**Plan de test – Fonction dissocierBadge**

**1 - Identification du test**

**Nom** : Test de dissociation d’un badge d’un utilisateur  
**Numéro** : T12.1

**2 - Référence du module testé**

routes/pgs.py – Fonction dissocierBadge

**3 - Objectif du test**

Vérifier que la route dissocie un badge d’un utilisateur si :

* L’utilisateur et le badge existent
* Le badge est effectivement associé

Et que toutes les erreurs sont gérées :

* Utilisateur ou badge inexistant
* Badge non encore associé

**4 - Procédure du test**

* **Initialisation** : Créer ou non un utilisateur et un badge avec ou sans lien entre les deux
* **Lancement** : Requête PUT sur /pgs/dissocier/utilisateur/{id\_utilisateur}/badge/{uid\_badge}  
  avec un JSON de type :

{

"uid": "123ABC45",

"id\_utilisateur": 2

}

* **Observation** : Vérifier le code HTTP et le retour JSON ou l’erreur

**5 - Résultats attendus**

| **N° Test** | **Condition** | **BDD Préparée** | **Résultat attendu** | **Statut attendu** |
| --- | --- | --- | --- | --- |
| 1 | Utilisateur inexistant | Aucun utilisateur avec cet ID | 404 Utilisateur non trouvé | Erreur |
| 2 | Badge inexistant | Aucun badge avec cet UID | 404 Badge non trouvé | Erreur |
| 3 | Badge non attribué | Badge existe mais id\_utilisateur = None | 400 Ce badge n’est pas déjà attribué à un utilisateur | Erreur |
| 4 | Dissociation réussie | Badge et utilisateur existants, badge bien lié | 200 Badge retourné avec id\_utilisateur = null | Succès |

**6 - Moyens à mettre en œuvre**

* **Logiciels** : FastAPI, SQLAlchemy, Pytest
* **Matériel** : Poste de développement avec environnement Python
* **Préconditions** :
  + Tables Utilisateur et Badge initialisées
  + Relation Badge.id\_utilisateur bien exploitée
  + Modèle AssoRequest défini dans schemas.py