

# Développement application mobile IONIC/ANGULAR

## Ionic : Développement d'applications hybrides :

### 1. Définition :

IONIC est un **framework open-source** permettant de développer des **applications mobiles hybrides** (compatibles Android, iOS et web) en utilisant des technologies web standards :

- ✧ **HTML** : Structure de l'application
- ✧ **CSS** : Style et mise en page
- ✧ **TypeScript** : Logique et fonctionnalités

### 2. Déploiement sur les plateformes mobiles :

- ✧ **Android** : ○ Nécessite l'ajout de **plugins JavaScript** spécifiques à Android pour accéder aux fonctionnalités natives (ex : appareil photo, géolocalisation).
- ✧ **iOS** : ○ De même, il faut **ajouter des plugins JS** compatibles avec iOS.
  - Requiert un environnement **macOS** pour la compilation et le déploiement sur l'App Store.

### 3. Structure d'une application Ionic :

Une application mobile Ionic est composée de :

- **Une application web** (HTML, CSS, TypeScript).
- **Des plugins** pour accéder aux **APIs natives** (appareil photo, capteurs, stockage, etc.).

### 4. Avantages d'Ionic :

- ✓ **Coût réduit** : Un seul code source pour plusieurs plateformes (Android, iOS, Web).
- ✓ **Développement rapide** : Reposant sur des technologies web bien maîtrisées.
- ✓ **Communauté active** : De nombreux plugins et documentation détaillée.

## 5. Inconvénients :

✗ **Consommation de mémoire** : Les applications Ionic utilisent une **WebView** (navigateur intégré), ce qui peut entraîner une consommation plus élevée en ressources (mémoire RAM et CPU).

✗ **Performances** : Légèrement moins rapides qu'une application native, surtout pour des applications complexes.

## 6. Les version d'Angular :

□ **AngularJS** (2009)

- **Angular 2** (2014) - Basé sur le modèle MVVM (Model-View-ViewModel) et conçu pour les applications monopages (SPA - Single Page Application).
- **Angular 3** n'existe pas officiellement, la version suivante après **Angular 2** est **Angular 4** (2017).

## 7. Les type d'application mobile.

Les applications mobiles peuvent être classées en trois grandes catégories :

□ **Application native** : Ce sont des applications développées spécifiquement pour un système d'exploitation (Android ou iOS). Elles offrent des performances optimales et un accès complet aux fonctionnalités du périphérique.

➤ Exemples :

- Applications Android développées en Kotlin ou Java
- Applications iOS développées en Swift ou Objective-C

□ **Application Cross platform native** : Ces applications sont développées avec un seul code source et peuvent fonctionner sur plusieurs plateformes. Elles offrent de meilleures performances que les applications hybrides.

➤ Technologies utilisées :

- Flutter (Dart)
- React Native (JavaScript/TypeScript)

□ **Application Web-Mobile Hybride** : Ces applications sont basées sur des technologies web (HTML, CSS, JavaScript) et s'exécutent dans une WebView au sein d'une application native. Ionic est un exemple populaire de ce type d'application.

## 8. Utilité de NodeJs :

Node.js joue un rôle essentiel dans le développement avec Ionic et Angular. Il est utilisé principalement pour :

- Gérer les dépendances avec NPM (Node Package Manager)
- Compiler TypeScript en JavaScript
- Assurer la gestion des modules JavaScript pour le développement d'applications web modernes

## 10. Commandes essentielles :

**Création du dossier www :**

**IONIC BUILD**

**Synchronisation des fichiers : IONIC**

**CAPACITOR SYNC**

**Configuration du projet Android : IONIC**

**CAPACITOR ADD ANDROID**

**Configuration du projet iOS : IONIC**

**CAPACITOR ADD IOS**

**Lancement du projet Android dans Android Studio : IONIC**

**CAPACITOR BUILD ANDROID**

## 11. Standalone vs. NgModules:

Un projet Standalone en Angular, introduit à partir d'Angular 14, permet de créer des composants autonomes sans utiliser de module (NgModule). Grâce au décorateur standalone: true, les dépendances peuvent être importées directement, ce qui simplifie la structure du projet et améliore la performance grâce à une meilleure optimisation du tree-shaking. Ce mode est idéal pour les applications modernes, légères et modulaires.

En revanche, un projet basé sur NgModules repose sur une structure modulaire traditionnelle où les composants, services et directives sont regroupés au sein d'un module défini par le décorateur @NgModule. Cette approche permet une meilleure organisation et une gestion centralisée des dépendances, ce qui la rend adaptée aux grands projets nécessitant une architecture modulaire stricte et une compatibilité avec les versions antérieures d'Angular.

## **Angular AngularJS et Angular:**

Deux frameworks de développement web développés par Google pour faciliter la création d'applications web dynamiques.

## **Différences entre AngularJS et Angular:**

Angular est basé sur une architecture de composants, tandis qu'AngularJS utilise une architecture de contrôleur et de portée. Angular offre également des performances supérieures à celles d'AngularJS grâce à une détection de changement plus rapide et une compilation AOT. Angular est écrit en TypeScript, qui est un langage de programmation typé statiquement basé sur JavaScript. D'autre part, AngularJS est écrit en JavaScript simple.

## **SPA:**

SPA signifie "Single Page Application" (Application à page unique). C'est un type d'application web qui charge une seule page HTML et met à jour dynamiquement le contenu en utilisant JavaScript, offrant une expérience utilisateur fluide et rapide, sans rechargement de page complet. - L'ensemble des éléments de l'application est chargé (contenu, images, CSS et JavaScript) dans un unique fichier HTML. - Les ressources nécessaires sont récupérées et affichées dynamiquement en fonction des actions de l'utilisateur.

## **Outils nécessaires pour configurer l'environnement Angular:**

Node.js, Angular CLI, un éditeur de code, un navigateur web, TypeScript et les dépendances du projet.

## **Deux utilités de l'écosystème Nodejs pour un projet Angular:**

Gestion des dépendances et création de serveurs d'application.

## **La commande CLI pour créer une application Angular: ng**

new FilmsDB

## **La commande CLI pour tester l'application: cd**

FilmsDB après ng test

## Les étapes à suivre pour intégrer et utiliser le Framework Bootstrap 4:

-Installer Bootstrap 4 NPM `npm install bootstrap`

### Pour intégrer et utiliser le Framework Bootstrap 4 dans le projet, voici les étapes à suivre (autre réponse):

-Installer Bootstrap 4 à l'aide de la commande NPM : `'npm install bootstrap'` -Ajouter les fichiers CSS et JavaScript de Bootstrap dans le fichier `angular.json` -Importer les classes Bootstrap dans les fichiers `.scss` ou `.css` du projet.

### Deux manières de déclarations des services en Angular:

L'injection de dépendances, fournisseur de module

### Comment on développe la navigation et le routage en Angular:

-Configuration : Importez le module `RouterModule` dans votre module principal et utilisez la méthode `forRoot()` pour configurer les routes de votre application.

-Liens de navigation : Utilisez la directive `routerLink` dans vos templates HTML pour créer des liens de navigation vers différentes routes. -Point de sortie : Utilisez la directive `routeroutlet` dans votre template principal pour indiquer où afficher le contenu des routes.

-Gestion des paramètres de route : Utilisez l'objet `ActivatedRoute` pour accéder aux paramètres de route dans le composant associé à la route.

-Routes enfants : Définissez des routes enfants pour créer une hiérarchie de navigation en utilisant des tableaux de routes.

### Service qui contient une seule méthode qui reçoit le nom de la ville comme paramètre et renvoi un tableau des données météorologiques : `weather.service.ts`

Dans le fichier `weather.service.ts`, vous pouvez créer le service `WeatherService` et implémenter la méthode `getWeatherData(city: string)` qui reçoit le nom de la ville comme paramètre et renvoie un tableau des données météorologiques.

### QSI : Décrire et comparer les deux manières pour développer les formulaires en Angular.

Il existe deux méthodes :

1. **Template-Driven Forms** : Gérés dans le HTML avec `ngModel`, simple mais moins flexible.
2. **Reactive Forms** : Gérés en TypeScript avec `FormGroup`, plus puissant et adapté aux formulaires complexes.

### QS2 : Citer les deux manières de déclarations des services en Angular.

- Dans `@NgModule (app.module.ts)` → Ajouter le service dans `providers`.
- Dans `@Injectable({ providedIn: 'root' })` → Angular le rend disponible partout.

### QS3 : Expliquer Comment on développe la navigation et le routage en Angular.

- Définir les routes dans `app-routing.module.ts` avec `RouterModule.forRoot()`.
- Utiliser `<router-outlet>` pour afficher les composants.
- Naviguer avec `[routerLink]` ou `this.router.navigate()`.

**QS4 : Donner et expliquer trois méthode pour faire le Data Binding en Angular**

- Interpolation (`{{ }}`) → Affiche une valeur dans le HTML.
- Property Binding (`[property]="value"`) → Lien entre le TypeScript et le HTML.
- Two-Way Binding (`[(ngModel)]="value"`) → Synchronisation entre le modèle et la vue.

**QS5 : Citer deux différences entre les frameworks AngularJS**

- Architecture :
  - AngularJS (version 1) utilise MVC (Model-View-Controller).
  - Angular (à partir de la version 2) utilise une architecture basée sur les composants.
- Langage :
  - AngularJS utilise JavaScript.
  - Angular utilise TypeScript, plus structuré et maintenable.

**QS6 : Expliquer la notion SPA.**

Une SPA (Single Page Application) est une application web qui :

- Charge une seule page HTML et met à jour dynamiquement son contenu sans recharger toute la page.
- Utilise JavaScript et AJAX pour interagir avec le serveur en arrière-plan.
- Améliore l'expérience utilisateur en rendant la navigation fluide et rapide.

Angular est conçu pour créer des SPAs grâce à son système de routage et data binding.

**QS7 : Citer les outils nécessaires pour configurer l'environnement de développement des projets Angular.**

- Node.js (avec `npm`) → Pour gérer les dépendances et exécuter Angular CLI.
- Angular CLI → Outil en ligne de commande pour créer et gérer des projets Angular.
- Un éditeur de code (comme Visual Studio Code).
- Un navigateur web (comme Chrome avec Angular DevTools pour le debug).

**QS8 : Donner deux utilités de l'écosystème Nodejs pour un projet Angular.**

- Gestion des dépendances → `npm` (Node Package Manager) permet d'installer les bibliothèques Angular.
- Serveur de développement → Angular CLI utilise un serveur Node.js pour exécuter l'application en local (`ng serve`).