

# Contenu

## Base

### Structure de données

- types scalaires standards (int / float / double / boolean /...)
- Array / List / Deque / Heap (FIFO - LIFO)
- Arbre et graphe
- Notions de complexité des algorithmes sur les conteneurs

Ressources :

- [Tutorial FR - Quatre méthodes de recherche dans les tableaux en JavaScript](#)
- [Tutorial FR - Comment utiliser .map\(\) pour itérer via Array Items dans JavaScript](#)
- [Tutorial FR - Comprendre les objets Map et Set en JavaScript](#)

### Notions de réseau

- outils de debug (ping curl telnet nslookup)
- Adresse IP / Masque de sous réseau / routage
  - [Course FR complete - Les bases du réseau : TCP/IP, IPv4 et IPv6](#)
- DNS
  - [Guide FR basic - Les bases du DNS](#)
- SSL & PKI
  - [Guide FR complete - SSL / TLS 101 pour les débutants](#)
- autre
  - Couche du modèle OSI : [Qu'est-ce que le modèle OSI ?](#)
  - [Guide FR basic - Le fonctionnement d'HTTP](#)
  - [Guide FR basic - Comprendre le protocole TCP](#)
    - Différent mode de discussion : connexion stateless vs statefull (ex: FTP vs HTTP)

### Outillage

- ligne de commande
  - [Mémo : Commandes de base Linux](#)
  - [Guide EN - Learn Enough Command Line to Be Dangerous](#)
- git / gitlab
  - [Guide FR complete - Pro Git](#)
- SSH
  - [Comprendre et maîtriser SSH](#)

- docker & docker-compose
  - [Tutorial FR complete - Docker : découverte des bases](#)
  - [Tutorial FR - Reverse Proxy Traefik avec Docker Compose et Docker Swarm](#)
- debugger
- profiler
- vagrant
- ansible
  - [Tutorial FR - Découverte d'Ansible par un](#)
- test unitaires
- CI & CD
  - [Turorial EN - Docker, Kaniko, Buildah](#)
  - [Tutorial FR - Builder simplement des images docker avec gitlab-ci \(sans dind\)](#)

## Languages

- POO
- Notions de programmation fonctionnelle
  - [Guide FR - Programmation fonctionnelle en JavaScript](#)
  - immutability
  - iterators
  - pattern matching

## Conception et gestion de projet

### UML

[What is UML - Visual Paradigm \(EN\)](#)

- Use case
- Diagramme de classe
  - Différents types de relation [Aggregation vs Composition](#)
- Diagramme de séquence

### Outils

- Markdown
- Mermaid / PlantUML / draw.io

## Architecture & patterns

- MVC
- REST
- SOA

# Bonnes pratiques

- Review de code pour maîtriser ses commits
- [Tutorial FR - SOLID : les 5 premiers principes de conception orientée objet](#)
- Clean code & Design patterns : [Refactoring Guru](#)
- Early exits

# UI

- [Tutorial FR - Comment utiliser l'API Fetch de JavaScript pour récupérer des données](#)

## VueJS pour Web et Desktop (avec Electron)

- [Guides EN](#)
- [Guides FR \(trad tjs en cours au 2021-10-08\)](#)

## ReactNative pour Mobile

- [Tutorial EN - basic](#)
- [Guides EN](#)

# Web backend

## PHP Vanilla / Symfony / Laravel

- [Tutorial EN - Full vanilla implementation](#)
- [Tutorial FR - APIPlatform @Kaherecode](#)
- [Tutorial FR - Créer une API JSON avec Laravel](#)

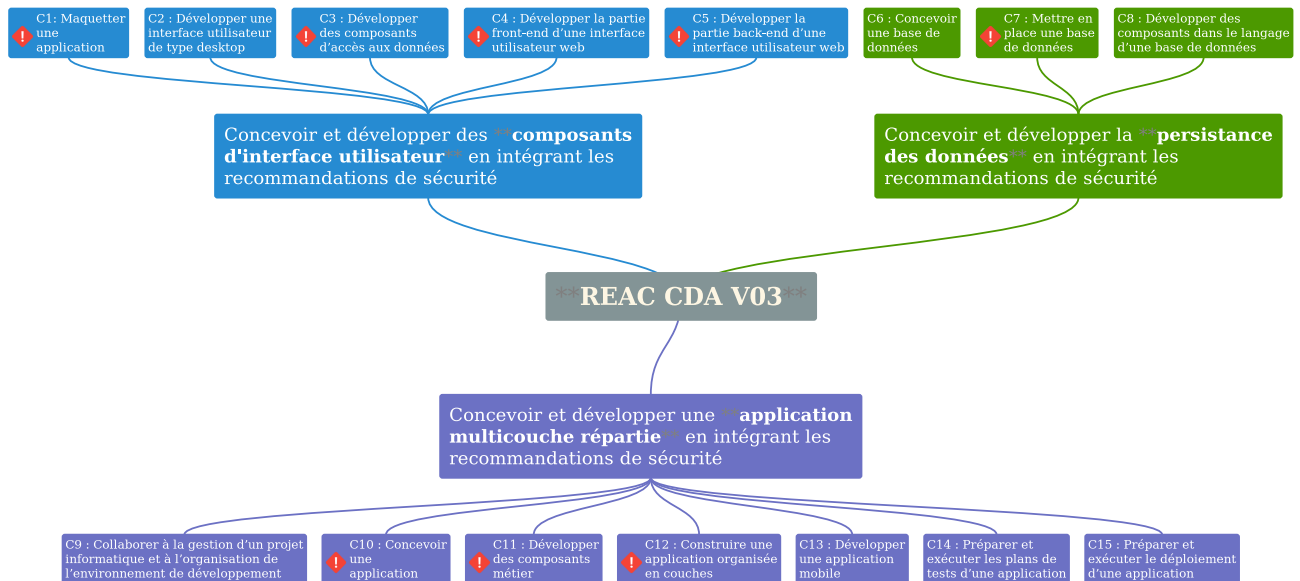
## Node & Express

- [Guides FR](#)

# Persistence

- Composants BDD : view / stored procedure / trigger
- Transactions :
  - [PHP | Transactions and Concurrency](#)
  - [Bien débuter avec les transactions SQL](#)
- Piliers SGBDR : [Propriétés ACID](#)
- SQL & NoSQL : [Théorème de CAP](#)

# Programme



## Démarrage avec des ateliers sur la partie "base" du contenu

- positionnement
- ajuster en fonction des retours

## Préparation du projet fil rouge

- déterminer le projet
- planification et démarrage des conceptions (**C9 & C10**)

## Mise en place du backend

- BDD et ORM (**C6 & C7 & C8**)
- contrôleurs REST (**C5 & C12**)
- logique métier à déterminer en fonction du projet (**C11**)
- test unitaires (**C14**)
- CI / CD avec Gitlab (**C14 & C15**)

## Application client Web en VueJS (C4)

- Maquettage (**C1**)
- MVC (**C12**)
- Utilisation du Store (**C3 & C12**)
- Utilisation du routeur (**C12**)
- exportation sous forme desktop avec Electron (**C2**)

- utilisation d'un UIKit complet (**C2**)

## **Application mobile en ReactNative (C13)**

- Maquettage (**C1**)
- MVC (**C12**)
- Utilisation du routeur (**C12**)
- utilisation d'un UIKit dédié (**C13**)