

Planification de l'architecture du Front-office

Basé sur le template de Dan Whalin

[@DanWahlin](https://twitter.com/DanWahlin)

<https://codewithdan.com>

Obtenez le template ici:

<http://codewithdan.me/angular-architecture-planning>

© Wahlin Consulting



Aperçu global
de l'application

Fonctionnalités
de l'application

Sécurité du
domaine

Règles du
domaine

Logging

Services/
Communication

Modèles de
données

Composants
fonctionnels

Fonctionnalités
partagées /
réutilisées

Planning

Utilisez les diapositives suivantes pour documenter les informations initiales sur votre application. L'objectif est de fournir un **moyen simple** de discuter et de documenter les aspects clés de votre application tout en gardant les discussions **hyper-focalisées**.

Dan Wahlin

[@DanWahlin](https://twitter.com/DanWahlin)

Aperçu global de l'application

Principaux objectifs

- Supporter la consultation de produits
- Supporter la consultation et création de commandes
- Supporter la consultation et remplissage de paniers

Principales exigences

- Afficher les produits sous forme de cartes / grille
- Supporter le filtrage, le tri, la pagination des produits
- Afficher les commandes avec pagination
- Supporter l'authentification/déconnexion avec email/mot de passe

Fonctionnalités de l'application

- Produits
 - Afficher les produits
 - Supporter les modes d'affichage grille/carte
 - Supporter le filtrage/la pagination/le tri
- Produit
 - Afficher les détails d'un produit
- Client
 - Créer/Mettre à jour/Supprimer un compte client
 - Afficher les détails du client/les commandes
 - Le formulaire doit être validé pré-envoi
- Commandes
 - Afficher les commandes
 - Supporter la pagination
- Login/Logout

Sécurité du domaine

- Authentification locale (email/mot de passe) pour la première version
- Jetons JWT pour une version future
 - Utilisation d'un intercepteur HTTP pour placer les jetons en en-tête automatiquement

Règles du domaine

- Chaque commande doit être associée à un utilisateur
- Les totaux des commandes doivent être affichés pour un client donné
- Le formulaire d'édition des clients doit être validé avant son envoi au serveur
- Les utilisateurs doivent être authentifiés pour consulter/modifier leurs informations personnelles et achever une commande
- Le login et mot de passe doivent être validés

Logging

Un service est responsable des logs, et le logging est effectué dans différentes zones en fonction de dev/stage/prod :

- Console logger
- Remontée des logs critiques au serveur

Services/Communication

- Services RESTful
- Services Angular :
- Services de données : utilisation du HTTPClient
 - Clients / Commandes / Produits
 - Login / Logout
- Service de logs (Logger)
- Mediateur/ Event bus si besoin
- HttpInterceptor pour placer automatiquement les jetons d'authentification en en-tête si utilisés

Modèles de données

Les modèles de données sont représentés à l'aide d'interfaces côté client (angular)

Interfaces pour représenter :

- Client
- Commande (détails d'une commande ?)
- Produit
- Auth (login / password)

Voir les JavaBeans réalisés dans le backend pour avoir une idée des champs disponibles.

Composants fonctionnels

Fonctionnalités partagées / réutilisées

- Toast/ Growl pour afficher des notification en cas de succès/échec/information
- Modaux (fenêtres de dialogue) pour demander la confirmation d'actions
- Une carte customisée pour l'affichage d'un produit

Fonctionnalités partagées (externes)

- Angular Material Design
- Bootstrap (CSS uniquement, pour le système de grille ?)
- Spécifiques au dev. :
 - Prettier, pour le formatage automatique de toute la code-base