: '/localisation',
    views: {
'menuContent': {
        templateUrl: 'templates/localisation.html',
controller: 'LocalCtrl'
    }
}
```

#### controllers.js

Ajouter à ce fichier le script contrôleur de l'application suivant

```

.controller('LocalCtrl', function($scope, $cordovaGeolocation,
$ionicLoading, $ionicPlatform) {
    $ionicPlatform.ready(function() {
        $ionicLoading.show({
            template: '<ion-spinner icon="bubbles"></ion-
spinner><br/>Acquiring location!'
        });

        var posOptions = {
            enableHighAccuracy: true,
            timeout: 20000,
            maximumAge: 10000
        };

        $cordovaGeolocation.getCurrentPosition(posOptions).then(function
(position) {
            var lat = position.coords.latitude;
            var longi = position.coords.longitude;
            var myLatlng = new google.maps.LatLng(lat, longi);
            var mapOptions = {
                center: myLatlng,
                zoom: 16,
                mapTypeId: google.maps.MapTypeId.ROADMAP
            };
            var map = new
google.maps.Map(document.getElementById("map"), mapOptions);
            $scope.map = map;
            $ionicLoading.hide();
        }, function(err) {
            $ionicLoading.hide();
            console.log(err);
        });
    });
});

```

### **Fichier style.css (dossier css)**

Ajoutez le code suivant à l'intérieur de `<ion-list></ion-list>` :

```

#map {
    height: 100%;
    width: 100%;
}
.scroll {
    height: 100%;
}

```

En fin lancer le projet

**ionic serve**

## Atelier 4 : Accès à une base de données distante MySQL

### Objectifs

- Accéder à une ou plusieurs tables d'une base de données MySQL d'un site web
- Réaliser l'interface mobile d'extraction des données

### Étapes

#### 1) Mise en place de l'environnement de travail

Création des contrôleurs

Création des vues

Création des routes

Développement PHP

#### 2) Création de la base de données

#### 3) Mise en place de l'environnement de travail

### Manipulation

L'atelier suppose soit :

- La possibilité de se connecter à une base de données MySQL distante telle que celle relative au site de l'institution.
- La mise en place d'un environnement local à base de XAMP ou équivalent.

#### 1) Création de la base de données

- Créer et peupler la table "jwm\_actualites"

Champ	Type et taille	Observations
id	Int(10)	Auto-incrément, clé primaire
intitule	Varchar(200)	---
image	Varchar(100)	Nom+Extension du fichier image
contenu	Text	---
date	Date	---

### Développement PHP

#### a. Connexion à la base de données

Rappel : Pour se connecter à une base de données MySQL, les informations suivantes sont nécessaires :

- L'adresse du serveur
- Le nom de la base de données
- Le nom d'utilisateur
- Le mot de passe

Généralement, il faut créer un fichier centralisé PHP contenant ces informations et se connectant à la base.

```
<?php
try{
    $con = "mysql:host=mysql.rnu.tn;dbname=institut";
    $pdo = new PDO($con, 'jwmu', 'motdepasse');
}catch(PDOException $e){
    $msg = 'Erreur PDO :'. $e->getMessage();
    die($msg);
}
?>
```

## b. Extraction des données

L'accès aux données stockées dans une base MySQL au niveau de l'application mobile se fait à travers le format JSON.

Le code ci-dessous permet l'extraction des enregistrements de la table "jwm\_actualites" en format JSON.

```
<?php
require("connexion.php");
$exec = $pdo->prepare("select * from jwm_actualites");
$exec->execute();
$sact = $exec->fetchAll(PDO::FETCH_ASSOC);
header("Content-Type: application/json;charset=utf-8");
echo(json_encode($sact));
?>
```

Le code ci-dessous permet l'extraction d'un seul enregistrement de la table "jwm\_actualites" en format JSON.

```
<?php
require("connexion.php");
$exec = $pdo->prepare("select * from jwm_actualites where id =
".$_GET["id"]);
$exec->execute();
$sact = $exec->fetchAll(PDO::FETCH_ASSOC);
header("Content-Type: application/json;charset=utf-8");
echo(json_encode($sact));
?>
```

## Création des routes

Pour pouvoir afficher l'interface de la liste des actualités ainsi que celle des détails d'une actualité donnée, il faut définir deux routes.

```
.state('actualites', {
    url: '/actualites',
    templateUrl: 'templates/actualites.html',
    controller: 'ActualitesCtrl' })

.state('actualite', {
    url: '/actualite/:id',
    templateUrl: 'templates/actualite.html',
    controller: 'ActualiteCtrl' })
```

## Création de vue : Affichage de la liste des actualités

```
<ion-view title="Actualités">
  <ion-content>
    <h2>Nos Actualités</h2>
    <ion-list>
      <ion-item ng-repeat="a in actualites" ng-
click="chargerActualite(a.id)">
        <h3> {{a.intitule}} </h3>
        
      </ion-item>
    </ion-list>
  </ion-content>
</ion-view>
```

## Création des contrôleurs

Pour acquérir les données déjà fournies par les fichiers PHP (en format JSON), il faut utiliser la méthode "get()".

Le code ci-dessous permet d'obtenir la liste des actualités.

```
.controller('ActualitesCtrl', function($http, $scope, $state,
$stateParams) {
  $scope.actualites = [];
  $scope.url = url;
  $scope.loadActualites= function(){
    $http.get(url+"/mbl-actualites.php")
      .success(function(data) {
        $scope.actualites = data;
      })
      .error(function(err) {
        console.log(err);
      })
  };
  $scope.loadActualites();
  $scope.chargerActualite=function(id) {
    $state.go("actualite",{id:id})
  }
})
```