**Plan de test – Fonction login**

**1 - Identification du test**

**Nom** : Test de connexion utilisateur PSW  
**Numéro** : T3.1

**2 - Référence du module testé**

routes/psw.py – Fonction login

**3 - Objectif du test**

Valider le bon fonctionnement de la route de connexion :

* Vérifier si l'identifiant correspond à un utilisateur existant
* Vérifier que le mot de passe déchiffré correspond à celui fourni
* Retourner les bonnes réponses selon chaque situation (succès ou erreurs)

**4 - Procédure du test**

* **Initialisation** : Création d’un utilisateur dans la base avec un identifiant et un mot de passe chiffré
* **Lancement** : Appel à la route /psw/login/ avec un JSON :

{

"identifiant": "admin123",

"mot\_de\_passe": "toto123"

}

* **Observation** : Vérification du code retour HTTP et du corps de la réponse (success, id\_utilisateur, role...)

**5 - Résultats attendus**

| **N° Test** | **Condition** | **BDD Préparée** | **Résultat attendu** | **Statut attendu** |
| --- | --- | --- | --- | --- |
| 1 | Utilisateur inconnu | Aucun utilisateur avec cet identifiant | 404 Utilisateur introuvable | Erreur |
| 2 | Mauvais mot de passe | Utilisateur trouvé, mais mauvais mot de passe fourni | 401 {"success": False} | Erreur |
| 3 | Connexion réussie | Utilisateur avec identifiant et mot de passe corrects | 200 {"success": True, "id\_utilisateur": X, "role": "Admin"} | Succès |

**6 - Moyens à mettre en œuvre**

* **Logiciels** : FastAPI, Pytest, chiffrement (module interne)
* **Matériel** : Poste de développement, base de données locale
* **Préconditions** :
  + Fonction decryptPassword correctement fonctionnelle dans chiffrement
  + Données d’utilisateurs présentes dans la base avec mots de passe chiffrés
  + Route /psw/login/ correctement montée dans FastAPI