Développement Mobile IONIC

Séquence 2

Plan

- Création de Pages
- Navigations entre les pages
- Les layouts
- Templates

Création des Pages sous Ionic

- Pour créer des pages sous ionic, même si on peut le faire manuellement, on peut utiliser la commande suivante pour générer des pages.
  - ***ionic generate page etudiants***
- Comme toute page web, On a:
  - l'entête spécifier par ***<ion-header> </ion-header>***
  - Le corps de l'application: ***<ion-content></ion-content>***
  - Le footer : ***<ion-footer></ion-footer>***
  - Ensuite il y'a d'autres composants comme la barre de navigation ***(<ion-navbar></ion-navbar>***), le titre ***(<ion-title></ion-title>),…..***

Routing des Pages Sous Ionic

- Pour ouvrir les pages créées, il faut faire du routage. Pour rappel notre technologie de base est angular.
- Tout d'babord il faut identifier le point d'entrée dans l'application.
- On niveau de l'architecture, on trouve dans " **src/app/home** " le "**HomePage** "



Routing des Pages Sous Ionic

Cette page renferme nos informations de routage de notre application.

Le mot clé ***loadChildren*** fournit le chemin des modules de nos pages

```
{ path: 'home', loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)},
```

Pour l'afficher maintenant, nous allons l'invoquer dans la page **index.html**

```
<body>
  <app-root></app-root>
</body>
```

Le HTML remplacera le ***app root*** contenu dans le fichier app
(**app/app.component.html**)

```
<ion-app>      You, 17 hours ago • Initial comm
  <ion-router-outlet></ion-router-outlet>
</ion-app>
```

Routing des Pages Sous Ionic

IL est aussi possible de passer des paramètres à une page.

Ainsi, on peut utiliser trois techniques de transmettre des données pour une :

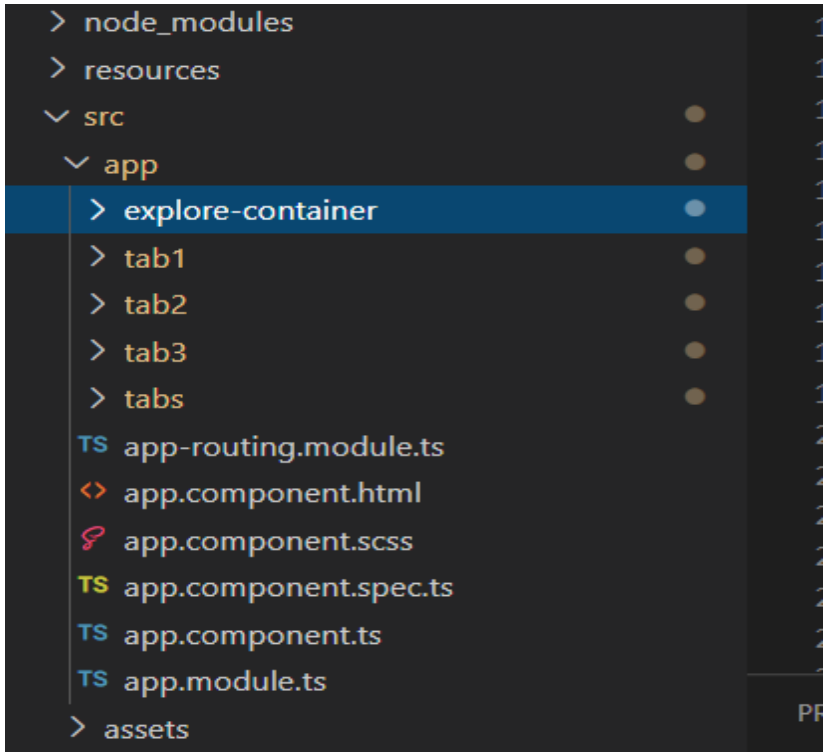
1- Utiliser un service qui stocke les informations, puis charger les informations via ce service sur la nouvelle page.

2- Passez un ID avec l'URL à la page suivante, puis faites une nouvelle demande à votre API (ou peut-être à votre service si vous mettez en cache les réponses) afin d'obtenir les données pour l'ID spécifique d'un objet.

3- Utilisez les extras de navigation introduits avec Angular 7.2

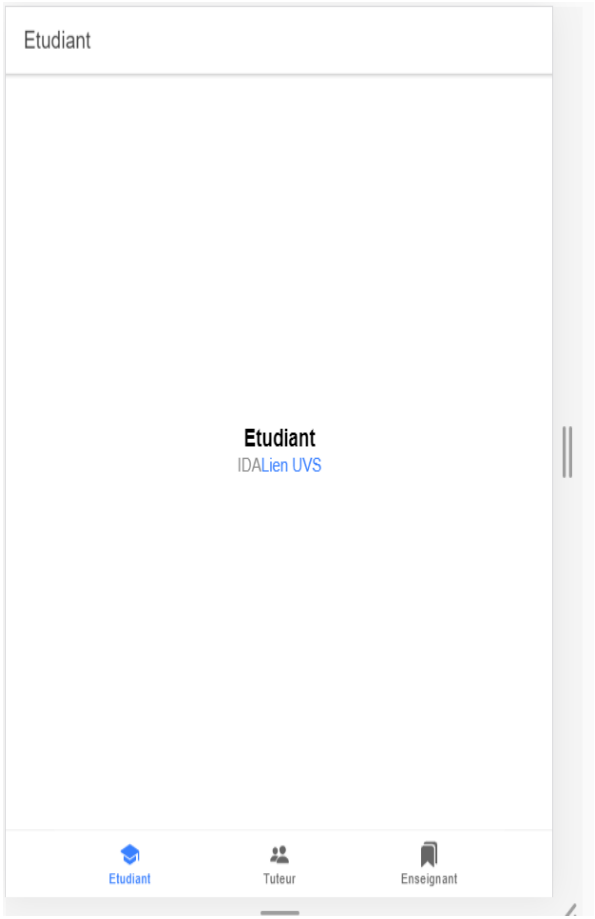
Tab Bar Navigation

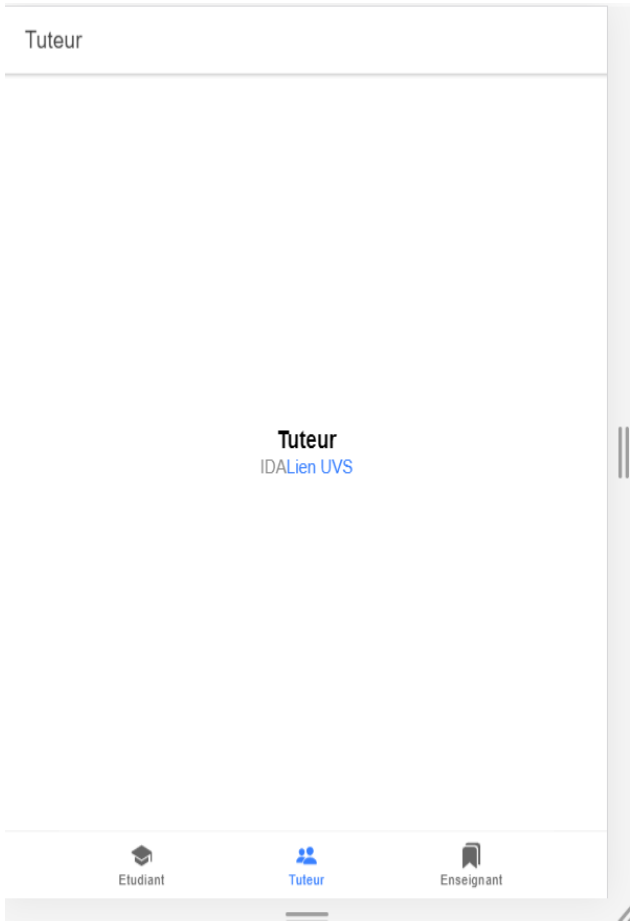
- Créer un projet avec avec des tabs menu en tapant la commande:
  - **ionic start myApp tabs –cordova**

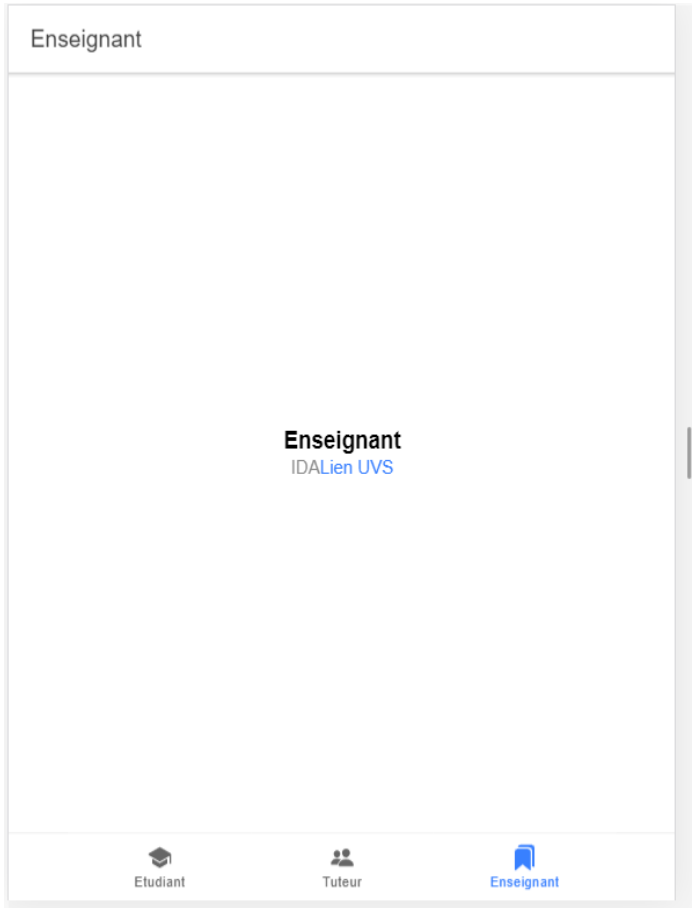



```
<ion-tabs>
  <ion-tab-bar slot="bottom">
    <ion-tab-button tab="tab1">
      <ion-icon name="school"></ion-icon>
      <ion-label> Etudiant</ion-label>
    </ion-tab-button>
    <ion-tab-button tab="tab2">
      <ion-icon name="people"></ion-icon>
      <ion-label> Tuteur</ion-label>
    </ion-tab-button>
    <ion-tab-button tab="tab3">
      <ion-icon name="bookmarks"></ion-icon>
      <ion-label>Enseignant</ion-label>
    </ion-tab-button>
  </ion-tab-bar>      You, 29 minutes ago • Initial co
</ion-tabs>
```

Exemple


Etudiant
Etudiant
IDALien UVS
Etudiant
Tuteur
Enseignant


Tuteur
Tuteur
IDALien UVS
Etudiant
Tuteur
Enseignant


Enseignant
Enseignant
IDALien UVS
Etudiant
Tuteur
Enseignant

Propriétés des Tabs

Input Properties

Attr	Type	Details
name	`string`	A unique name for the tabs
selectedIndex	`number`	The default selected tab index when first loaded. If a selected index isn't provided then it will use `0`, the first tab.
tabsHighlight	`boolean`	If true, show the tab highlight bar under the selected tab.
tabsLayout	`string`	Set the tabbar layout: `icon-top`, `icon-start`, `icon-end`, `icon-bottom`, `icon-hide`, `title-hide`.
tabsPlacement	`string`	Set position of the tabbar: `top`, `bottom`.

Manipulation Instance Tabs

`previousTab(trimHistory)`

Get the previously selected Tab which is currently not disabled or hidden.

Param	Type	Details
trimHistory	`boolean`	If the selection history should be trimmed up to the previous tab selection or not.

→ Returns: `Tab`

`select(tabOrIndex)`

Param	Type	Details
tabOrIndex	`number` | `Tab`	Index, or the Tab instance, of the tab to select.

`viewCtrl`

`getByIndex(index)`

Param	Type	Details
index	`number`	Index of the tab you want to get

→ Returns: `Tab`

Returns the tab who's index matches the one passed

`getSecondaryIdentifier()`

`getSelected()`

▸ Returns: `Tab`

eturns the currently selected tab

`getType()`

`goToRoot()`

En-tête

- La mise en page la plus simple disponible se compose d’une en-tête et d'un contenu. La plupart des pages d'une application ont généralement les deux, mais une en-tête n'est pas nécessaire pour utiliser le contenu.

```
<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>
```

Footer

Comme vous pouvez le voir, une barre d'outils dans un en-tête apparaît au-dessus du contenu. Parfois, une application doit avoir une barre d'outils sous le contenu, c'est-à-dire lorsqu'un pied de page est utilisé.

```
<ion-app>
  <ion-content class="ion-padding">
    <h1>Main Content</h1>
  </ion-content>

  <ion-footer>
    <ion-toolbar>
      <ion-title>Footer</ion-title>
    </ion-toolbar>
  </ion-footer>
</ion-app>
```

Menu layout

```
<ion-app>
  <ion-split-pane contentId="main-content">
    <ion-menu contentId="main-content" type="overlay">
      <ion-content>
        <ion-list id="inbox-list">
          <ion-list-header>Inbox</ion-list-header>
          <ion-note>hi@ionicframework.com</ion-note>

          <ion-menu-toggle auto-hide="false" *ngFor="let p of appPages; let i = index">      You, 7 minutes ago • Initial commit
            <ion-item (click)="selectedIndex = i" routerDirection="root" [routerLink]="[p.url]" lines="none" detail="false" [class
              <ion-icon slot="start" [ios]="p.icon + '-outline'" [md]="p.icon + '-sharp'"></ion-icon>
              <ion-label>{{ p.title }}</ion-label>
            </ion-item>
          </ion-menu-toggle>
        </ion-list>

        <ion-list id="labels-list">
          <ion-list-header>Labels</ion-list-header>

          <ion-item *ngFor="let label of labels" lines="none">
            <ion-icon slot="start" ios="bookmark-outline" md="bookmark-sharp"></ion-icon>
            <ion-label>{{ label }}</ion-label>
          </ion-item>
        </ion-list>
      </ion-content>
    </ion-menu>
    <ion-router-outlet id="main-content"></ion-router-outlet>
  </ion-split-pane>
</ion-app>
```

Menu layout

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

import { Platform } from '@ionic/angular';
import { SplashScreen } from '@ionic-native/splash-screen/ngx';
import { StatusBar } from '@ionic-native/status-bar/ngx';

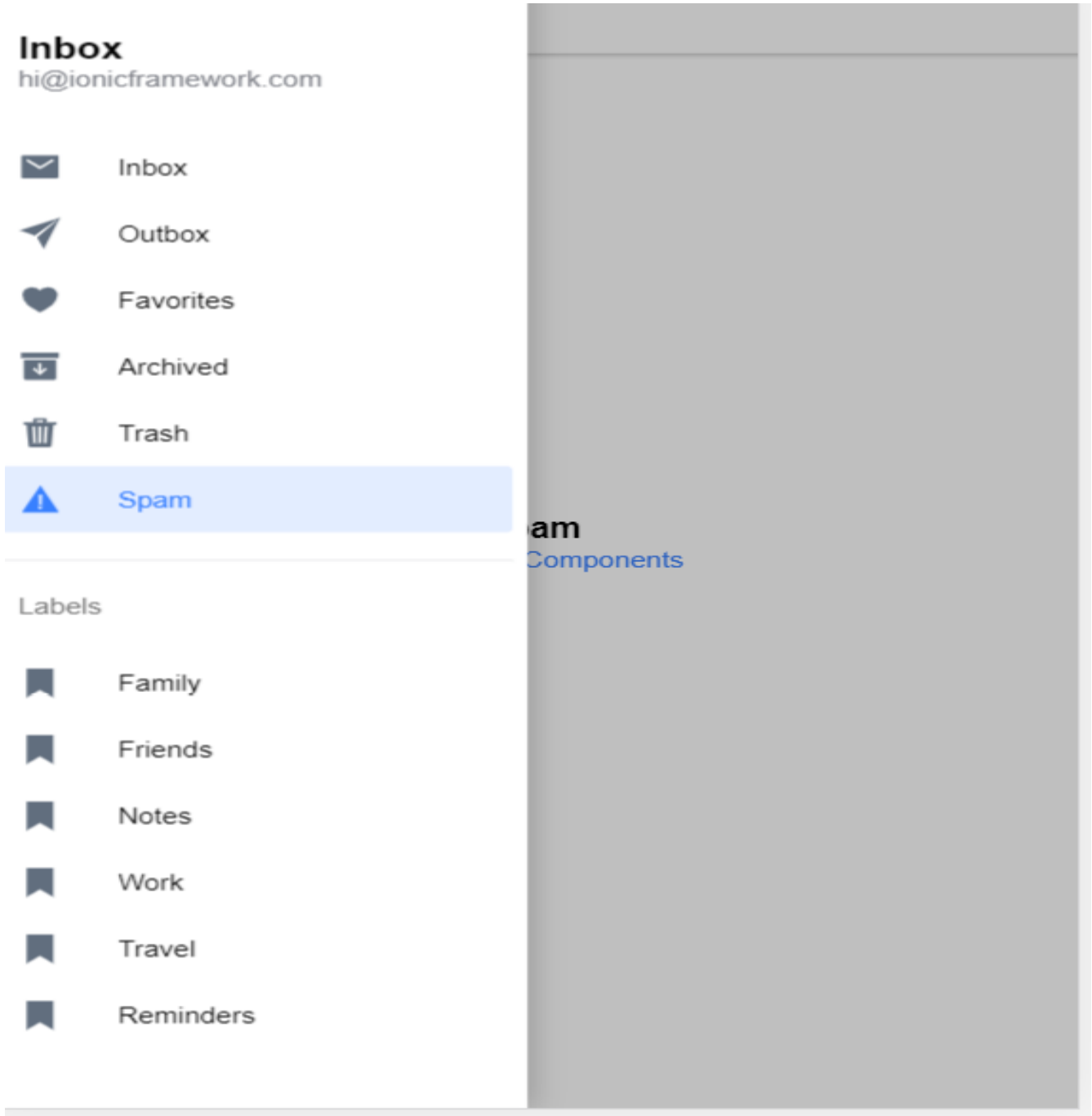
You, 9 minutes ago | 1 author (You)
@Component({
  selector: 'app-root',
  templateUrl: 'app.component.html',
  styleUrls: ['app.component.scss']
})
export class AppComponent implements OnInit {
  public selectedIndex = 0;
  public appPages = [
    {
      title: 'Inbox',
      url: '/folder/Inbox',
      icon: 'mail'
    },
    {
      title: 'Outbox',
      url: '/folder/Outbox',
      icon: 'paper-plane'
    },
    {       You, 10 minutes ago • Initial commit
      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(
    private platform: Platform,
    private splashScreen: SplashScreen,
    private statusBar: StatusBar
  ) {
    this.initializeApp();
  }

  initializeApp() {
    this.platform.ready().then(() => {
      this.statusBar.styleDefault();
      this.splashScreen.hide();
    });
  }
```

```
ngOnInit() {
  const path = window.location.pathname.split(
    'folder/')[1];
  if (path !== undefined) {
    this.selectedIndex = this.appPages.findIndex
    (page => page.title.toLowerCase() === path.toLowerCase());
  }
}
}
```


Inbox
hi@ionicframework.com
Inbox
Outbox
Favorites
Archived
Trash
Spam
Labels
Family
Friends
Notes
Work
Travel
Reminders
am
Components

Responsive Grid

- Attributs et style:
  - Ionic propose un ensemble de ressources ou d'attributs qui peuvent être utilisées sur n'importe quel élément pour faire des modifications sur le texte(marging, padding,…)

Class	Style Rule	Description
`.ion-text-left`	`text-align: left`	The inline contents are aligned to the left edge of the line box.
`.ion-text-right`	`text-align: right`	The inline contents are aligned to the right edge of the line box.

Pour plus de styles: https://ionicframework.com/docs/layout/css-utilities

Exemple

Ionic propose également un système de grille css pour permettre une meilleur gestion de blocs de contenus. Il est assez similaire dans sa syntaxe à celui que propose Bootstrap.

```
<ion-header>
  <ion-toolbar color="primary">
    <ion-title>UVS IONIC DEV MOBILE</ion-title>
  </ion-toolbar>
</ion-header>

<h2>{{ name }}</h2>
<ion-grid>
  <ion-row>
    <ion-col col-lg-1>
      <img src="assets/imgs/logoUVS.JPG"/>
    </ion-col>
    <ion-col>
          IDA UVS
      </ion-col>
    <ion-col>
      Développeur Web et Mobile
    </ion-col>      You, 4 hours ago • Uncommitte
  </ion-row>
  <ion-row>
    <ion-col>
      Bienvenue à la formation de IONIC
    </ion-col>
  </ion-row>
</ion-grid>
```

Exemple


Etudiant
UVS IONIC DEV MOBILE
Etudiant
IDA UVS
Développeur Web et Mobile
UVS
UNIVERSITE VIRTUELLE DU SENEGAL
foo nekk foofu ta
Bienvenue à la formation de IONIC
Etudiant
Tuteur
Enseignant

Thèmes et Customisation

- Ionic Framework est conçu pour être une ardoise vierge qui peut facilement être personnalisée et modifiée pour s'adapter à une marque, tout en respectant les normes des différentes plates-formes.
- Theming Ionic apps est maintenant plus facile que jamais. Parce que le framework est construit avec CSS, il est livré avec des styles par défaut précuits qui sont extrêmement faciles à changer et à modifier.

Colors

- Ionic possède neuf couleurs par défaut. Ces couleurs peuvent être utilisées à l'intérieur des components. Chaque couleur renferme au moins les deux propriétés suivantes: ***shade*** et ***tint.***



Plateformes Standard

- Les composants ionic adaptent leur apparence et leur comportement en fonction de la plateforme sur laquelle l'application s'exécute. Nous appelons cela le **style adaptatif**. Cela permet aux développeurs de créer des applications qui utilisent la même base de code pour plusieurs plates-formes, tout en restant «natives» pour ces plates-formes particulières.

- Ionic a deux modes qui sont utilisés pour personnaliser l'apparence des composants basés sur la plate-forme: ios et md. Chaque plate-forme a un mode par défaut, mais celui-ci peut facilement être configuré. Pour plus d'informations sur la personnalisation d'une application basée sur la plateforme, voir Styles de plateforme.

Thèmes

- Ionic fournit plusieurs variables globales qui sont utilisées dans les composants pour changer le thème par défaut d'une application entière.
- Les couleurs d'application sont utiles pour changer l'apparence de la plupart des composants ioniques, et les couleurs étagées sont utilisées comme variations dans certains des composants ioniques.
- Les couleurs d'application sont utilisées à plusieurs endroits dans Ionic. Ils sont utiles pour créer facilement des thèmes sombres ou des thèmes qui correspondent à une marque.

Utilisation de SASS

- Ionic est construit sur Sass (Syntactically Awesome Stylesheets), un langage de génération de feuilles de style, robuste et facile à prendre en main.

- La définition ou la redéfinition de style css dynamique se fait depuis le fichier **src/theme/variables.scss.** Par défaut vous trouverez des styles qui sont déjà définis.

FIN DE LA SEQUENCE
Questions 2

References

- https://ionicframework.com/docs/
- https://openclassrooms.com/fr/courses/5098931-developpez-une-application-mobile-multiplateforme-avec-ionic-3
- https://ionic.mobiletuto.com/v3/chap10/chap10-5.html