

Guide d'étapes clés : Développer le front-end en utilisant Angular

Comment utiliser ce document ?

Ce guide vous propose un découpage du projet en étapes. Vous pouvez suivre ces étapes selon vos besoins. Dans chacune, vous trouverez :

- des recommandations pour compléter la mission ;
- les points de vigilance à garder en tête ;
- une estimation de votre avancement sur l'ensemble du projet (attention, celui-ci peut varier d'un apprenant à l'autre).

Suivre ce guide vous permettra ainsi :

- d'organiser votre temps ;
- de gagner en autonomie ;
- d'utiliser les cours et ressources de façon efficace ;
- de mobiliser une méthodologie professionnelle que vous pourrez réutiliser.

Gardez en tête que votre progression sur les étapes n'est qu'une estimation, et sera différente selon votre vitesse de progression.

Recommandations générales

Si vous n'êtes pas encore familier avec Angular, je vous recommande de commencer par lire [le cours dédié, "Débutez avec Angular"](#), ainsi que la [documentation officielle](#). Même si lire la documentation peut prendre du temps, cela vous permettra d'avancer beaucoup plus vite dans les étapes du projet, une fois que vous aurez assimilé les concepts et bonnes pratiques du framework.

En plus, vous pouvez relire les fondamentaux de programmation front-end si nécessaire, en suivant les cours OpenClassrooms :

[Apprenez à créer votre site web avec HTML5 et CSS3](#)
[Écrivez du JavaScript pour le web](#)

Étape 1 : Mise en place du starter code

5 % de progression

Avant de démarrer cette étape, je dois avoir :

- Git installé sur mon poste.
- Node et npm installés également.
- Compris comment dupliquer un projet depuis GitHub en utilisant Git.

Une fois cette étape terminée, je devrais avoir :

- Le projet disponible en local sur mon ordinateur.

Recommandations :

- Si vous n'êtes pas à l'aise avec Git, vous pouvez suivre ce [cours](#).
- Pour cloner le starter code, vous pouvez suivre ce [tutoriel officiel](#) (en anglais).

Étape 2 : Compréhension et définition des données

20 % de progression

Avant de démarrer cette étape, je dois avoir :

- Le projet cloné en local.
- Les données récupérées via une requête HTTP.

Une fois cette étape terminée, je devrais avoir :

- Défini des types / interfaces pour les données (TypeScript).

Recommandations :

- Assurez-vous de bien comprendre comment fonctionnent les requêtes HTTP / services avec Angular.

- Utiliser TypeScript permet de rendre JavaScript fortement typé, ce qui est très utile pour éviter les erreurs et rendre votre application plus robuste.

Ressources :

- [Documentation sur TypeScript](#).
- Chapitre du cours Angular : “[Affichez des données](#)”.
- Documentation Angular concernant les [requêtes HTTP](#).
- Documentation Angular concernant les [services](#).

Étape 3 : Choix et mise en place de la librairie

50 % de progression

Avant de démarrer cette étape, je dois avoir :

- Pris connaissance du design dans les maquettes fournies par le collègue Omar.
- Compris les interactions entre les pages.

Une fois cette étape terminée, je devrais avoir :

- Choisi une librairie pour créer des graphiques.
- Créé un premier graphique.

Recommandations :

Assurez-vous d’avoir compris comment la navigation fonctionne avec Angular, en vous aidant de la [documentation officielle](#).

Lisez bien la documentation de la librairie que vous choisirez.

Voici des exemples de librairies :

- [ngx-charts](#)
- [ng-chartist](#)
- [ng2-charts](#)

Points de vigilance :

N’oubliez pas de formater les données comme attendu par votre librairie.

Étape 4 : Création du “dashboard”

70 % de progression

Avant de démarrer cette étape, je dois avoir :

- Mis en place une librairie de graphique.

Une fois cette étape terminée, je devrais avoir :

- Implémenté la page d'accueil avec le graphique demandé dans les spécifications, en accord avec le design.

Recommandations :

Vous aurez déjà pris connaissance des designs sur Figma ; assurez-vous de les respecter en mettant en place le dashboard.

Attention, il est important que l'application soit utilisable sur tous les appareils (ordinateur, tablette et téléphone). Il faudra donc que vos interfaces soient **responsive**.

Étape 5 : Création de la page "détail"

85 % de progression

Avant de démarrer cette étape, je dois avoir :

- Créé la page "Dashboard".

Une fois cette étape terminée, je devrais avoir :

- Mis en place la navigation entre la page Détail et page d'accueil (dans les deux sens).
- La page "détail" telle que définie dans le design et les spécifications.

Recommandations :

Assurez-vous de toujours garder l'utilisateur informé pendant le changement d'état de votre application (chargement des données, erreur, etc.).

Points de vigilance :

Concernant la navigation entre les pages de l'application, prenez le temps de lire les ressources suivantes :

- Chapitre du cours Angular : "[Passez en SPA avec le routing](#)".
- [Documentation officielle](#).

Faites attention au nombre de requêtes HTTP réalisées pour récupérer les données nécessaires (vous pouvez le voir en inspectant la page, dans l'onglet "Network").

Étape 6 : Nettoyez et documentez votre code

100 % de progression

Avant de démarrer cette étape, je dois avoir :

- Terminé toutes les fonctionnalités attendues dans le cahier des charges.
- Implémenté le design de l'application.

Une fois cette étape terminée, je devrais avoir :

- Le code final de l'application, fonctionnel, et disponible sur votre GitHub en visibilité publique.

Recommandations :

- Assurez-vous d'utiliser les bonnes pratiques d'Angular (<https://angular.io/guide/security>).
- Testez votre application pour éviter les éventuels bugs : assurez-vous que les fonctionnalités se comportent comme attendu.
- Rédigez un bref README qui explique le fonctionnement de votre code et le run de l'application.

Points de vigilance :

- Assurez-vous de gérer correctement les potentielles erreurs : même si aucune erreur n'est apparente, toute application est exposée à ce risque. Pour garantir une expérience agréable à l'utilisateur, il convient de gérer correctement les erreurs inattendues.
Par exemple, une erreur déjà gérée dans le starter code peut être celle de la page non trouvée : l'utilisateur est ainsi redirigé vers une page l'informant qu'une erreur inattendue s'est produite.
Mais il existe de nombreux cas particulier à gérer ; par exemple :
 - Si vous utilisez un paramètre dans l'URL, que se passe-t-il si ce dernier n'est pas présent ou s'il est incorrect ?
 - Que se passe-t-il si l'utilisateur n'a pas accès à internet ?
 - etc.

Projet terminé !