Objectif de la semaine Il ne s'agit pas simplement de savoir faire des call API mais de savoir où les faire dans l'application frontend et comment architecturer de la façon la plus efficace possible son code afin maximiser sa maintenabilité

1 Les requêtes AJAX/Call API

1. L'architecture 2-tiers (Client - Serveur)
2. L'architecture 3-tiers (Client -Serveur d'application - Serveur de BDD)
3. Les requêtes HTTP
4. Le Client Navigateur / Le demandeur de service
5. La latence potentielle
6. L'asynchronisme
7. Le fournisseur de service / le backend

Le changement de nommage les requêtes AJAX -> les requêtes sous format JSON

pré-requis

1. La communication entre les différentes couches de l'architecture de l'application côté client
2. La compréhension des tableaux, des objets
3. La compréhension de l'asynchronisme
4. La compréhension des fonctions de rappel

2 La compréhension des couches d'une architecture client

1. la couche de présentation (ui) avec des dossiers comme pages et components
2. La création d'une couche s'occupant de la logique métier des pages
3. La création d'une couche s'occupant de la communique avec l'extérieur

Logique métier Le dossier containers

Communique avec l'extérieur Le dossier services et le dossier data-sources

3 La couche de présentation

1. Révision de la modulation du code avec les composants
2. Appel des composants dans des composants parents (les pages)
3. Révision des objets et des tableaux
4. Révision des conditions et du ternaire pour rendre les composants réutilisables

4 La couche de la logique métier

1. Initiation au classe
2. Une classe instanciable pour contenir la logique métier d'une page
3. Le constructeur
4. Les méthodes d'instance
5. Les méthodes de classe
6. La gestion des événements (ex : submit) de l'interface

5 Un service dédié à un objet domaine

1. Le dossier services
2. Un service de gestion d'un objet domaine/métier
3. Un service qui fait appelle à une classe de réalisation de requêtes HTTP

6 La couche de communication vers l'extérieur

1. Le dossier data-sources
2. Une classe aux méthodes statiques réutilisables
3. Une classe de réalisation de requêtes GET, etc
4. Les verbes HTTP et leur fonctions