

Projet – Gestion TGV & Billets

Vous développez une petite API pour gérer des **TGV** et les **billets** associés.

Un billet appartient à **un et un seul TGV**. À la création d'un billet, l'**ID du TGV doit être validé**.

Modèles attendus

1) TGV

- `numero`: *string* (unique, ex: "TGV-8471")
- `origine`: *string* (ex: "Paris Gare de Lyon")
- `destination`: *string* (ex: "Lyon Part-Dieu")
- `heureDepart`: *Date*

Optionnel : `placesTotales` (*number*), `placesRestantes` (*number*, calculable).

2) Billet

- `tgv`: *ObjectId* → référence vers **TGV** (obligatoire)
- `passagerNom`: *string*
- `classe`: *string* ("Seconde" | "Première")
- `prix`: *number* (≥ 0)
- `place`: *string* (ex: "12A")
- `dateAchat`: *Date* (défaut: *now*)

Règle métier : **On ne peut créer un billet que si le `tgv` référencé existe.**

Tâches à réaliser (CRUD)

TGV – Routes

- **POST** `/api/tgvs` : créer un TGV
- **GET** `/api/tgvs` : lister tous les TGV (+ filtres simples `?origine=...&destination=...`)
- **GET** `/api/tgvs/:id` : récupérer un TGV
- **PUT** `/api/tgvs/:id` : modifier un TGV
- **DELETE** `/api/tgvs/:id` : supprimer un TGV
 - Interdire la suppression si des billets existent.

Billets – Routes

- **POST** `/api/billets` : créer un billet
 - **Valider** que `req.body.tgv` est un **ObjectId valide** et correspond à un TGV existant.
- **GET** `/api/billets` : lister les billets (supporter `?tgv=ID` pour filtrer par TGV)
- **GET** `/api/billets/:id` : récupérer un billet
- **PUT** `/api/billets/:id` : modifier un billet
- **DELETE** `/api/billets/:id` : supprimer un billet