**Plan de test – Fonction activerBadge**

**1 - Identification du test**

**Nom** : Test d’activation/désactivation de badge  
**Numéro** : T13.1

**2 - Référence du module testé**

routes/pgs.py – Fonction activerBadge

**3 - Objectif du test**

Tester la mise à jour de l’état actif d’un badge :

* L’activer si désactivé
* Le désactiver si activé
* Retourner une erreur si l’état demandé est déjà celui en base
* Retourner une erreur si le badge est introuvable

**4 - Procédure du test**

* **Initialisation** : Créer un badge avec un uid et un actif défini (True ou False)
* **Lancement** : Requête PUT sur /pgs/badge/ avec :

json

CopierModifier

{

"uid": "123ABC45",

"actif": false

}

* **Observation** : Vérifier le status\_code et le detail ou le contenu retourné

**5 - Résultats attendus**

| **N° Test** | **Condition** | **BDD Préparée** | **Résultat attendu** | **Statut attendu** |
| --- | --- | --- | --- | --- |
| 1 | Badge inexistant | Aucun badge avec cet UID | 404 Badge non trouvé | Erreur |
| 2 | Badge déjà actif | Badge avec actif = True, requête actif = True | 403 Badge déjà activé | Erreur |
| 3 | Badge déjà désactivé | Badge avec actif = False, requête actif = False | 403 Badge déjà désactivé | Erreur |
| 4 | Activation OK | Badge actif = False, requête actif = True | 200 Badge avec actif = True | Succès |
| 5 | Désactivation OK | Badge actif = True, requête actif = False | 200 Badge avec actif = False | Succès |

**6 - Moyens à mettre en œuvre**

* **Logiciels** : FastAPI, SQLAlchemy, Pytest
* **Matériel** : Poste de dev avec environnement Python
* **Préconditions** :
  + Table Badge existante
  + Modèle ActiBadge défini dans schemas
  + Route /pgs/badge/ correctement intégrée à FastAPI