# PROJET DU BUSINESS CASE POUR L'EXAMEN DE FORMATION DEVELOPPEUR WEB/WEB MOBILE

curl -H "Accept: application/vnd.github.v3+json" <https://api.github.com/repos/kurtonuet/sf-api-pressing/commits>

# easyadmin upload file

composer require symfony/mime composer require vich/uploader-bundle

* docs : vich/uploader-bundle <https://github.com/dustin10/VichUploaderBundle/blob/master/docs/installation.md>

# Test

composer require --dev symfony/test-pack symfony/http-client

created et update entity trait php voir gpt php bin/console lexik:jwt:generate-keypair --overwrite

## À FAIRE

API

* created et update entity
* choix du service -> article -> quantité
* Faire une route pour récupérer les commandes sans employé : api/commandes/noassign (fait)
* Faire une route pour récupérer les commandes du client : api/client/commandes (fait)
* Faire une route pour Assigner une Commande a un employé ROLE\_Employee: api/employee/items (fait)
* Faire une route pour Assigner une Commande a un employé ROLE\_Employee: api/employee/items
* Faire une route pour Employé qui indique qu'il a fini l' item Employee:api/item/{id}/complete
* Faire avec les catégories parent

### Authentification

### Création d'un projet API

symfony new sf-api-pressing --version=6.4

### Installation des composants

Installation du bundle orm-fixtures (fait)

composer require --dev orm-fixtures

Installation du bundle fakerphp (fait)

composer require --dev fakerphp/faker

Installation du bundle symfony/orm-pack (fait)

composer require symfony/orm-pack

Installation du bundle maker (fait)

composer require --dev maker

Installation du bundle security (fait)

composer require security

Installation du bundle API (fait)

composer require api

Installation du bundle lexik JWT (fait)

composer require lexik/jwt-authentication-bundle

php bin/console lexik:jwt:generate-keypair

### CREATION DES ENTITEES

security: is\_granteg("ROLE\_ADMIN")

* Article.php (fait v2)
* Material.php (fait v2)
* Meansofpayment.php (fait v2)
* Category.php (fait)
* Client (fait)
* Commande.php (fait)
* Employee (fait)
* Item.php (fait)
* ItemStatus.php (fait)
* Service.php (fait)
* User.php (fait)

### Configuration Pour JWT

* security.yaml (fait) - firewall (fait) - acces control (à terminer)
* route.yaml (fait)

## FIXTURES

* Article.php (fait)
* Category.php (fait) (avoir)
* Commande.php
* Item.php
* ItemEtat.php (fait)
* Material.php (fait)
* Meansofpayment.php (fait)
* Service.php (fait)
* ServiceStatus.php (fait)
* User.php (fait) ok

## GROUPS DE SERIALISATION

##### Category.php

- category:list:read - category:write

##### Client.php

- client :read - client :write

##### Commande.php

- commande:read - commande:write

##### Employee

- employee:read - employee:write

##### Item.php

- item :read - item :write

##### ItemStatus.php

- itemStatus:read - itemStatus:write

##### Service.php

- service :read - service :write

##### User.php

- user:read - user:write  
  
# Page côté front

### Dashboard avec EasyAdmin 4(a faire)

Installation du bundle EasyAdmin (fait)

composer require easycorp/easyadmin-bundle

création du dashboard (fait)

php bin/console make:admin:dashboard

Creéation des CRUD pour chaque entité (fait)

php bin/console make:admin:crud

# FRONT

### Login

routes: /api/check\_login

## Visiteur

### Landing page

liste 3 services

routes: /api/services

### Présentation des services

routes: /api/services

## Espace Utilisateur

## Espace Client

#### Page dépôt

### Processuse de dépot

>  
>- 1 Choix du service  
> - route: GET /api/services  
>- 2 Choix des article ManyToMany entre SERVICE et ARTICLE  
> - routes: GET /api/articles  
>- 3 Choix du quantité  
> - route: /api/services  
>- 4 Valider le dépot  
> - route:POST /api/items

### Page liste commandes

controller route personnalisé : /api/commande/user/{id}

### Page panier

stocker dans le localStorage

### Page liste Profile

routes: /api/user/{id}

### Page liste contact

routes: /api/contact/

## Espace Admin

#### Page liste des ustilisateurs

routes: /api/users

#### Page liste commandes

#### Page liste des taches

#### Page Profile

## Espace Employee

#### Page liste des taches

#### Page Profile

#### Page contact

### Espace Manager

### Espace Employee

## Employé - admin

list des commande pour l'employer

## Admin

composer require easycorp/easyadmin-bundle php bin/console make:admin:dashboard

* Admin
  + liste de commande
* routes: /api/commandes
  + liste des users
* routes: /api/user
  + employer
  + client

| **Bundle** | **Commande** |
| --- | --- |

|  |  |
| --- | --- |
| orm-fixtures | composer require --dev orm-fixtures |

|  |  |
| --- | --- |
| fakerphp/faker | composer require --dev fakerphp/faker |

|  |  |
| --- | --- |
| symfony/orm-pack | composer require symfony/orm-pack |

|  |  |
| --- | --- |
| maker | composer require --dev maker |

|  |  |
| --- | --- |
| security | composer require security |

|  |  |
| --- | --- |
| API | composer require api |

|  |  |
| --- | --- |
| lexik/jwt-authentication-bundle | composer require lexik/jwt-authentication-bundle |