

# ANNUAIRE DE CONTACTS

Ou contacts dans l'annuaire, au choix...

## Contenu

---

Description du projet.....	1
Fonctionnalités attendues.....	1
Découpage de l'application .....	2
Étapes du projet.....	2
Tester mon code.....	3

## Description du projet

---

Vous connaissez probablement déjà le principe d'un annuaire de contacts. Une liste de personnes sauvegardées dans une application ou un répertoire papier (!!!) avec leur coordonnées (nom, prénom etc...).

L'avantage des annuaires de contact électronique est la possibilité de facilement mettre à jour la liste (ajout, modification, suppression) et de pouvoir effectuer des recherches via des filtres et fonctions de tri.

Vous avez la mission de développer un annuaire de contact modulaire et extensible !

## Fonctionnalités attendues

---

Les fonctionnalités attendues sont:

contact = une personne + ses coordonnées  
groupe de contact = un ensemble de contacts

### Contacts

- Ajouter un contact
- Modifier un contact
- Supprimer un contact
- Bloquer un contact
- Lister les contacts
- Rechercher des contacts

### Groupes de contacts

- Ajouter un groupe de contact
- Modifier un groupe de contacts
- Supprimer un groupe de contacts
- Ajouter un contact dans un groupe
- Supprimer un contact d'un groupe
- Ajouter un groupe dans un autre groupe (sous-groupe)
- Afficher les contacts d'un groupe

3 groupes de contacts existent par défaut dans l'annuaire:

- Default : tous les contacts sans exception
- Favorites : un groupe spécial pour ajouter au maximum 10 contacts "favoris"
- Recent : un groupe automatique qui liste les 10 derniers contacts ajoutés ou modifiés

Ces 3 groupes ne peuvent pas être renommés ou supprimés.

## Découpage de l'application

---

Le logiciel sera divisé en 2 applications (et donc 2 phases distinctes) :

- 1) Une API NodeJS Rest restituant les données au format JSON
- 2) Une application HTML/CS/JS permettant de manipuler ces données dans un navigateur

## Étapes du projet

---

La 1<sup>ère</sup> étape consiste à créer un dépôt git (sur github ou gitlab) et d'y inviter les membres de votre équipe.

/!\ 1 seul dépôt par équipe ! /!\

Le dépôt doit respecter la structure suivante :

```
nom_du_projet
  /docs/
  /src/
  /src/api/
  /src/app/
  /tests/
  README.md
```

Le répertoire "**docs**" contient la documentation du projet au format Markdown, HTML ou PDF.

Le répertoire "**src**" contient 2 sous répertoires :

- **api** : contient le code source de l'api (phase 1)
- **app** : contient le code source de l'application frontend (phase 2)

Le répertoire "**tests**" contient les fichiers de tests de vos objets.

Au début, vous devrez réaliser le travail qui sera demandé via le système de tickets de github.

Cela concerne essentiellement la modélisation des modèles d'objets nécessaire à l'application.

Une fois la couche "Modèle" validée, elle sera rendue visible en tant qu'API Rest.

La 2<sup>nde</sup> phase consistera à développer l'application web HTML permettant d'interagir avec l'API.

Un document PDF détaillé vous sera fourni à ce moment là.

Vous l'avez compris, tout le code NodeJS se trouve sous le répertoire `"/src/api/"`.

Le répertoire `"/src/app/"` sera utilisé dans la 2<sup>nde</sup> phase du projet.

## Tester mon code

---

Si vous développez un objet (une classe) ou une fonction, il est intéressant de pouvoir tester son travail.

Le répertoire "**tests**" est prévu pour cela.

Par exemple, je dois implémenter une classe "Person".

- 1) je code ma classe et en fait un module NodeJs ( `module.exports = Person;` )
- 2) Dans le répertoire "**tests**", je crée un fichier "Person\_test\_mickael.js"
- 3) J'importe ma classe ( `const Person = require('../src/api/Models/Person.js');` )
- 4) J'y place mon code de test (créer des Person, les modifier, les afficher dans la console...)
- 5) Je teste 😊

---

--- FIN DU DOCUMENT ---

<http://www.arfp.asso.fr>