

**Formulaire d'inscription** : aucune obligation de remplir les champs correctement, pour le mail il suffit de mettre une lettre et un @, le numéro de téléphone, l'adresse, le SIRET, la raison sociale ne subissent aucune vérification.

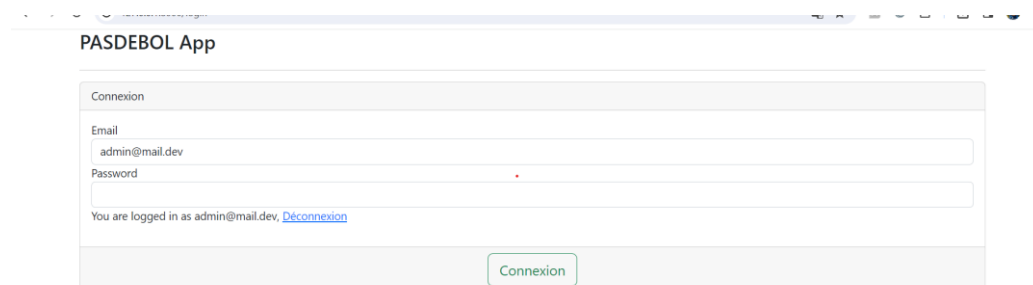
Le mot de passe ne respecte pas la RGPD, car il suffit de 6 caractères, et il n'y a aucune obligation de caractères (chiffres, caractères spéciaux, majuscule...).

Lors de la création d'un compte utilisateur, aucun rôle ne sera attribué.

**Formulaire de connexion** : le formulaire de connexion n'est pas obligatoire on peut se rendre sur les pages suivantes sans y accéder en copiant l'URL :  
<http://127.0.0.1:8000/espace-entreprise/dashboard> pour accéder au « compte » utilisateur  
et <http://127.0.0.1:8000/backoffice/dashboard> pour accéder au « compte » administrateur.

Le mot de passe est souvent identique ou similaire entre chaque utilisateurs et admins.

Si on ferme le serveur ou qu'on ouvre une autre page, notre token de connexion reste et le système indiquera qu'on est déjà connecté.



The screenshot shows a web browser window with the title "PASDEBOL App". Inside the browser, there is a login form titled "Connexion". The form has two input fields: "Email" with the value "admin@mail.dev" and "Password" which is empty. Below the password field, there is a message: "You are logged in as admin@mail.dev, [Déconnexion](#)". At the bottom of the form, there is a green button labeled "Connexion".

**Security.yaml** : les rôles ne sont pas utilisés car dans le fichier security.yaml, les rôles admin et user sont commentés.

```

# where to redirect after logout
target: app_login

# activate different ways to authenticate
# https://symfony.com/doc/current/security.html#the-firewall

# https://symfony.com/doc/current/security/impersonating_user.html
# switch_user: true

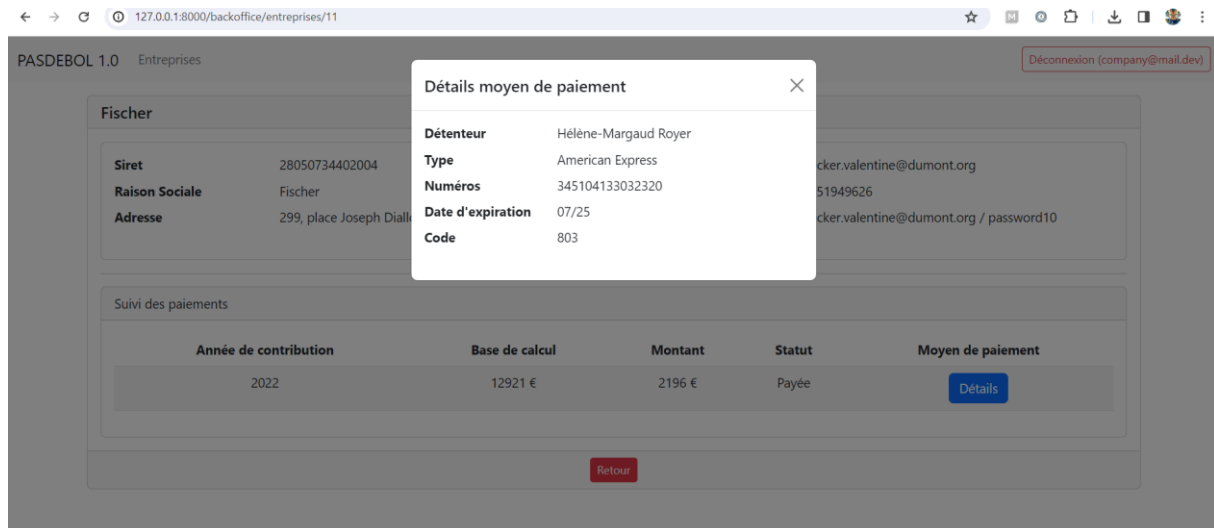
# Easy way to control access for large sections of your site
# Note: Only the *first* access control that matches will be used
access_control:
    # - { path: ^/admin, roles: ROLE_ADMIN }
    # - { path: ^/profile, roles: ROLE_USER }

when@test:
    security:
        password_hashers:
            # By default, password hashers are resource intensive and take time. This is
            # important to generate secure password hashes. In tests however, secure hashes
            # are not important, waste resources and increase test times. The following
            # reduces the work factor to the lowest possible values.
            Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:
                algorithm: auto
                cost: 4 # Lowest possible value for bcrypt
                time_cost: 3 # Lowest possible value for argon
                memory_cost: 10 # Lowest possible value for argon

```

**Page user** : dans la page <http://127.0.0.1:8000/espace-entreprise/contribution/11> , il suffit de changer le numéro après contribution pour pouvoir accéder à toutes les contributions qui sont inscrites dans le site quel que soit l'utilisateur.

**Page admin** : dans la page <http://127.0.0.1:8000/backoffice/entreprises/> , on peut avoir accès à toutes les données à caractère personnel sur n'importe quel utilisateur, nom, prénom, numéro de téléphone, coordonnées bancaires... ; adresse mail et mot de passe du compte.



**Base de données :** accès à la base seulement avec le nom d'utilisateur root, sans devoir inscrire un mot de passe, ni même un nom d'utilisateur personnalisé.

Dans la table payment, on peut avoir accès à toutes les données bancaires des clients.

Dans la table user, les passwords sont hachés mais le plainpassword est inscrit donc le mot de passe est visible en clair.

Dans la table company, il y a toutes les données personnelles des clients qui sont inscrites en clair.

**Gitignore :** le vendor est commenté donc il sera possible de l'envoyer à travers un push.

```

1  / .idea
2  ###> symfony/framework-bundle ###
3  /.env.local
4  /.env.local.php
5  /.env.*.local
6  /config/secrets/prod/prod.decrypt.private.php
7  /public/bundles/
8  /var/
9  #/vendor/
10 ###< symfony/framework-bundle ###
11
12 ###> phpunit/phpunit ###
13 /phpunit.xml
14 .phpunit.result.cache
15 ###< phpunit/phpunit ###
16
17 ###> symfony/phpunit-bridge ###
18 .phpunit.result.cache
19 /phpunit.xml
20 ###< symfony/phpunit-bridge ###
21

```

**Année fixe :** Declaration Year est fixé à 2023 et ne se met pas à jour en fonction de l'année, ce qui va créer un problème si on change d'année.

```
TAX_RATE=0.17
DECLARATION_YEAR=2023
OPEN_FOR_DECLARATION=1
```

Lors de la création et de la visualisation des déclarations, certaines ne sont pas cohérentes. Ici, nous pouvons voir qu'il y a écrit déclaration pour 2022 mais que les dates de déclarations et de paiement sont notées en 2023.

Déclaration pour l'année 2022	
<b>Date de déclaration</b>	22/12/2023
<b>Base de calcul déclarée</b>	65871 €
<b>Montant dû</b>	11198 €
<b>Date de paiement</b>	22/12/23 08:54
<b>Nom</b>	Arnaude-Constance Benoit
<b>Numéro carte</b>	4146592708807212
<b>Date d'expiration</b>	10/26
<b>Code</b>	153

[Retour](#)

Cela crée aussi une confusion dans le calcul des statistiques pour ce qu'il est payé en 2022 et 2023 car ils sont identiques.

ASDEBOL 1.0 Entreprises [Déconnexion \(company@mail.dev\)](#)

Statistiques				
Nb d'entreprises	Calculées 2023	Payées 2023	Calculées 2022	Payées 2022
12	94 €	75663 €	102524 €	75663 €

## Résolution des problématiques :

### Formulaire d'inscription :

- Vérifier la conformité des champs email, téléphone, SIRET, adresse, avant d'envoyer le formulaire.
- Vérifier la conformité du mot de passe selon le RGPD, il doit contenir min 12 caractères, des chiffres, des majuscules, des minuscules et des caractères spéciaux.

### Attribution de rôles à la création d'un compte :

- Vérifier que les rôles sont implémentés à chaque création de comptes.

#### **Formulaire de connexion et sécurité de l'accès :**

- Rendre la connexion obligatoire pour accéder aux pages.
- Vérifier le cloisonnement des pages pour que l'user puisse accéder seulement à ses propres pages, sans pouvoir accéder à celles des admins ni celles des autres user.
- Rajouter un système de session pour qu'au bout d'un certain temps l'utilisateur soit déconnecté, ou en cas de fermeture d'une page...

#### **Configuration de security.yaml :**

- Décommentez et configurez correctement les rôles dans le fichier security.yaml.

#### **Base de données :**

- Complexifier l'accès à la BDD, en rajoutant un système de connexion via user, ou au minimum rajouter un mot de passe.
- Il ne faut pas avoir de plaintext visible, le mot de passe doit seulement être haché.
- Cryptez les informations bancaires.
- Ne pas donner accès à toutes les données à caractère personnel sur la base.

#### **Fichier .gitignore :**

- Décommentez le fichier vendor dans .gitignore pour empêcher son envoi.

#### **Gestion de l'année fixe dans les déclarations :**

- Utilisez une fonction dynamique pour récupérer l'année en cours afin d'éviter des problèmes liés au changement d'année.