

# Sujets pour le projet informatique de deuxième année (2020-2021)

---

## Sujets pour le projet informatique de deuxième année (2020-2021)

### Création d'une API REST

Présentation

Fonctionnalités de base attendues

Fonctionnalités de avancé

Compléments

### Codez votre jeu Pokemon en Python

Présentation

Fonctionnalités de base attendues

Fonctionnalités avancées

### Bibliothèque de musiques

Présentation

Fonctionnalités de base attendues

Fonctionnalités avancé

### Tableau de bord COVID-19

Présentation

Fonctionnalités de base attendues

Fonctionnalités avancées

Compléments

### Location de vélos urbains

Présentation

Fonctionnalités de base attendues

Fonctionnalités avancées

Compléments

### API-HOUR

Présentation

Fonctionnalités de base attendues

Fonctionnalités avancées

### Trading de cryptomonnaie

Présentation

Fonctionnalités de base

Fonctionnalités avancées

### Création d'un jeu de cartes

Présentation

Fonctionnalités de base attendues

Fonctionnalités de avancé

## Création d'une API REST

---

Tuteur : Thierry Mathé

## Présentation

Les API Web sont des outils informatiques permettant à deux applications d'échanger des informations. Pour standardiser les API web, Roy Fieldring a proposé en 2000 le "representational state transfer". L'objectif de ce projet est la création d'une API reposant sur des requêtes HTTP respectant certains aspects de ce standard.

## Fonctionnalités de base attendues

Le minimum syndical. De préférence doit permettre de :

- Consommer un webservice / import-export massif de données json/xml/html
- Manipuler des données en base intéressante (pas uniquement des comptes pour la connexion)
- Une vraie fonctionnalité métier facile
- Une vraie fonctionnalité métier moyenne

## Fonctionnalités de avancé

Liste de fonctionnalités intéressantes. Dans un monde parfait si tout les groupe pouvaient en faire une ça serait bien

## Compléments

Partie non obligatoire, mais si vous avez envie d'ajouter des choses :

- Manipuler des données en base intéressante (pas uniquement des comptes pour la connexion)
- Une vraie fonctionnalité métier facile
- Une vraie fonctionnalité métier moyenne



## Codez votre jeu Pokemon en Python



Tuteur : Sylvain Leveugle

## Présentation

Le but de l'application sera de coder quelques fonctionnalités simples d'un mini jeu pokemon. La connaissance du jeu n'est pas nécessaire.

Les élèves auront accès à l'API <https://pokeapi.co/> afin de récupérer les informations sur les pokemon.

Le jeu sera jouable en format texte, et permettra de gérer son équipe de Pokemon, partir en rencontrer d'autres, ou affronter d'autres dresseurs.

## Fonctionnalités de base attendues

L'application devra au minimum :

- Récupérer les informations liées au pokemon sur l'API fournie
- Stocker dans une base de données son pokemon actuel, plus une réserve de pokemon
- La base de données contiendra aussi des listes de dresseurs avec des pokemons prêts à être affrontés
- Le pokédex (la liste des pokemon rencontrés) devra aussi être stocké en base de données et accessible par le joueur
- L'utilisateur de l'application devra pouvoir :
  - Aller rencontrer des pokemon sauvages pour les affronter/les attraper
  - Aller affronter des dresseurs de pokemon pour gagner de l'argent
  - Acheter des pokeballs pour attraper des pokemons
  - Stocker/retirer des pokemons de son PC (la base de données)

- La gestion des stats de combat, des attaques... pourra être gérée de manière simplifiée si nécessaire

## Fonctionnalités avancées

Ces autres fonctionnalités peuvent être implémentées pour enrichir le sujet :

- Affichage des photos des pokemons par l'application
- Gérer les faiblesses des pokemon en combat
- Pouvoir échanger des pokemon avec un autre utilisateur (difficile)
- Pouvoir affronter un autre utilisateur (difficile)
- Avoir des équipes de pokemon plutôt qu'un pokemon
- Autre fonctionnalité au choix

## Bibliothèque de musiques

---

Tuteur : Jean-Michel Leclercq

### Présentation

Utilisation de l'API : <https://openwhyd.github.io/openwhyd/API>

### Fonctionnalités de base attendues

- Recherche de titres via l'API
- Gestion d'un compte applicatif
- Constitution de playlists :
  - Ajout de chanson
  - Suppression de chanson
- Ajout de 2 fonctionnalités de votre choix :)

### Fonctionnalités avancé

- Lire le flux audio (Youtube / SoundCloud)
- Afficher / Lien vers la couverture / image de la chanson

## Tableau de bord COVID-19

---

Tuteur : Frédéric Chantrel

### Présentation

Explorer les données en open data, les importer dans une base, les exposer à travers une API, construire un tableau de bord en mode console

Vous commencerez par ingérer les données dans une base de données, ensuite vous mettrez à disposition ces données à travers une API REST et enfin vous construirez une IHM en mode console présentant un tableau de bord. Cette IHM devra utiliser l'API que vous aurez développée au préalable.

## Fonctionnalités de base attendues

Produire les indicateurs suivants par jour et/ou période, par département, par région et par catégorie d'âge :

- Le nombre de patients ayant été hospitalisés pour COVID-19 et de retour à domicile en raison de l'amélioration de leur état de santé
- Le nombre de patients actuellement hospitalisés pour COVID-19
- Le nombre de patients actuellement en réanimation ou soins intensifs
- Le nombre cumulé de décès de patients hospitalisés pour COVID-19 à l'hôpital
- Le nombre de tests de dépistage de COVID-19 réalisés en laboratoire de ville et le nombre de positifs

Calculer les 2 indicateurs suivants :

### 1. Activité épidémique (taux d'incidence)

Le taux d'incidence correspond au nombre de tests virologiques positifs pour 100.000 habitants sur une semaine. Il est calculé, à l'échelle départementale, de la manière suivante :  $(100000 \times \text{nombre de cas positif}) / \text{population}$ .

Trois niveaux ont été fixés pour cet indicateur :

- vert : en dessous de 10 personnes testées positives sur 100.000 personnes testées, sur une semaine glissante ;
- orange : au-delà de 10 personnes testées positives sur 100.000 personnes testées, sur une semaine glissante ;
- rouge : au-delà de 50 personnes testées positives sur 100.000 personnes testées, sur une semaine glissante.

### 2. Taux de positivité des tests virologiques

Le taux de positivité correspond au nombre de personnes testées positives sur une semaine. Il est calculé, à l'échelle départementale, de la manière suivante :  $(100 \times \text{nombre de tests positifs}) / \text{nombre de tests réalisés}$ .

Trois niveaux ont été fixés pour cet indicateur :

- vert : taux de positivité entre 0 et 5% ;
- orange : taux de positivité entre 5 et 10% ;
- rouge : taux de positivité supérieur à 10%.

## Fonctionnalités avancées

Si le temps le permet vous pourrez utiliser d'autres sources de données pour enrichir votre analyse ; par exemple travailler sur l'évolution du R0 ou sur la tension hospitalière sur la capacité en réanimation mais toute autre suggestion sera la bienvenue.

## Compléments

Les sources de données suivantes pourront être utilisées :

- population par département : <https://public.opendatasoft.com/explore/dataset/population-francaise-par-departement-2018/table/?flg=fr&disjunctive.departement>
- lien entre départements et régions : <https://www.data.gouv.fr/fr/datasets/departements-et-leurs-regions/>
- données hospitalières relatives au COVID-19 : <https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-relatives-a-lepidemie-de-covid-19/>
- données relatives aux tests de dépistage : <https://www.data.gouv.fr/fr/datasets/donnees-relatives-aux-tests-de-depistage-de-covid-19-realises-en-laboratoire-de-ville/>

Tuteur : Geoffrey PAGNIEZ

## Présentation

VéloStar, Bicloo, Vélib... Si ces noms vous parlent c'est que le domaine de la location de vélos urbains est en plein développement. Cependant, il y a autant d'applications que de réseau... mais pourquoi ne pas en développer une seule ? Grâce à l'API [CitiBikes](#), c'est maintenant possible ! Le but de cette application va être d'exploiter les données publiques de la location de vélo urbains et de fournir un service pour faciliter leur utilisation.

## Fonctionnalités de base attendues

- Affichage des villes possédant un système de location de vélos
- Affichage des stations par ville
- Gestion de favoris (villes et stations de location)
- Affichage du nombre de vélos dans une station
- Etablissement d'un itinéraire (station de départ -> station d'arrivée)

## Fonctionnalités avancées

- Localisation de la station la plus proche (simulation de positionnement GPS)
- Propositions de stations avec vélos disponibles la plus proche
- Statistiques simples sur le nombre d'emplacements libres et de vélos par stations
- Toute idée originale est bienvenue :)

## Compléments

Dans une démarche écologique, votre application devra appeler le moins possible l'API et remonter uniquement les informations nécessaires



Tuteur : Donatien ENEMAN

## Présentation

Vous en avez marre du Get 27 et de la vodka grenadine ? Votre BDE manque d'inspiration pour les Chartres ? Que faire avec un fond de bouteille de rhum ? A vous de créer l'application qui permettra de résoudre tous ces problèmes.

Le but de votre application sera de constituer un moteur de recherche de cocktails. Pour cela vous vous appuyerez sur l'API : <https://www.thecocktaildb.com/api.php>.

## Fonctionnalités de base attendues

- Effectuer des appels à l'API afin de récupérer des cocktails en fonction de leur nom, ou d'ingrédients.
- Pouvoir ajouter des recettes de cocktails (base de données).
- Donner des avis sur des cocktails.
- Gestion de favoris (ajout et suppression).
- Gestion d'historique.
- Une gestion de compte applicatif (non connecté, connecté, admin).

## Fonctionnalités avancées

- L'API dans sa version gratuite ne prend en charge que les recherches avec un seul paramètre. Implémentez à votre manière une recherche à plusieurs paramètres.
- Traduire les descriptions des cocktails en français (utilisation d'une deuxième API, et oui vive l'architecture micro service :) ).
- Affichage de l'image du cocktail.
- Autres idées de votre part.

## Trading de cryptomonnaie

---

Tuteur : Charles NGOR

### Présentation

Le projet consiste à créer une application permettant de faire du trading de cryptomonnaies. Vous pourrez acheter et vendre des cryptomonnaies tel que le Bitcoin dans le but de réaliser des bénéfices et devenir millionnaire.

### Fonctionnalités de base

- Vous disposerez d'un montant initial d'argent à investir
- Vous pourrez acheter puis vendre au minimum 3 cryptomonnaies différentes
- Consulter l'évolution de votre solde
- Consulter vos transactions en cours
- Consulter le prix actuel des cryptomonnaies
- Afficher l'historique de vos transactions avec les bénéfices et pertes

Si votre enveloppe atteint zéro, votre compte sera liquidé et votre investissement définitivement perdu.

Pour récupérer les données des prix des cryptos, vous utilisez l'API (service Web) CoinGecko : <https://www.coingecko.com/en/api>

Une bibliothèque Python est disponible pour communiquer avec l'API CoinGecko. A chaque démarrage de l'application, vous devrez utiliser cette API pour récupérer l'historique des prix et les stocker dans une base de données. Il est conseillé d'économiser le volume de données téléchargé à partir de l'API pour des raisons d'ordre écologique et de performance.

### Fonctionnalités avancées

Vous pourrez consulter l'évolution des prix moyens découpés par jour, par tranche de 4 heures, par heure afin analyser le cours de l'action. Vous êtes libre de proposer des fonctionnalités complémentaires.

*Le trading présente un risque élevé de perdre de l'argent rapidement.*

## ♥ ♠ Création d'un jeu de cartes ♦ ♣

---

Tuteur : Corentin Brémard

# Présentation

Vous allez développer votre propre jeu de cartes.  
Du solitaire au poker, à vous de choisir votre jeux préféré.

Vous aurez à disposition une API (<http://deckofcardsapi.com/>) vous permettant de mélanger et tirer vos cartes. Votre application se chargera du reste.

A titre d'exemple, voici [les 10 jeux de cartes préférés des français](#). Ou alors d'autres variantes avec [un simple paquet de cartes](#)

## Fonctionnalités de base attendues

- Gestion des joueurs (création/suppression/modification) via une base de données
- Définition des différentes actions réalisables par un joueur
- Sauvegarde de parties en BDD

## Fonctionnalités de avancé

Etes-vous capable de rajouter un nouveau jeux à votre application en moins de 5 minutes ?  
Ou alors comment rajouter de nouveaux jeux en changeant le moins de code possible ?

A vous d'abstraire la notion de "jeux de cartes" afin de présenter une nouvelle achitecture d'application avec laquelle vous seriez capable de rajouter un nouveau jeux en ne modifiant que quelques lignes de code.