

# Projet Front-End

Il vous est demandé de concevoir, développer et livrer une application front-end basée sur le framework Angular (version 18 minimum), s'appuyant sur la consommation d'une API publique gratuite de votre choix.

Ce projet a pour objectif d'évaluer votre capacité à mobiliser les connaissances et compétences acquises durant le module, tant sur le plan technique que sur le plan méthodologique, en produisant une application fonctionnelle, robuste, maintenable et conforme aux bonnes pratiques du développement front-end moderne.

Le projet peut être réalisé individuellement ou en binôme.

**Attention :** dans le cas d'un projet réalisé en binôme, les exigences en termes de complexité, de qualité du code, visuelles et de fonctionnalités attendues seront proportionnellement plus élevées.

Le choix du thème est libre, sous réserve que l'API utilisée soit publique et accessible gratuitement.

Dans le cas de l'utilisation d'une API personnelle ou nécessitant une clé d'authentification, celle-ci devra impérativement être valide, fonctionnelle et fournie lors de la remise du projet.

## **OBJECTIFS PÉDAGOGIQUES :**

Ce projet vise à évaluer votre maîtrise des technologies et concepts tels que Angular et son écosystème tout comme les langages TypeScript, HTML et CSS, ainsi que votre capacité à concevoir une application front-end respectant les principes fondamentaux du développement logiciel.

À ce titre, votre application devra notamment :

- Avoir un code propre, lisible et maintenable, fidèle aux principes et conventions Angular
- Concevoir et organiser le projet de manière modulaire et cohérente.
- Mettre en œuvre un système de routing structuré et robuste.
- Développer des formulaires interactifs respectant les règles de validation et d'ergonomie.
- Gérer les flux asynchrones à l'aide de RxJS, en tenant compte des états de l'application.
- Proposer une interface utilisateur claire, ergonomique et intuitive.

## **CONSIGNES TECHNIQUES :**

### **1. Choix de l'API et gestion asynchrone**

Vous devrez sélectionner une API en adéquation avec le sujet de votre application. Celle-ci devra être publique et gratuite, ou, le cas échéant, accessible via une clé personnelle fournie.

L'ensemble des interactions avec l'API devront être gérées de manière asynchrone, en utilisant RxJS et les mécanismes proposés par Angular.

Votre application devra par conséquent prendre en compte les différents états liés aux appels réseau, notamment :

- L'affichage d'un indicateur de chargement
- La gestion et l'affichage de messages d'erreur clairs en cas de problème
- La récupération et l'exploitation sécurisée des données

Quelques exemples d'API (liste non exhaustive) :

- [OpenWeatherMap](#) : données météorologiques.
- [The Dog API](#) : informations à propos des chiens.
- [PokéAPI](#) : données relatives aux Pokémon.
- [SpaceX API](#) : informations sur les missions et lancements.

## 2. Architecture logicielle et écriture de code

La qualité de l'architecture logicielle constitue un critère majeur lors du développement Angular. Il est donc attendu que l'application ne se limite pas à un fonctionnement correct, mais qu'elle présente un code structuré, lisible et maintenable.

Votre projet devra donc respecter une structure claire, avec une séparation nette des responsabilités entre les composants, les services et les autres éléments de l'application.

Une attention particulière devra être portée à l'organisation des fichiers et des dossiers du projet. La structure du code devra être propre, cohérente et intuitive, facilitant la compréhension du projet par un tiers. Les fichiers devront être rangés de manière logique, sans redondance, et refléter clairement le rôle de chaque entité. Cette organisation devra permettre une maintenance aisée ainsi qu'une évolution future du projet.

## 3. Routing

L'application devra obligatoirement utiliser le module de routing Angular.

Le système de routes devra inclure :

- une route dédiée à la gestion des erreurs (page 404)
- au minimum deux routes personnalisées, cohérentes avec le sujet et l'API sélectionnée.

Des Guards Angular devront être mis en place afin de contrôler et sécuriser l'accès à certaines routes de manière appropriée et robuste.

## 4. Formulaires

Dans la mesure des possibilités offertes par l'API choisie, vous devrez implémenter au moins un formulaire Angular, respectant les bonnes pratiques en terme d'ergonomie et validation tout en permettant :

- la recherche ou le filtrage d'éléments,
- la validation des données saisies,
- l'affichage de messages d'erreur explicites en cas de saisie invalide.

## 5. Design et Styles

Bien que l'aspect visuel ne constitue pas le cœur principal de l'évaluation, une attention sera portée à la qualité de l'interface utilisateur.

Il est attendu que l'application soit un minimum intuitive, ergonomique et visuellement claire et agréable.

L'utilisation d'un framework CSS léger est autorisée (exemples : Bootstrap, Tailwind CSS, etc.), à condition qu'il soit utilisé de manière pertinente et maîtrisée.

## OUVERTURE ET VALORISATION DES INITIATIVES PERSONNELLES

Les exigences présentées dans ce sujet constituent un socle minimal attendu pour l'évaluation du projet. Vous êtes toutefois fortement encouragés à ne pas vous limiter strictement à ces contraintes et à proposer des fonctionnalités, optimisations ou choix techniques supplémentaires pertinents.

Toute initiative personnelle, amélioration fonctionnelle, approfondissement technique ou attention particulière portée à la qualité globale du projet (architecture, performance, ergonomie, accessibilité, tests, documentation, etc.) sera prise en compte et valorisée lors de l'évaluation, dans la mesure où ces ajouts sont cohérents, justifiés et correctement intégrés à l'application.

## LIVRABLES ATTENDUS

La remise du projet devra inclure les éléments suivants :

- Le projet Angular complet ( $\geq$  ng18), fonctionnel et exécutable (sans `/node_modules`). Il peut être fourni via un lien Git ou un fichier compressé (.zip).
- Un document texte contenant :
  - Les membres du groupe
  - L'API sélectionnée avec un lien vers la documentation (une justification technique du choix de l'API est optionnelle mais appréciable)
  - Des règles de lancement de l'API si celle-ci est personnelle
  - Tout autre informations concernant le projet qui vous semblerai utiles, comme des spécifications utiles, des axes d'améliorations, des difficultés rencontrées, ...

**Point de vigilance** : L'utilisation abusive, non maîtrisée ou non justifiée de l'IA entraînera des pénalités sur l'ensemble de l'évaluation.

**Date limite de rendu** : 22 Février 2026, 23:59