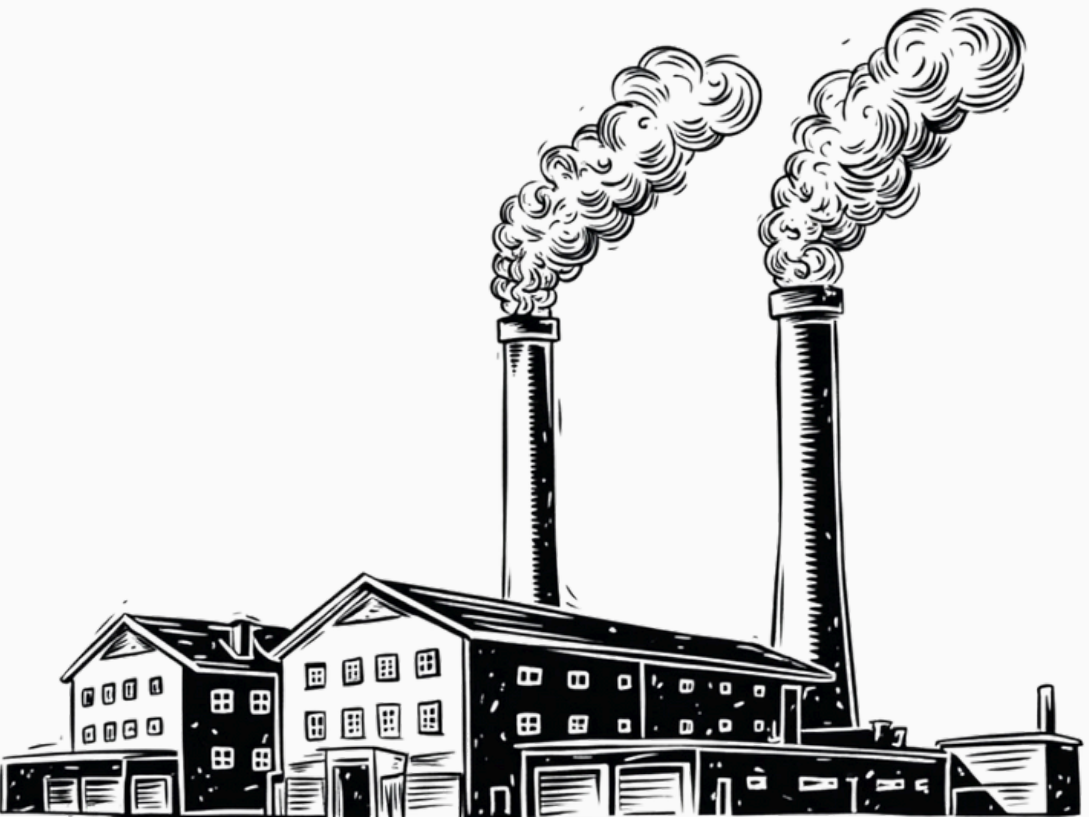
FLORIAN BÊME

FACTORY PATTERN

DÉLÈGUE LA CRÉATION À CE PATTERN.



01

C’est quoi ?

Il permet de centraliser la création d'objets dans une seule factory. C'est elle qui crée tes objets à la demande et instancie ce dont tu as besoin.

02

Le problême

Tu crées manuellement chaque objet dans ton code...

❌ Des if/else ou switch répétés partout

❌ Logique de création dupliquée

❌ Modifier un format = refaire tout le code

03

La solution : Factory

Une seule factory pour créer tous tes objets!

✅ Logique centralisée au même endroit

✅ Pas de duplication de code

✅ Modification facile = un seul lieu à changer

04

Implémentation

```
class ResponseFactory {
  // Méthodes spécifiques pour chaque type
  success(data) {
    return {
      status: 200,
      success: true,
      data
    };
  }
  /**...**/
  error(message) {
    return {
      status: 500,
      success: false,
      error: message
    };
  }
}
```

```
// SANS FACTORY ❌
app.get('/users', (req, res) => {
  const users = getUsersFromDB()
  // Tu crées la réponse manuellement à chaque fois
  res.status(200).json({
    status: 200,
    success: true,
    data: users,
  })
})

// AVEC FACTORY ✅
const responseFactory = new ResponseFactory();

app.get('/users', (req, res) => {
  const users = getUsersFromDB();
  const response = responseFactory.success(users);
  res.status(response.status).json(response);
});
```