

🚀 Questions Générales sur Ionic

1/Quand et par qui Ionic a-t-il été créé ?

Ionic a été créé en 2013 par Max Lynch, Ben Sperry et Adam Bradley chez Drifty Co.

2/Quelles sont les principales versions d'Ionic ?

✓ Ionic 1 (2013) → Basé sur AngularJS

✓ Ionic 2 (2016) → Réécriture complète avec Angular 2+

✓ Ionic 3 (2017) → Amélioration des performances

✓ Ionic 4 (2019) → Basé sur Web Components, support pour React & Vue

✓ Ionic 5 (2020) → Améliorations UI et performances

✓ Ionic 6 (2021) → Support natif amélioré

3/Quelles technologies Ionic utilise-t-il ?

HTML, CSS, JavaScript, TypeScript et des frameworks comme Angular, React et Vue.

4/Quelle est la différence entre Ionic et Flutter ?

✓ Ionic → Basé sur des technologies web (HTML, CSS, JS).

✓ Flutter → Utilise Dart et une approche full-native.

5/Quels sont les outils nécessaires pour développer avec Ionic ?

✓ Node.js & npm ✓ Ionic CLI ✓ Capacitor ou Cordova ✓ Un éditeur de code (VS Code recommandé)

6/Pourquoi Capacitor a-t-il remplacé Cordova ?

Capacitor est plus moderne, mieux intégré avec les APIs natives et supporte TypeScript.

7/Quels sont les avantages des Web Components dans Ionic ?

✓ Compatible avec plusieurs frameworks (Angular, React, Vue).

✓ Réutilisable et performant.

8/Comment Ionic gère-t-il l'accès aux fonctionnalités natives ?

Via Capacitor ou Cordova, en utilisant des plugins (`Camera`, `Geolocation`, etc.).

9/Quelle est la différence entre une PWA et une application Ionic ?

✓ PWA (Progressive Web App) → Application web optimisée pour mobile.

✓ Ionic → Framework permettant de créer des PWAs ou des applications mobiles hybrides.

10/Quels sont les défis liés à Ionic ?

✗ Performances limitées pour des apps très complexes.

✗ Dépendance aux WebViews et frameworks web.

🚀 Questions Générales sur Angular

1/Qui a créé Angular et quand ?

AngularJS a été créé en 2010 par Google.

2/Pourquoi AngularJS a-t-il été remplacé par Angular ?

AngularJS était basé sur MVC et avait des problèmes de performance et de scalabilité.

3/Quelles sont les principales versions d'Angular ?

✓ AngularJS (2010) → Basé sur JavaScript et MVC.

✓ Angular 2 (2016) → Complète réécriture avec TypeScript.

✓ Angular 4 à 9 (2017-2020) → Améliorations progressives.

✓ Angular 10+ (2020-...) → Meilleures performances et support long terme.

4/Quelle est la différence entre Angular et React ?

✓ Angular → Framework complet basé sur TypeScript.

✓ React → Bibliothèque JavaScript plus légère pour les interfaces UI.

5/Quels sont les principaux concepts d'Angular ?

✓ Composants ✓ Modules ✓ Data Binding

✓ Services & Injection de dépendances ✓ Routing

6/Pourquoi Angular utilise-t-il TypeScript ?

TypeScript apporte une meilleure gestion des erreurs, une structure plus rigoureuse et une meilleure maintenabilité.

7/Qu'est-ce qu'un SPA en Angular ?

Une Single Page Application (SPA) charge une seule page HTML et met à jour son contenu dynamiquement.

8/Comment Angular optimise-t-il les performances des applications ?

- ✓ Lazy Loading pour charger les modules à la demande.
- ✓ Change Detection optimisé.
- ✓ AOT (Ahead-of-Time Compilation) pour améliorer le rendu.

9/Quelle est l'importance de RxJS en Angular ?

RxJS permet la gestion des événements asynchrones avec les Observables.

10/Quels sont les outils nécessaires pour Angular ?

- ✓ Node.js & npm
- ✓ Angular CLI
- ✓ TypeScript
- ✓ Un éditeur comme VS Code

🚀 Questions & Réponses - Ionic

1/Qu'est-ce qu'Ionic et pourquoi l'utiliser ?

Ionic est un framework open-source permettant de créer des applications mobiles hybrides en utilisant HTML, CSS et TypeScript.

2/Quelle est la différence entre une application native et une application hybride ?

Une application native est développée spécifiquement pour un OS, tandis qu'une application hybride fonctionne sur plusieurs plateformes via une WebView.

3/Quels sont les avantages d'Ionic ?

- ✓ Un seul code source pour Android, iOS et Web.
- ✓ Développement rapide.
- ✓ Grande communauté et nombreux plugins.

4/Quels sont les inconvénients d'Ionic ?

- ✗ Performances inférieures aux applications natives.
- ✗ Consommation élevée de mémoire.

5/Quelle est la commande pour créer un projet Ionic ?

``ionic start nom_du_projet blank``

6/Comment exécuter une application Ionic en mode développement ?

``ionic serve``

7/Quelle est la commande pour ajouter une plateforme Android ?

``ionic capacitor add android``

8/Comment fonctionne la navigation dans Ionic ?

En utilisant ``RouterModule`` et les balises ``[routerLink]``.

9/Quelle est la différence entre Cordova et Capacitor ?

Capacitor est plus moderne et simplifie l'intégration des APIs natives.

10/Comment accéder aux fonctionnalités natives dans Ionic ?

En utilisant des plugins Capacitor ou Cordova.

11/Quels sont les composants UI de base fournis par Ionic ?

``<ion-button>`,`<ion-card>`,`<ion-input>`,`<ion-tabs>`...`

12/Quelle est la commande pour compiler un projet Ionic pour Android ?

``ionic capacitor build android``

13/Comment intégrer Firebase dans une application Ionic ?

Installer Firebase SDK et configurer ``firebaseConfig``.

14/Comment gérer l'état global dans une application Ionic ?

En utilisant des services Angular ou NgRx.

15/Comment fonctionne le mode hors ligne dans Ionic ?

En stockant les données localement avec SQLite ou Ionic Storage.

16/Quels sont les événements du cycle de vie d'une page Ionic ?

``ionViewWillEnter`,`ionViewDidEnter`,`ionViewWillLeave`,`ionViewDidLeave``.

17/Quelle est la commande pour générer un composant Ionic ?

``ionic generate component NomComposant``

18/Comment publier une application Ionic sur le Play Store ?

Générer un APK avec `ionic capacitor build android`, puis soumettre sur Google Play Console.

19/Quelle est la différence entre un service et un composant dans Ionic ?

Un service gère la logique métier, tandis qu'un composant affiche des éléments UI.

20/Quelles sont les commandes essentielles pour synchroniser un projet Ionic ?

✔ `ionic capacitor sync` ✔ `ionic capacitor update`

🚀 Questions & Réponses - Angular

1/Qu'est-ce qu'Angular et pourquoi l'utiliser ?

Un framework basé sur TypeScript permettant de développer des applications web dynamiques et modulaires.

2/Quelle est la différence entre Angular et AngularJS ?

AngularJS utilise MVC et JavaScript, tandis qu'Angular utilise une architecture de composants et TypeScript.

3/Comment créer un projet Angular ?

`ng new nom_du_projet`

4/Quels sont les fichiers principaux d'un projet Angular ?

✔ `src/app/` : fichiers de l'application.

✔ `angular.json` : configuration.

5/Quelle est la structure d'un composant Angular ?

Un fichier `.ts`, un fichier `.html` et un fichier `.css`.

6/Comment ajouter un service Angular ?

`ng generate service NomService`

7/Quelles sont les formes de Data Binding en Angular ?

✔ Interpolation `{{ }}` ✔ Property Binding `[property]`

✔ Event Binding `(event)` ✔ Two-Way Binding `[(ngModel)]`

8/Quelle est la commande pour démarrer un projet Angular ?

``ng serve``

9/Comment fonctionne le routing en Angular ?

En définissant les routes dans ``app-routing.module.ts`` avec ``RouterModule.forRoot([])``.

10/Quelle est la différence entre ``ngIf`` et ``ngSwitch`` ?

✓ ``ngIf`` affiche ou masque un élément selon une condition.

✓ ``ngSwitch`` affiche différents éléments selon une expression.

11/Qu'est-ce que le Lazy Loading en Angular ?

Une technique permettant de charger les modules uniquement quand ils sont nécessaires.

12/Comment gérer les formulaires en Angular ?

Avec ``Template-Driven Forms`` et ``Reactive Forms``.

13/Comment fonctionne RxJS dans Angular ?

RxJS est utilisé pour la gestion des flux asynchrones et des ``Observables``.

14/Quelles sont les étapes pour ajouter Bootstrap à Angular ?

✓ ``npm install bootstrap`` puis ajout dans ``angular.json``.

15/Quelle est la commande pour générer un module Angular ?

``ng generate module NomModule``

16/Comment faire des appels API en Angular ?

En utilisant ``HttpClientModule`` et ``HttpClient``.

17/Comment utiliser les directives Angular ?

✓ ``ngClass`` pour modifier les classes CSS dynamiquement.

✓ ``ngStyle`` pour appliquer du style conditionnellement.

18/Quelle est la différence entre ``@Input`` et ``@Output`` en Angular ?

✓ ``@Input()`` permet de passer des données d'un parent à un enfant.

✓ ``@Output()`` permet d'envoyer des événements d'un enfant au parent.

19/Comment fonctionne l'Injection de dépendances en Angular ?

Angular gère automatiquement l'injection des services dans les composants.

20/Quelle est la commande pour exécuter les tests unitaires en Angular ?

`ng test`