# Intégration de Power BI dans un portail de gestion du Software Testing

Ce document explique en détail comment intégrer Power BI dans un portail web pour visualiser de manière dynamique les statistiques issues d’un backend Spring Boot utilisant une base de données H2.

## 🎯 Objectif

Intégrer un rapport Power BI dynamique dans un portail React, basé sur des données générées par une application backend Spring Boot avec une base H2.

## 🧩 Étapes d’intégration Power BI dans le portail

### 1. Générer les données dans Spring Boot

Créer une API REST ou un export CSV exposant les taux d’occupation par ressource, par exemple : /api/statistics/occupation ou /export/occupation.csv

### 2. Connecter Power BI à ton backend

Dans Power BI Desktop :  
- Aller à « Obtenir des données » > Web  
- Entrer l’URL de l’API ou du fichier CSV  
- Charger les données dynamiquement

### 3. Créer ton rapport Power BI

Créer un graphique en barres (ressource vs taux d’occupation), ajouter des KPI, enregistrer le fichier .pbix et le publier dans Power BI Service.

### 4. Publier sur Power BI Service

Depuis Power BI Desktop, publier le rapport dans ton espace de travail Power BI Service, puis copier le lien « intégrer dans une application web ou un portail ».

### 5. Intégration dans le frontend React

Insérer le rapport via iframe dans ton portail React :  
  
```  
const PowerBIDashboard = () => (  
 <div style={{ width: '100%', height: '600px' }}>  
 <iframe  
 title="Statistiques Power BI"  
 width="100%"  
 height="100%"  
 src="https://app.powerbi.com/view?r=TON\_LIEN\_PUBLIÉ"  
 frameBorder="0"  
 allowFullScreen  
 ></iframe>  
 </div>  
);  
```

### 6. Méthode sécurisée : Power BI Embedded

Utiliser Azure AD, générer un Embed Token et intégrer avec powerbi-client SDK pour une intégration sécurisée avec filtrage utilisateur.

## 🔄 Rafraîchissement des données

Les options de mise à jour dépendent de ta méthode d’intégration :

- Fichier CSV local : manuel  
- API Web (CSV ou JSON) : planifiable dans Power BI Service  
- Power BI Embedded : via SDK avec refresh automatique  
- Connexion directe à H2 : nécessite Power BI Data Gateway

### 🛠 Utiliser Power BI Gateway pour la base H2

1. Installer Power BI Gateway (Standard Mode)  
2. Configurer la source JDBC vers H2  
3. Publier le rapport Power BI  
4. Activer le rafraîchissement planifié depuis Power BI Service

### 💡 Intégrer un bouton de rafraîchissement dans React

Exemple simple :  
  
```  
const handleRefresh = () => {  
 window.location.reload();  
};  
```

## 📌 Résumé

- API backend expose les données  
- Power BI charge dynamiquement via Web  
- Dashboard intégré en iframe ou via SDK  
- Rafraîchissement planifiable ou programmé

Excellente question Hana ! Pour afficher **le taux d’occupation par ressource dans ton portail de Software Testing** de manière **dynamique et visuelle**, tu as totalement raison de vouloir utiliser une **solution BI (Business Intelligence)**. Je vais te guider **pas à pas de A à Z**, en gardant les choses **simples, claires et professionnelles**.

## 🎯 ****Objectif****

Afficher dans le Dashboard du portail un **indicateur visuel dynamique** (graphe, matrice, tableau croisé…) du **taux d’occupation des testeurs QA** ou des ressources QA, calculé en fonction des tâches de test assignées.

## 🧩 Étapes globales :

**1. Collecte & stockage des données**  
**2. Préparation des données (ETL ou Data Prep)**  
**3. Modélisation & Calcul du taux d’occupation**  
**4. Création du dashboard (BI)**  
**5. Intégration dans ton portail**

## ✅ Étape 1 : ****Collecte des données****

Tu dois d’abord **centraliser les données** liées :

* Aux **ressources QA** (nom, ID, équipe, disponibilité)
* Aux **demandes de test** (projet, date, durée estimée, testeur assigné)
* Aux **plannings** ou charges de travail (heures travaillées / heures disponibles)

🛠️ Ces données peuvent venir :

* D’une **base de données PostgreSQL / MySQL**
* D’un fichier Excel partagé (au début)
* D’un outil de gestion type **Jira** si utilisé

## ✅ Étape 2 : ****Préparation (ETL)****

Tu peux utiliser un outil simple comme **Microsoft Power BI**, **Tableau**, ou **Metabase**.

Exemple avec **Power BI** :

* Connecte-toi à ta base de données via ODBC ou PostgreSQL/MySQL connector
* Crée une **jointure** entre :
  + la table des ressources
  + la table des affectations de test
  + la table du planning ou des heures disponibles
* Nettoie les colonnes inutiles

## ✅ Étape 3 : ****Calcul du taux d’occupation****

💡 **Taux d’occupation (%) = (heures assignées aux tests / heures disponibles) × 100**

Dans Power BI :

* Crée une **mesure DAX** :

DAX

CopierModifier

TauxOccupation =

DIVIDE(

SUM(FaitsTests[Heures\_Assignées]),

SUM(Ressources[Heures\_Disponibles]),

0

)

## ✅ Étape 4 : ****Création du dashboard dynamique****

Utilise des visuels interactifs dans Power BI :

* **Matrice ou tableau croisé** : nom des QA en ligne, jours en colonnes
* **Graphique en barres ou jauge** : taux d’occupation par QA
* **Filtres** dynamiques : par semaine, mois, projet

Tu peux ajouter des **couleurs conditionnelles** :

* Vert si taux < 70%
* Orange si entre 70-90%
* Rouge si > 90% (risque de surcharge)

## ✅ Étape 5 : ****Intégration dans ton portail****

Deux options selon ton stack :

### 🟢 Si tu utilises ****Power BI**** :

* Publie ton rapport sur le **Power BI Service**
* Récupère le **lien d’embed sécurisé**
* Intègre-le dans ton portail via une **iframe** avec un **token d’accès** sécurisé :

jsx

CopierModifier

<iframe

width="100%"

height="500"

src="https://app.powerbi.com/reportEmbed?...token=xyz"

frameBorder="0"

allowFullScreen="true">

</iframe>

### 🟢 Si tu utilises ****Metabase**** :

* Même logique : publie la carte, génère le lien, intègre via iframe

## 🎁 Bonus – Alternative full React

Si tu veux tout faire en React sans outil BI :

* Tu collectes les données avec ton backend (ex : /api/test/load-by-user)
* Tu calcules les taux dans le frontend
* Tu affiches avec recharts, chart.js ou react-table

Mais pour de la **visualisation pro** & gestion future, je recommande BI externe (Power BI ou Metabase).

## 🧠 Exemple visuel

**Tableau de bord** :

| **Ressource QA** | **Lundi** | **Mardi** | **Mercredi** | **Jeudi** | **Vendredi** | **Moyenne (%)** |
| --- | --- | --- | --- | --- | --- | --- |
| Hanane | 80% | 90% | 75% | 85% | 100% | 86% |
| Zakaria | 45% | 55% | 60% | 65% | 50% | 55% |

Souhaites-tu que je t’aide à créer un **modèle Power BI** ou un exemple de **calcul back + graphique React** ?

4o

##### Vous avez dit :

oui

###### ChatGPT a dit :

Parfait ! Voici deux options détaillées pour afficher dynamiquement le **taux d’occupation des ressources QA** dans ton portail de Software Testing. Je vais te proposer :

1. 🔵 Une solution **Power BI** (avec intégration dans React)
2. 🟢 Une solution **Full React + Backend Spring Boot** (si tu veux tout faire sans outil externe)

## 🔵 ****Option 1 : Power BI + Intégration React (Recommandé pour Dashboard Pro)****

### ⚙️ 1. Préparer les données

Tu dois disposer d’une **base de données centralisée** avec ces tables :

* resources :
  + id, name, total\_available\_hours\_per\_day
* test\_assignments :
  + id, resource\_id, date, task\_name, estimated\_hours

🗄️ Exemple :

sql

CopierModifier

SELECT

r.name,

t.date,

SUM(t.estimated\_hours) AS total\_assigned,

r.total\_available\_hours\_per\_day

FROM resources r

JOIN test\_assignments t ON r.id = t.resource\_id

GROUP BY r.name, t.date, r.total\_available\_hours\_per\_day

### 📊 2. Importer les données dans Power BI

* Ouvre Power BI Desktop
* Choisis ta base PostgreSQL / MySQL
* Applique la requête ci-dessus
* Crée une **mesure DAX** :

dax

CopierModifier

TauxOccupation =

DIVIDE(

SUM(test\_assignments[estimated\_hours]),

MAX(resources[total\_available\_hours\_per\_day]),

0

) \* 100

### 📈 3. Créer un visuel dynamique

* Utilise une **matrice** ou un **graphique heatmap**
* Ajoute un **filtre temporel** (semaine, mois)
* Applique un **format conditionnel de couleur**

### 🌐 4. Publier sur Power BI Service

* Clique sur “Publier”
* Sur le Power BI Service, choisis “Fichier > Intégrer > Site web sécurisé”

Tu obtiendras un lien d'intégration (iframe).

### 🧩 5. Intégration dans React

Dans ton composant Dashboard.jsx :

jsx

CopierModifier

const Dashboard = () => {

return (

<div className="p-4">

<h2 className="text-xl font-bold mb-4">Taux d’occupation des ressources QA</h2>

<iframe

title="Power BI QA Dashboard"

width="100%"

height="600px"

src="https://app.powerbi.com/reportEmbed?...token=xyz"

frameBorder="0"

allowFullScreen

></iframe>

</div>

);

};

export default Dashboard;

## 🟢 ****Option 2 : React + Spring Boot (Full Custom)****

### 1. Backend (Spring Boot)

Crée un endpoint /api/qa/occupation :

java

CopierModifier

@GetMapping("/occupation")

public List<OccupationDTO> getOccupationByDay() {

return testAssignmentService.getDailyOccupation();

}

**DTO Example**:

java

CopierModifier

public class OccupationDTO {

private String resourceName;

private String date;

private double occupationRate;

}

Service :

java

CopierModifier

// Pour chaque ressource et chaque jour, calculer: (total\_assigned\_hours / total\_available\_hours) \* 100

### 2. Frontend (React)

Utilise une lib comme recharts :

bash

CopierModifier

npm install recharts

Composant OccupationChart.jsx :

jsx

CopierModifier

import { BarChart, Bar, XAxis, YAxis, Tooltip, Legend, ResponsiveContainer } from 'recharts';

import { useEffect, useState } from 'react';

import axios from 'axios';

const OccupationChart = () => {

const [data, setData] = useState([]);

useEffect(() => {

axios.get("/api/qa/occupation").then(res => setData(res.data));

}, []);

return (

<div className="p-4">

<h2 className="text-xl font-bold mb-4">Taux d’occupation des QA</h2>

<ResponsiveContainer width="100%" height={400}>

<BarChart data={data} margin={{ top: 20, right: 20, left: 0, bottom: 5 }}>

<XAxis dataKey="resourceName" />

<YAxis />

<Tooltip />

<Legend />

<Bar dataKey="occupationRate" fill="#8884d8" />

</BarChart>

</ResponsiveContainer>

</div>

);

};

export default OccupationChart;

## ✅ Résultat attendu :

Un tableau de bord dynamique affichant :

* Les QA sur l’axe des X
* Le **taux d’occupation (%)** sur l’axe des Y
* Des filtres pour la période
* Des couleurs conditionnelles ou des alertes si surcharge

## 🧠 Recommandation

| **Tu veux...** | **Choix recommandé** |
| --- | --- |
| Solution simple, rapide et visuelle | ✅ Power BI / Metabase |
| Plus de contrôle, moins d’outils externes | ✅ React + Spring |
| Facilité pour le business / RH / audit | ✅ Power BI |

Souhaites-tu que je te prépare :

* Un exemple Power BI (fichier .pbix) ?
* Le code backend complet avec Spring ?
* Ou les deux dans un repo GitHub ?

4o

##### Vous avez dit :

je travaille juste avec H2, donner moi une solution pour la base e données excel par exemple

###### ChatGPT a dit :

Très bien ! Si tu travailles avec une **base de données H2** **et** que tu veux exploiter un **fichier Excel** (ou générer dynamiquement des taux d’occupation pour affichage), voici une **solution complète et professionnelle**, simple à mettre en œuvre **de A à Z** :

## ✅ Objectif :

Afficher dynamiquement dans ton portail :

📊 **Taux d’occupation des ressources QA**, basé sur des données (ou exports) Excel.

## 🟡 Étapes de la solution :

### 🧩 1. Préparer ton fichier Excel .xlsx

Crée un fichier comme ceci :

| **Ressource** | **Date** | **Heures Assignées** | **Heures Disponibles** |
| --- | --- | --- | --- |
| Hana | 2025-05-24 | 5 | 8 |
| Sara | 2025-05-24 | 6 | 8 |

Nom du fichier : occupation.xlsx

### ⚙️ 2. Lire le fichier Excel dans Spring Boot (avec Apache POI)

#### 📦 Ajoute la dépendance dans pom.xml :

xml

CopierModifier

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>5.2.5</version>

</dependency>

#### 🔧 Crée un service pour lire le fichier :

java

CopierModifier

@Service

public class ExcelOccupationService {

public List<OccupationDTO> readOccupationFromExcel() throws IOException {

List<OccupationDTO> occupations = new ArrayList<>();

FileInputStream file = new FileInputStream("src/main/resources/occupation.xlsx");

Workbook workbook = new XSSFWorkbook(file);

Sheet sheet = workbook.getSheetAt(0);

for (Row row : sheet) {

if (row.getRowNum() == 0) continue; // Skip header

String name = row.getCell(0).getStringCellValue();

String date = row.getCell(1).getStringCellValue();

double assigned = row.getCell(2).getNumericCellValue();

double available = row.getCell(3).getNumericCellValue();

double rate = (available == 0) ? 0 : (assigned / available) \* 100;

occupations.add(new OccupationDTO(name, date, rate));

}

workbook.close();

file.close();

return occupations;

}

}

#### 🎯 DTO :

java

CopierModifier

public class OccupationDTO {

private String resourceName;

private String date;

private double occupationRate;

// Constructeur, Getters, Setters

public OccupationDTO(String resourceName, String date, double occupationRate) {

this.resourceName = resourceName;

this.date = date;

this.occupationRate = occupationRate;

}

// Getters et Setters

}

#### 📡 Endpoint REST :

java

CopierModifier

@RestController

@RequestMapping("/api/occupation")

public class OccupationController {

@Autowired

private ExcelOccupationService excelService;

@GetMapping

public List<OccupationDTO> getOccupationRates() throws IOException {

return excelService.readOccupationFromExcel();

}

}

### 🖥️ 3. Frontend React – Affichage avec Recharts

#### Installe recharts :

bash

CopierModifier

npm install recharts

#### Code OccupationChart.jsx :

jsx

CopierModifier

import { useEffect, useState } from "react";

import axios from "axios";

import {

BarChart, Bar, XAxis, YAxis, Tooltip, Legend, ResponsiveContainer

} from "recharts";

const OccupationChart = () => {

const [data, setData] = useState([]);

useEffect(() => {

axios.get("/api/occupation").then(res => setData(res.data));

}, []);

return (

<div className="p-4">

<h2 className="text-xl font-bold mb-4">Taux d’occupation des QA</h2>

<ResponsiveContainer width="100%" height={400}>

<BarChart data={data}>

<XAxis dataKey="resourceName" />

<YAxis />

<Tooltip />

<Legend />

<Bar dataKey="occupationRate" fill="#82ca9d" name="Taux (%)" />

</BarChart>

</ResponsiveContainer>

</div>

);

};

export default OccupationChart;

## ✅ Résultat final

* 🟢 **Tu modifies le fichier Excel** → Pas besoin de changer la base H2.
* 🔄 Le portail recharge dynamiquement les **taux calculés**.
* 📊 Tu as un joli graphique pro dans React.
* 🚀 Aucun outil externe requis (comme Power BI).

### 🔽 Tu veux aller plus loin ?

* Ajouter l’import Excel côté admin (upload du fichier via le frontend).
* Sauvegarder les taux dans H2 pour historique.
* Ajouter des filtres par date ou ressource dans l’interface.

Souhaite-tu que je t’aide aussi à générer ce fichier Excel automatiquement ?

4o

##### Vous avez dit :

mais je veux travailler avec Power BI et je veux que ca soit dynamique c'est a dire a partir des données de portail , je peux utiliser H2?

###### ChatGPT a dit :

Travailler avec **Power BI** de façon **dynamique à partir des données de ton portail** est tout à fait possible, **même avec une base H2**, mais avec certaines limitations. Voici une explication claire, et une **solution professionnelle complète**, adaptée à ton projet PFE :

## 🎯 Objectif :

**Afficher dynamiquement dans Power BI** le **taux d’occupation des ressources QA**, basé sur les **données du portail (stockées en H2)**.

## 🧩 Problème avec H2 et Power BI :

### 🔴 Par défaut, ****Power BI ne supporte pas H2 directement****.

H2 est une base embarquée (in-memory ou file-based), non conçue pour l’analyse BI.

## ✅ Solution professionnelle :

### 💡 Utiliser un ****export automatique en CSV ou Excel**** depuis Spring Boot → exploité dynamiquement dans ****Power BI****.

### 🟢 Étapes simples à mettre en œuvre :

### ✅ 1. Ajouter une API d’export depuis Spring Boot

Tu peux exposer un endpoint qui génère **dynamically** un fichier .csv ou .xlsx contenant les taux d’occupation.

#### Exemple : /export/occupation

java

CopierModifier

@GetMapping(value = "/export/occupation", produces = "text/csv")

public void exportOccupation(HttpServletResponse response) throws IOException {

response.setContentType("text/csv");

response.setHeader("Content-Disposition", "attachment; filename=\"occupation.csv\"");

PrintWriter writer = response.getWriter();

writer.println("Ressource,Date,Heures Assignées,Heures Disponibles,Taux (%)");

for (Occupation occ : occupationRepository.findAll()) {

double taux = (occ.getAvailableHours() == 0) ? 0 :

(double) occ.getAssignedHours() / occ.getAvailableHours() \* 100;

writer.printf("%s,%s,%d,%d,%.2f%n",

occ.getName(), occ.getDate(), occ.getAssignedHours(), occ.getAvailableHours(), taux);

}

writer.flush();

writer.close();

}

### ✅ 2. Dans Power BI :

#### ➤ Choisir ****Get Data**** > ****From Web****

Et mettre :

bash

CopierModifier

http://localhost:8080/export/occupation

Power BI va automatiquement charger le fichier CSV depuis ton portail à chaque actualisation.

✅ Tu peux aussi planifier le **refresh automatique** si tu déploies l’API sur un serveur (ex : Azure, Render, etc.).

### ✅ 3. Construire le Dashboard :

Dans Power BI :

* Ajouter des **cartes KPI**, **graphes**, **filtres par ressources**, etc.
* Exemple : Graphique barre → Taux d’occupation par ressource.

## ✨ Résultat :

* Données générées dynamiquement depuis le portail.
* Intégration simple sans changer la base H2.
* Power BI se connecte automatiquement via le lien.
* Tu peux faire des visualisations professionnelles pour ton PFE.
* Bonus : tu montres ta capacité à **intégrer BI + dev backend** 👏

## ❓Tu veux aller plus loin ?

Voici d'autres options :

* 🔁 Rafraîchissement automatique avec Power BI Service (nécessite déploiement public ou passerelle).
* ⬆️ Changer H2 vers PostgreSQL ou MySQL pour liaison directe.
* 📊 Ajouter d’autres métriques : nombre de cas de tests, taux de bugs, temps moyen de traitement, etc.

Souhaites-tu que je t’aide à créer le modèle Power BI avec les visuels et KPI adaptés ?

4o

##### Vous avez dit :

et comment integrer ca coté frontend

###### ChatGPT a dit :

Parfait ! Maintenant que tu génères dynamiquement les données dans ton backend Spring Boot pour Power BI, voici **comment intégrer cela côté frontend** dans ton **portail de gestion du Software Testing** :

## 🎯 Objectif :

**Afficher dynamiquement dans le frontend (React par exemple)** le **rapport Power BI** (ex. taux d’occupation par ressource) **généré à partir des données du backend**.

### 🛠️ Étapes pour intégrer Power BI côté frontend :

### ✅ 1. Publier ton rapport Power BI (si besoin)

Si tu veux intégrer **directement un visuel Power BI dans ton portail**, tu dois :

* Publier ton rapport sur Power BI Service
* Obtenir un **lien d’intégration publique ou sécurisé**

**⚠️ Si tu veux éviter l’authentification**, tu peux utiliser l’option :  
**File > Publish to web** (public embed – attention aux données sensibles).

### ✅ 2. Intégrer le rapport dans ton frontend (React par exemple)

Tu peux utiliser une **balise iframe** ou une **intégration via SDK Power BI JavaScript**.

#### 🅰️ Méthode simple : intégration via iframe

jsx

CopierModifier

<div style={{ height: '600px', width: '100%' }}>

<iframe

title="Dashboard Power BI"

width="100%"

height="100%"

src="https://app.powerbi.com/view?r=TON\_LIEN\_PUBLIC"

frameBorder="0"

allowFullScreen="true"

></iframe>

</div>

🔹 **TON\_LIEN\_PUBLIC** : obtenu via Power BI Service > Fichier > Publier sur le web > Obtenir le lien.

#### 🅱️ Méthode avancée : SDK Power BI JavaScript (authentifié + interactif)

Si tu veux plus de contrôle (filtrage, events, etc.) :

1. Installe le SDK :

bash

CopierModifier

npm install powerbi-client

1. Intègre ton composant Power BI :

jsx

CopierModifier

import { models, Embed, service } from "powerbi-client";

import { PowerBIEmbed } from "powerbi-client-react";

function DashboardEmbed() {

return (

<PowerBIEmbed

embedConfig={{

type: 'report', // Or 'dashboard'

id: 'REPORT\_ID',

embedUrl: 'https://app.powerbi.com/reportEmbed?reportId=xxx&groupId=xxx',

accessToken: 'TON\_TOKEN',

tokenType: models.TokenType.Embed,

settings: {

panes: {

filters: { visible: false },

pageNavigation: { visible: true }

}

}

}}

cssClassName={"report-style-class"}

getEmbeddedComponent={(embeddedReport) => {

console.log('Report loaded');

}}

/>

);

}

⚠️ Cette méthode nécessite :

* Avoir une **licence Power BI Pro**
* Générer un **Embed Token** (avec Azure AD ou Power BI Embedded)

### ✅ 3. Bonus : Bouton de téléchargement (CSV)

Tu peux aussi ajouter un bouton dans le frontend qui **télécharge les données CSV** depuis ton backend.

jsx

CopierModifier

<a href="http://localhost:8080/export/occupation" download>

<button>Télécharger les données</button>

</a>

## ✅ Résultat final :

* Tu as une **interface frontend riche** qui :
  + Affiche dynamiquement un **dashboard Power BI**
  + Donne une vue synthétique et professionnelle des **taux d’occupation**
  + Propose une **exportation des données**

## 🧠 Exemple d'UX :

* Menu : "Dashboard"
* Sous-menu : "Taux d’occupation des ressources"
* Visuel Power BI embarqué + bouton "Exporter CSV"
* Option filtre par projet, période, ressource (via Power BI ou ton API)

Souhaites-tu que je t'aide à générer un exemple de composant React prêt à coller dans ton portail ?

4o

##### Vous avez dit :

je calcule le taux d'occupation comme ca : si chaqueressource a plus de 3 demandes son taux est de 100%

###### ChatGPT a dit :

Parfait ! Tu veux calculer le **taux d’occupation d’une ressource** en fonction du **nombre de demandes qui lui sont assignées**, avec la règle suivante :

### ✅ Règle métier :

**Si une ressource a plus de 3 demandes**, son **taux d’occupation = 100%**.  
Sinon :  
**Taux = (nb\_demandes / 3) \* 100**

### 🎯 Objectif :

Afficher ce taux dynamiquement :

* Dans le **backend Spring Boot**
* Et l’exporter vers **Power BI**
* Et/ou l’afficher dans le **frontend**

## 🧩 Étapes d’implémentation

### 🟦 1. Calcul côté Backend (Spring Boot)

Tu peux créer un DTO avec calcul direct du taux :

#### ✅ Exemple OccupationDTO.java

java

CopierModifier

public class OccupationDTO {

private String ressource;

private long nbDemandes;

private double tauxOccupation;

public OccupationDTO(String ressource, long nbDemandes) {

this.ressource = ressource;

this.nbDemandes = nbDemandes;

this.tauxOccupation = Math.min(100, (nbDemandes / 3.0) \* 100);

}

// getters

}

#### ✅ Exemple dans ton service :

java

CopierModifier

public List<OccupationDTO> getOccupationParRessource() {

List<Object[]> results = demandeRepository.countDemandesParRessource();

return results.stream()

.map(obj -> new OccupationDTO((String) obj[0], (Long) obj[1]))

.collect(Collectors.toList());

}

#### ✅ Dans le Repository :

java

CopierModifier

@Query("SELECT d.ressource.nom, COUNT(d) FROM DemandeTest d GROUP BY d.ressource.nom")

List<Object[]> countDemandesParRessource();

### 🟩 2. Export en CSV pour Power BI :

java

CopierModifier

@GetMapping("/export/occupation")

public void exportOccupation(HttpServletResponse response) throws IOException {

response.setContentType("text/csv");

response.setHeader("Content-Disposition", "attachment; filename=\"occupation.csv\"");

PrintWriter writer = response.getWriter();

writer.println("Ressource,Nombre de Demandes,Taux d'Occupation");

List<OccupationDTO> stats = service.getOccupationParRessource();

for (OccupationDTO dto : stats) {

writer.printf("%s,%d,%.2f%%\n", dto.getRessource(), dto.getNbDemandes(), dto.getTauxOccupation());

}

writer.close();

}

### 🟨 3. Côté Frontend (React)

Tu peux afficher le tableau avec les taux :

jsx

CopierModifier

const data = [

{ ressource: 'Hana', demandes: 5, taux: '100%' },

{ ressource: 'Yassine', demandes: 2, taux: '66.67%' },

];

return (

<table>

<thead>

<tr>

<th>Ressource</th>

<th>Demandes</th>

<th>Taux d’occupation</th>

</tr>

</thead>

<tbody>

{data.map((row, i) => (

<tr key={i}>

<td>{row.ressource}</td>

<td>{row.demandes}</td>

<td>{row.taux}</td>

</tr>

))}

</tbody>

</table>

);

### ✅ BONUS : Visualisation dans Power BI

Une fois le CSV généré, Power BI affichera automatiquement un **bar chart du taux d’occupation** par ressource, avec :

* Axe X : Ressources
* Axe Y : Taux (%)
* Filtre par projet ou semaine possible

Souhaites-tu que je t’aide à créer aussi un visuel Power BI prêt avec les formules ou un template .pbix ?

4o

##### Vous avez dit :

je veux integre les statistiques de power bi dans mon portail

###### ChatGPT a dit :

Parfait ! Intégrer les **statistiques Power BI dans ton portail de gestion du Software Testing** est une excellente idée pour donner une **vue dynamique et professionnelle** sur les données (ex. taux d’occupation par ressource). Voici **comment tout faire étape par étape**, de **la génération des stats** jusqu’à **l’intégration dans ton frontend React**.

## 🎯 Objectif final :

👉 Intégrer un **rapport Power BI dynamique** dans ton portail web (ex : React), basé sur les données H2/Spring Boot (statistiques sur le taux d’occupation).

## 🧩 Étapes d’intégration Power BI dans ton portail

### ✅ 1. Générer les données dynamiques dans Spring Boot

Comme tu l’as déjà fait :  
Crée une API REST (/api/statistics/occupation) ou un **endpoint CSV téléchargeable** avec les taux d’occupation par ressource.

### ✅ 2. Connecter Power BI à ton backend

#### Cas 1 : Ton backend expose un fichier CSV

* Exemple : http://localhost:8080/export/occupation
* Ouvre **Power BI Desktop**
* Clique sur **"Obtenir des données" > Web**
* Entre l’URL de ton backend
* Power BI chargera automatiquement les données et tu pourras créer ton rapport

🔁 Tu peux **actualiser les données dynamiquement** en cliquant sur "Actualiser" dans Power BI Desktop ou Service.

### ✅ 3. Créer ton rapport Power BI

Crée un rapport avec :

* Un **graphique en barres** : Ressource vs Taux d’occupation
* D'autres KPIs (moyenne, max, min si utile)

Enregistre-le localement (.pbix) puis publie-le dans le **Power BI Service** (cloud) pour pouvoir l’intégrer.

### ✅ 4. Publier ton rapport sur Power BI Service

1. Clique sur **Publier** dans Power BI Desktop
2. Choisis ton espace de travail
3. Une fois publié, va sur https://app.powerbi.com
4. Ouvre le rapport > Fichier > **Publier sur le Web** (pour un lien public)  
   ou utilise **"Intégrer" > "Dans une application web ou un portail"**

🛡️ Attention : "Publier sur le web" est **public**, donc ne mets pas de données sensibles. Sinon, utilise **Power BI Embedded** pour plus de sécurité.

### ✅ 5. Intégrer le rapport dans ton portail React

#### Méthode simple : via iframe

Voici un exemple de composant React :

jsx

CopierModifier

const PowerBIDashboard = () => {

return (

<div style={{ width: '100%', height: '600px' }}>

<iframe

title="Statistiques Power BI"

width="100%"

height="100%"

src="https://app.powerbi.com/view?r=TON\_LIEN\_PUBLIÉ"

frameBorder="0"

allowFullScreen={true}

></iframe>

</div>

);

};

export default PowerBIDashboard;

🟨 Remplace TON\_LIEN\_PUBLIÉ par le lien copié depuis Power BI Service.

### ✅ 6. (Optionnel) Méthode sécurisée avec Power BI Embedded

Si tu veux que seuls les utilisateurs connectés au portail voient le rapport :

* Crée un **app registration** dans **Azure Active Directory**
* Génère un **Embed Token**
* Utilise le **SDK powerbi-client** pour intégrer le rapport via un composant React

Tu pourras gérer :

* Authentification
* Filtres dynamiques
* Sécurité par utilisateur

Si tu veux cette méthode, je peux te guider dans la configuration Azure et le code React + backend.

## 🎁 Résultat visuel dans ton portail :

* Tu accèdes à la page "Statistiques"
* Tu vois une belle **visualisation Power BI** intégrée dans ton portail
* Les données sont **dynamiques**, issues du backend Spring Boot (H2 → CSV/API → Power BI)

Souhaites-tu que je t’aide maintenant à :

* Créer un visuel .pbix de base ?
* Ou générer une vraie intégration sécurisée avec token ?

Je peux aussi te générer le composant React complet avec des options interactives.

4o

##### Vous avez dit :

et ca rafraichit au cas de changement de données

###### ChatGPT a dit :

Excellente question ! Oui, **le rafraîchissement des données** dans Power BI peut être **automatique ou manuel**, **selon la méthode d'intégration** que tu choisis. Voici comment ça fonctionne, et comment tu peux faire en sorte que tes statistiques soient **toujours à jour dans ton portail** :

## 🔄 Rafraîchissement des données – Les options selon ta configuration

### 🎯 1. ****Tu utilises un fichier CSV ou une API exposée par ton backend (Spring Boot)****

#### ✅ ****Dans Power BI Desktop**** (local) :

* Tu dois **cliquer sur "Actualiser" manuellement** pour charger les nouvelles données depuis ton backend.

#### ✅ ****Dans Power BI Service (cloud)**** :

* Tu peux programmer un **rafraîchissement automatique** :
  + Va dans ton **espace de travail**
  + Ouvre **l'ensemble de données**
  + Clique sur **"Planifier le rafraîchissement"**
  + Choisis la fréquence (ex : toutes les heures)

⚠️ Mais pour que cela fonctionne :

* Ton backend (ou API CSV) doit être **accessible publiquement ou via une passerelle**
* Si tu es en local (localhost ou H2 local), tu devras configurer une **Gateway Data Source** (voir plus bas)

### 🧰 Comment faire pour que Power BI accède à tes données H2 ?

#### 🔵 Option 1 : ****Exposer un export CSV via ton backend****

* Exemple : http://monbackend/api/export/statistiques.csv
* Dans Power BI Desktop, tu charges ce fichier web
* Dans Power BI Service, tu programmes un rafraîchissement

✅ Simple, pas besoin de connexion directe à H2

### 🟠 Option 2 : Utiliser une ****passerelle Power BI (Data Gateway)****

Power BI peut accéder à ta base H2 via JDBC **si tu configures une passerelle** sur ton poste local ou sur un serveur.

Étapes :

1. Installe [Power BI Gateway (Standard Mode)](https://powerbi.microsoft.com/fr-fr/gateway/)
2. Configure la source de données (connexion JDBC à H2)
3. Publie ton rapport .pbix
4. Dans le Power BI Service, configure le **rafraîchissement planifié**

## 🚀 Résumé – Comportement du rafraîchissement

| **Intégration** | **Rafraîchissement automatique possible ?** | **Comment ?** |
| --- | --- | --- |
| 📁 CSV statique | ❌ Non | Tu dois republier le fichier |
| 🌐 API CSV/API Web | ✅ Oui | En planifiant dans Power BI Service |
| 🧩 Power BI Embedded + API | ✅ Oui | Via actualisation côté backend |
| 🖥️ Connexion H2 (via Gateway) | ✅ Oui | Nécessite Data Gateway installée |

## 💡 En plus : Intégrer un ****bouton "Rafraîchir"**** dans ton portail React

jsx

CopierModifier

const handleRefresh = () => {

window.location.reload(); // Recharge l'iframe

};

Ou utiliser Power BI Embedded SDK pour déclencher un report.refresh() si tu utilises l'intégration sécurisée.

Souhaites-tu :

* Un exemple de publication CSV + planification ?
* Ou un guide express pour la Data Gateway avec H2 ?
* Ou un exemple React + Power BI Embedded sécurisé avec token + bouton Refresh dynamique ?

Je peux t’envoyer exactement ce qu’il faut selon ton choix.

4o

##### Vous avez dit :

je veux tout ca en detail dans un word

###### ChatGPT a dit :

Voici le document Word détaillant l'intégration de Power BI dans ton portail :

📄 Télécharger le document Integration\_PowerBI\_Portail\_Software\_Testing.docx

Il inclut toutes les étapes : génération des données, création du rapport Power BI, publication, intégration dans React, et gestion du rafraîchissement. Si tu veux que je l’adapte à un exemple précis ou l’enrichisse avec des captures, fais-moi signe !

4o