

# Mise à jour d'un package

---

On veut une fonctionnalité permettant de mettre à jour un colis.

## Endpoint REST

On veut un endpoint en PATCH sur `/packages/<id>`.

Le body de la requête doit être un JSON au format suivant :

```
{
  "number": "10",
  "street": "Avenue de la paix",
  "postalCode": "75001",
  "city": "Paris",
  "country": "France",
  "details": "3ème étage porte droite",
  "phoneNumber": "0685945263",
  "email": "jean.dupont@gmail.com",
  "status": "in-transit",
  "deliveryPersonId": "38a8ac6e-198c-11ee-a789-00155d0798e7"
}
```

Tous les champs sont optionnels, autrement dit le JSON représentant l'objet vide `{}` est parfaitement valide en entrée, mais ne produira aucun résultat sur le package.

Les champs `deliveryPersonId` et `details` peuvent contenir la valeur `null`, il faudra alors mettre le champ à `null` sur le colis.

Chaque champ (s'il est fourni) doit faire l'objet des mêmes contraintes de validation que pour l'endpoint de création de package (POST sur `/packages`).

Les réponses possibles sont les suivantes:

### Succès

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  ...
  ... le contenu du package mis à jour
  ...
}
```

### Package inexistant

```
HTTP/1.1 404 Not Found
Content-Type: application/json

"Le package <package-id> n'existe pas"
```

## Livreur inexistant

```
HTTP/1.1 404 Not found
Content-Type: application/json

"Le livreur <user-id> n'existe pas"
```

## Format des champs invalides

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

"Le champ \"<nom>\" est invalide"
```

## Action interdite (règles métier liées aux status)

```
HTTP/1.1 403 Forbidden
Content-Type: application/json

"Cette action n'est pas autorisée"
```

## Autres erreurs

Toutes les autres erreurs clients doivent faire l'objet d'un code 400 sans corps de réponse.

Toutes les erreurs serveur doivent faire l'objet d'un code 500 sans corps de réponse.

## Logique métier

On veut que l'application mette à jour chaque champ fourni sur le colis.

## Changement de statut

Les règles de changement de statut sont les suivantes:

- Si le package est **pending**, il peut passer dans n'importe quel autre statut.
- Si le package est **in-transit**, il ne peut passer que en **returned** ou **delivered**.
- Si le package est **delivered**, on ne peut que passer en **returned**.
- Si le package est **returned**, il est impossible de modifier le statut du colis.

## Notification de changement de statut

Lors d'un changement de statut, un email doit être envoyé à l'adresse mail indiquée dans les données du colis.

Le corps du mail doit être en texte brut (`text/plain`):

- pour le passage en `in-transit`:

```
Votre colis est en cours de livraison et sera bientôt remis à l'adresse "<adresse complète>".
```

- Pour le passage en `delivered`:

```
Votre colis a été livré à l'adresse "<adresse complète>".
```

- Pour le passage en `returned`:

```
Votre colis a été retourné à l'expéditeur.
```

## Assignation, ou changement de livreur

En cas d'assignation ou de changement de livreur, il faudra vérifier si l'utilisateur est bien existant.

Il faudra également appliquer ce comportement sur l'endpoint de création de colis sur `POST /packages`.

## Sécurité

Un administrateur peut tout faire.

Un livreur peut modifier tous les champs des colis qui lui sont assignés, sauf le champ `deliveryPersonId`. Autrement dit, il ne peut pas réassigner les colis qui lui sont assignés.

Un utilisateur non authentifié ne peut rien faire.

## BONUS: Tests d'intégration

Des tests doivent être mis en place avec `@SpringBootTest`.

Les tests devront vérifier:

- un cas de mise à jour effectuée avec succès.
- les cas d'erreur lié à un changement de statut sur un colis.