



# Club Power BI

Mardi 18 février chez AZEO – Nantes

« Développement de custom visuals  
et REX SNCF Gares & Connexions »



/Club-Power-BI



@ClubPowerBI



/ClubPowerBI



/ClubPowerBI



@ClubPowerBI

# Notre hôte : AZEO Nantes





**Power BI  
Factory**

**Contactez-nous**



# Les stats du Club Power BI

Janvier 2020

**2,5** années d'existence

**2150+** inscrits sur le groupe Meetup du Club Power BI

**55** meetups (Paris, Lille, Aix-en-Provence, Nantes, Lyon, Strasbourg)

**940+** discussions sur le forum (en français...ou presque)

**30000** vues sur Youtube

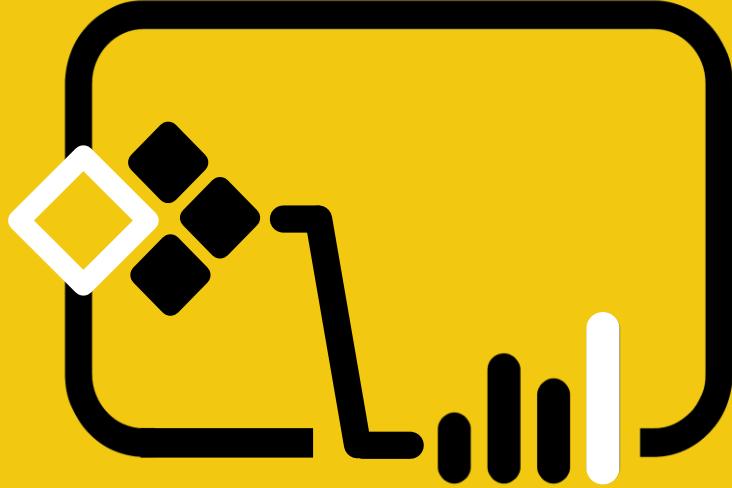
**1300+** abonnés sur Youtube

**960+** followers sur Twitter

**500+** likes sur la page Facebook

**1800+** emails référencés dans la newsletter

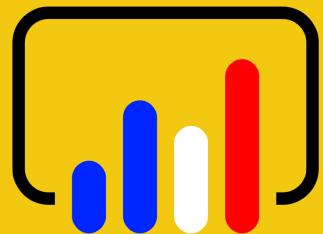




# Power Saturday

14 et 15 juin 2019, Paris

<http://powersaturday.com/>



Une conférence des communautés Club Power BI

ANNONCE  
ClubPowerBI



# Club Power BI

Sud Est

## Power BI DataFlows Comment (bien) les utiliser

Marseille | Jeudi 05 mars 2020

meetup hébergé par



SAISON 3  
2019/2020



Melissa Coates  
@SQLChick

Just announced by @GuyInACube at #PowerPlatformWT. The Feb release of #PowerBI Desktop will drop today (maybe tomorrow) & will include incremental refresh for Pro users. That got a big round of applause. 🙌 🙌

[Traduire le Tweet](#)

3:39 PM · 11 févr. 2020 · Twitter for iPhone

48 Retweets 123 J'aime



Microsoft Power BI @MSPowerBI · 14h

En réponse à @AndrewTegala

Hi there. Andrew! Although there's no info about the release date yet, we recommend staying tuned for the latest **updates** here: [msft.it/6014TwLJ8](https://msft.it/6014TwLJ8). Let us know if you have further questions.



What's new in the latest Power BI Desktop update - P...

This article describes in detail the latest monthly update for Power BI Desktop.

[docs.microsoft.com](https://docs.microsoft.com)



1



@ClubPowerBI

# **Les dernières nouveautés sur Power BI**

Mohamed Chelly

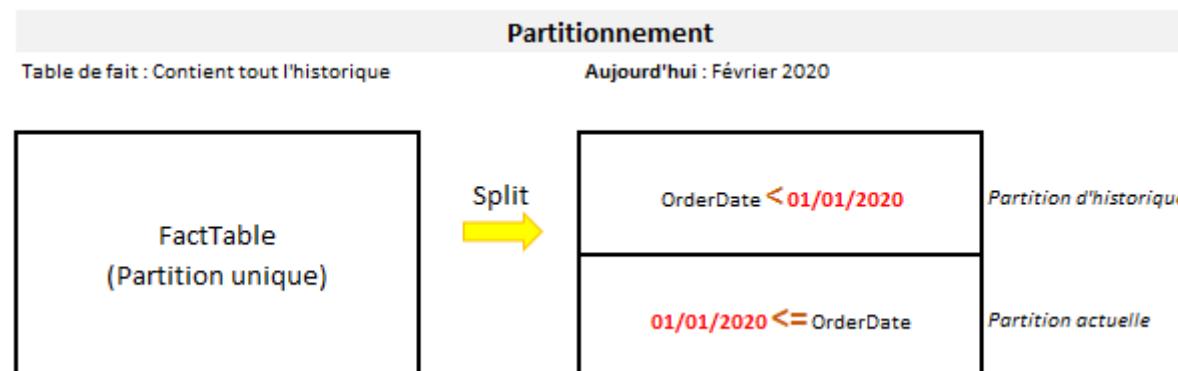
# Actualisation incrémentielle de données

## ❖ Ne pas charger l'intégralité des données

- Gérer les données sur une période glissante : Par exemple maintenir que les données des 5 dernières années.
- Actualiser (Annule & Remplace) les données sur les N dernières périodes.
- Actualiser (Annule & Remplace) uniquement les données qui ont changé depuis la dernière actualisation (la veille, ...)

## ❖ Disponible uniquement en Power BI Premium (18/02/2020)

- Bientôt disponible en Power BI Pro et Power BI free – sera en GA en 05/2020
- Utilise les techniques de partitionnement d'Analysis Services tabulaire

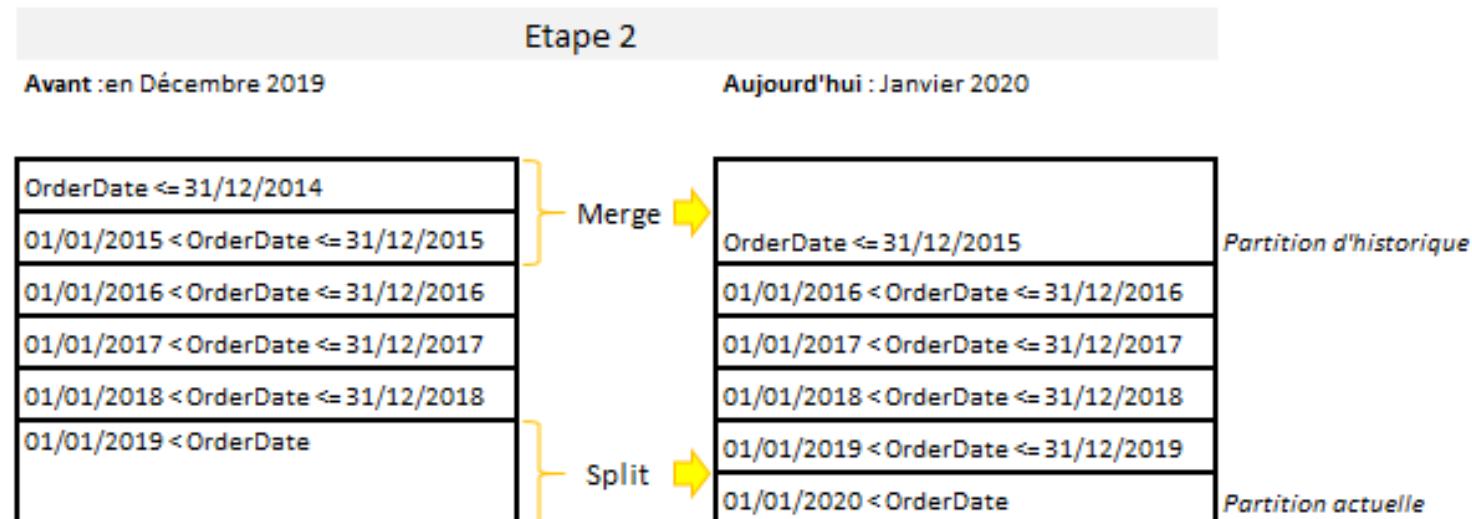


# Actualisation incrémentielle de données

## Principe de partitionnement



- ⇒ **Deux traitements à mettre en place :**
- Maintenir les partitions selon les règles de gestion souhaitées
  - Mettre à jour les données (process) en identifiant les partitions à mettre à jour



# Actualisation incrémentielle de données

## ⇒ Activation de l'actualisation incrémentielle dans Power BI Desktop :

<https://docs.microsoft.com/fr-fr/power-bi/service-premium-incremental-refresh>

- L'actualisation incrémentielle est encore en préversion, penser donc à l'activer.
- Créer les paramètres **RangeStart** et **RangeEnd** (noms réservés). Ces paramètres servent à gérer dynamiquement les bornes des partitions dans **Power Query**. Les valeurs que prendront ces paramètres sont attribuées par **Power BI Service**.
- Initialiser l'utilisation des ces paramètres dans la requête de la table concernée par la mise à jour incrémentielle. **Attention au Query Folding!**
- Activer et configurer la mise à jour incrémentielle sur la table concernée
  - Rien ne va se passer dans Power BI Desktop. La mise à jour incrémentielle sera appliquée après publication dans Power BI Service (Workspaces Premium)

# Actualisation incrémentielle de données

⇒ Configuration de mise à jour incrémentielle sur une table :

<https://docs.microsoft.com/fr-fr/power-bi/service-premium-incremental-refresh>

**Incremental refresh**

You can improve the speed of refresh for large tables by using incremental refresh in Premium workspaces. This setting will apply once you've published a report to the Power BI service.

Once you've deployed this table to the Power BI service, you won't be able to download it back to Power BI Desktop. [Learn more](#)

| Table               | Incremental refresh                    |
|---------------------|--|
| FactInternetSales ▾ | <input checked="" type="checkbox"/> On |

Store rows where column "OrderDate" is in the last:

|   |       |   |
|---|-------|---|
| 5 | Years | ▼ |
|---|-------|---|

Refresh rows where column "OrderDate" is in the last:

|   |      |   |
|---|------|---|
| 5 | Days | ▼ |
|---|------|---|

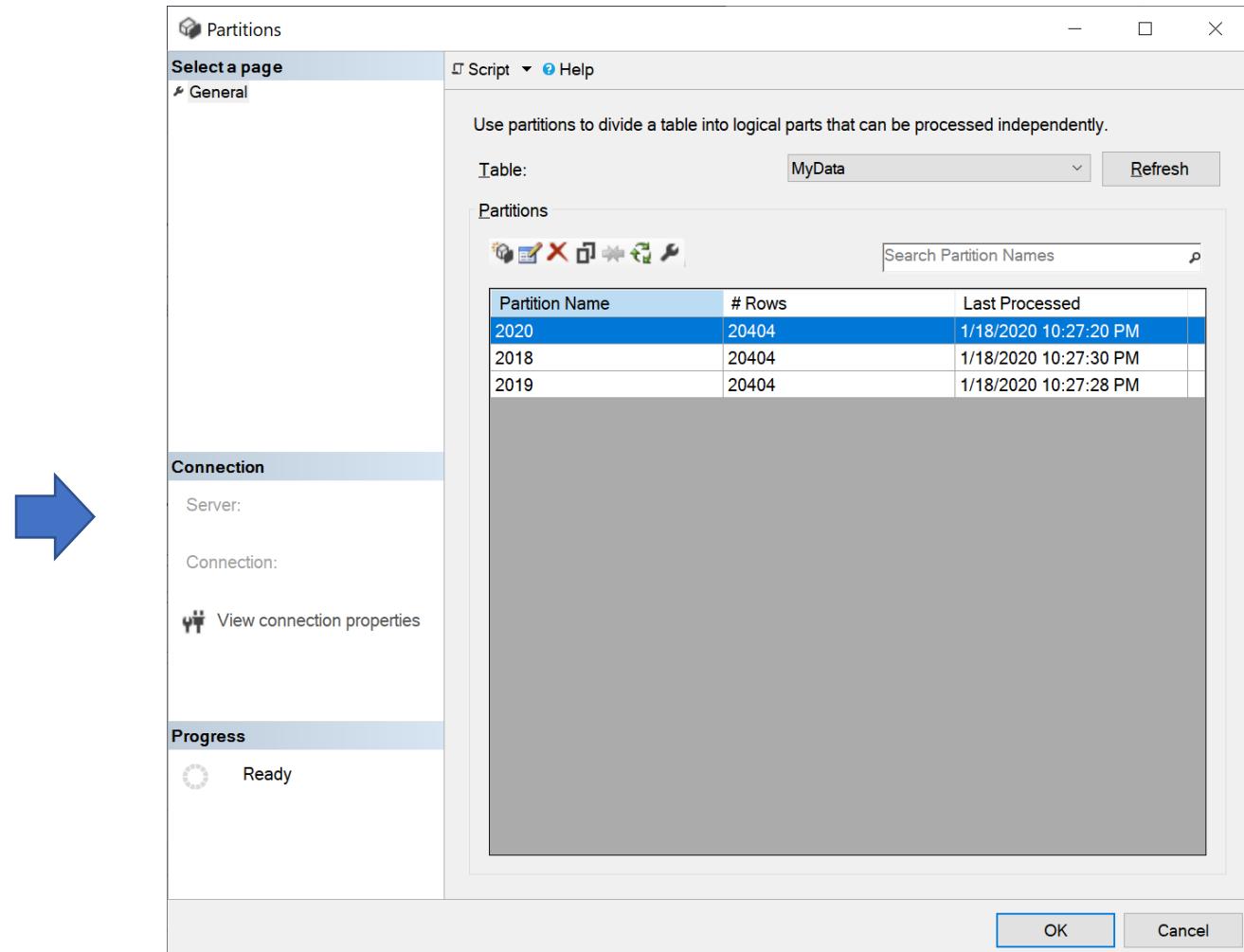
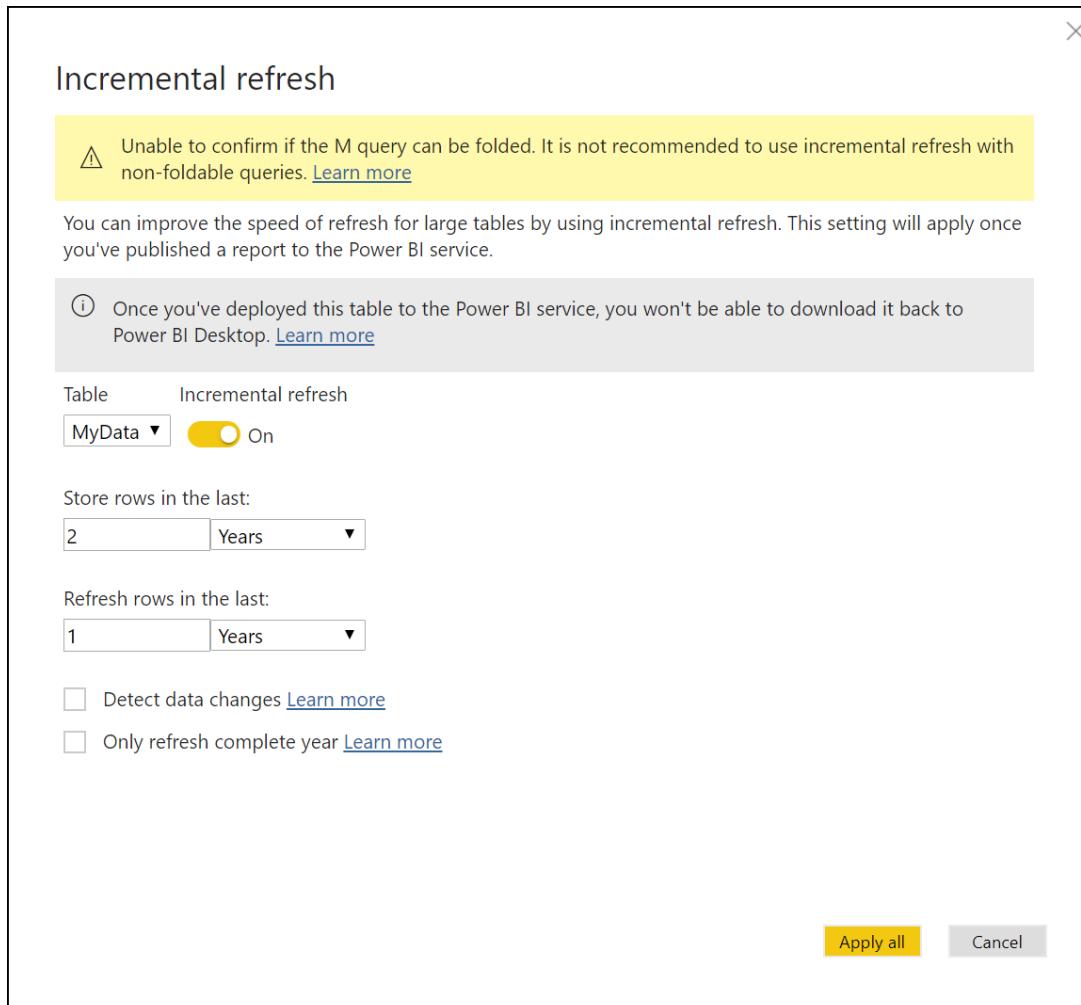
Detect data changes [Learn more](#)

Only refresh complete days [Learn more](#)

# Actualisation incrémentielle de données

⇒ Exemple :

<https://www.poweredsolutions.co/2020/01/19/incremental-refresh-for-files-in-a-folder-or-sharepoint-power-bi/>



# Actualisation incrémentielle de données

⇒ Exemple :

Stocker les lignes où la colonne "OrderDate" est dans les derniers :

1 Trimestres ▾

Actualiser les lignes où la colonne "OrderDate" est dans les derniers :

10 Jours ▾

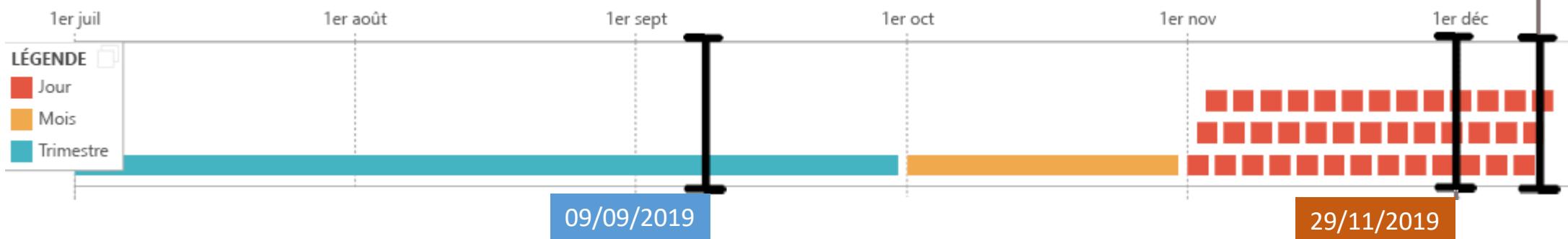
Déetecter les changements de données [En savoir plus](#)

Actualiser uniquement les éléments complets jours [En savoir plus](#)

Au 09/12/2019 la plage incrémentielle : [29/11/2019 : 09/12/2019[

- Novembre contient des jours de la période incrémentielle
- T4 contient des jours de la période incrémentielle
- T3 est l'unique trimestre complet

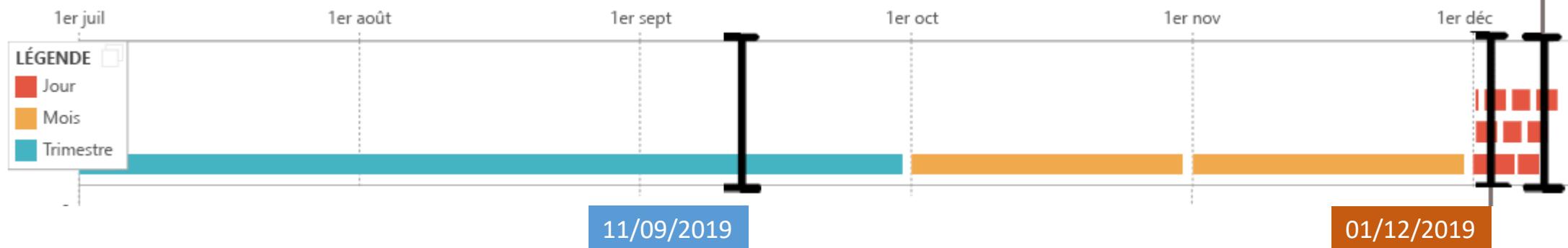
09/12/2019



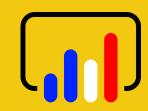
Au 11/12/2019 la plage incrémentielle : [01/12/2019 : 10/12/2019[

- Novembre ne contient plus des jours de la période incrémentielle

11/12/2019



démoc



@ClubPowerBI

# **Comment développer un custom visual**

Erwan BIZERAY - AZEO



# Introduction

- ❖ Power BI propose nativement un grand nombre de visuels, disposant de nombreux paramètres de mises en forme.
- ❖ Mais vos besoins dépasseront peut-être ces fonctionnalités
  - Graphique spécifique à un secteur professionnel
  - Imagination des demandeurs ou des UX designers
  - Refus d'utiliser les librairies graphiques de R et Python
- ❖ Microsoft met à disposition des développeurs un ensemble d'outils basés sur des frameworks « Open Source » JavaScript tels que le TypeScript, D3.js, {Less}....ou encore HTML5, CSS.



# Objectifs de la session

## ❖ Les custom viz : quels cas d'usage ?

- Quel potentiel pour quelle difficulté de réalisation ?

## ❖ Custom viz : comment se lancer ?

- Compétences requises
- Configuration de l'environnement de développement

## ❖ Custom viz : en gros, comment ça marche ?

- Création
- Compilation et tests
- Diffusion



# Sommaire

- ❖ Compétences requises
- ❖ Concepts de base
- ❖ Environnement technique
  - configuration & installation
- ❖ Compilation et tests du visuel
- ❖ Diffuser le visuel
- ❖ Cycle de développement



# Quand faut-il penser aux custom viz ?

- ❖ Lorsque le besoin de visualisation ne peut pas être remplacé par un (ou plusieurs) visuels natifs
  - Exemple : graphique de séquençage ADN
  - Rappel méthodologique : 1 visuel = 1 message
- ❖ Lorsque le besoin est pérenne
  - Le développement prendra du temps
  - Sinon, utiliser ggplot2 (R) ou matplotlib (python) ponctuellement
- ❖ Quand le visuel est réalisable en D3.js
  - Voir la galerie D3 : <https://www.d3-graph-gallery.com/index.html>



# Compétences requises

## ❖ TypeScript : « *superset of JavaScript* »

- TypeScript est un sur-ensemble de JavaScript qui prend en charge des fonctionnalités plus avancées et un accès précoce aux fonctionnalités ES6/ES7.
  - Le code des visuels « **src/visual.ts** », ainsi que les modules et interfaces **doivent être écrits en TypeScript**.
  - Tous les fichiers « .ts » TypeScript** doivent être stockés dans le répertoire « **/src** » et ajoutés au tableau des fichiers « **tsconfig.json** ».

```
public update(options: VisualUpdateOptions) {  
    let dataView: DataView = options.dataViews[0];  
    let width: number = options.viewport.width;  
    let height: number = options.viewport.height;
```

## ❖ D3.js : « *Data-Driven Documents* »

- D3.js utilise des fonctions pré-construites de JavaScript pour sélectionner des éléments, créer des objets **SVG**, les styliser, ou y ajouter des transitions, des effets dynamiques ou des **infobulles**. Ces objets peuvent aussi être stylisés à grande échelle à l'aide du célèbre langage **CSS**. Il se base notamment sur la manipulation du DOM ainsi que sur des vecteurs.
  - Historiquement, une majorité des visuels personnalisés disponibles en ligne sur le site **Custom Visual Gallery** ont été développés à partir de modèles de code D3.js

```
// Update...  
var p = d3.select("body")  
    .selectAll("p")  
    .data([4, 8, 15, 16, 23, 42])  
    .text(function(d) { return d; });  
  
// Enter...  
p.enter().append("p")  
    .text(function(d) { return d; });  
  
// Exit...  
p.exit().remove();
```



# **Configurer l'environnement de développement**



# Configurer l'environnement de développement

- ❖ Prérequis :
  - PowerShell v4 ou +
    - par défaut sur Win8.1 ou Server 2012 R2
  - Terminal (pour les utilisateurs OSX)
- ❖ Node.js (>v10)
- ❖ D3 Javascript Library
- ❖ Core.js
- ❖ powerbi-visual-api
- ❖ Une interface de développement
  - Visual Studio Code ou autres (attention aux licences)
- ❖ Paramétrage « développeur » sur le service Power BI

*La procédure complète et détaillée sera disponible dans le support PDF de la présentation*



# Installation des principaux packages

- ❖ Installation de Power BI Visuals CLI Tools (pbviz)
  - Installation via Node Package Manager (npm)
    - `npm install -g powerbi-visuals-tools`
    - `npm i -g powerbi-visuals-tools`
- ❖ Installation du certificat local
  - `pbviz --install-cert`
- ❖ Installation de la bibliothèque D3 JavaScript
  - `npm i d3@^5.0.0 -save`
- ❖ Installation des définitions de type pour D3 JavaScript
  - `npm i @types/d3@^5.0.0 -save`
- ❖ Installation de la bibliothèque Core-js
  - `npm i core-js@3.2.1 -save`
  - Cette commande installe la bibliothèque standard modulaire pour JavaScript. Elle comprend des polyremplissages pour ECMAScript jusqu'à 2019
- ❖ Installation du package Power BI Custom Visuals API
  - `npm i powerbi-visuals-api --save-dev`
  - Cette commande installe les définitions d'API de visuels Power BI





# Installation ou contrôle de la version PS1

- ❖ Windows PowerShell est installé par défaut avec chaque copie de Windows, à compter de Windows 7 SP1 et de Windows Server 2008 R2 SP1.
  - Si vous envisagez d'utiliser PowerShell 6 et versions ultérieures, vous devez installer PowerShell Core plutôt que Windows PowerShell. Pour cela, consultez [Installation de PowerShell Core sous Windows](#).
  - Si vous souhaitez contrôler ou vérifier la version du PowerShell existante, veuillez cliquer sur ce [lien](#).

```
Administrator: Windows PowerShell
PS U:\> $PSVersionTable
Name                           Value
----                           ---
CLRVersion                     2.0.50727.3615
BuildVersion                   6.0.6002.18111
PSVersion                      2.0
WSManStackVersion              2.0
PSCompatibleVersions            {1.0, 2.0}
SerializationVersion             1.1.0.1
PSRemotingProtocolVersion       2.1

PS U:\> Get-Host

Name          : ConsoleHost
Version       : 2.0
InstanceId    : 26df63d8-62de-45d9-9837-5bf118ba53a3
UI           : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture : fr-FR
CurrentUILanguage : en-US
PrivateData   : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
IsRunspacePushed : False
Runspace      : System.Management.Automation.Runspaces.LocalRunspace
```



# Installation de la plateforme Node.js

**Node.js** est une **plateforme logicielle libre en JavaScript** orientée vers les applications **réseau événementielles** hautement **concurrentes** qui doivent pouvoir **monter en charge**. Parmi les **modules natifs** de **Node.js**, on retrouve **HTTP** qui permet le développement de serveur HTTP. Concrètement, **Node.js** est un **environnement bas niveau** permettant l'exécution de **JavaScript côté serveur**.

Pour installer **Node.js**, dans un navigateur web, accédez au site [Node.js](#):

1. Téléchargez la dernière mise à jour du programme d'installation MSI.
2. Exécutez le programme d'installation, puis suivez les étapes d'installation. Acceptez les termes du contrat de licence et tous les paramètres par défaut.
3. Redémarrez l'ordinateur.



# Installation de Visual Studio Code

Pour installer Visual Studio Code, depuis un navigateur web, accédez à ce [lien](#).

1. Téléchargez la mise à jour du programme d'installation MSI que vous souhaitez utiliser.

2. Exécutez le programme d'installation, puis suivez les étapes d'installation. Acceptez les termes du contrat de licence et tous les paramètres par défaut.

**3. Redémarrez l'ordinateur.**

The screenshot shows the official Microsoft Visual Studio website. At the top, there's a banner for "Visual Studio" with the tagline "Best-in-class tools for any developer". Below the banner, three main download sections are displayed:

- Visual Studio**: Shows a screenshot of the full-featured IDE interface. Text below it says "Full-featured IDE to code, debug, test, and deploy to any platform". A "Download Visual Studio" button is present.
- Visual Studio Code**: Shows a screenshot of the code editor interface. Text below it says "Editing and debugging on any OS (By using Visual Studio Code you agree to its license and privacy statement)". A "Download Visual Studio Code" button is present.
- Visual Studio for Mac**: Shows a screenshot of the Mac interface. Text below it says "Develop apps and games for iOS, Android, and web using .NET". A "Download Visual Studio for Mac" button is present.

In the center, there's a large "Download Visual Studio Code" button with the text "Free and built on open source. Integrated Git, debugging and extensions." Below this button are icons for Windows, Linux (Tux), and Mac, each with a download arrow and a corresponding ".zip" file link.

| Platform | File Type | Link                                  |
|----------|-----------|---------------------------------------|
| Windows  | .zip      | <a href="#">Windows 7, 8, 10</a>      |
| .deb     | .deb      | <a href="#">Debian, Ubuntu</a>        |
| .rpm     | .rpm      | <a href="#">Red Hat, Fedora, SUSE</a> |
| Mac      | .zip      | <a href="#">macOS 10.10+</a>          |

At the bottom, there are links for "User Installer", "System Installer", ".deb", ".rpm", ".tar.gz", and "Snap Store".

# Activation des customs sur Power BI Service

- ❖ Depuis les paramètres du service
  - activer le visuel des développeurs

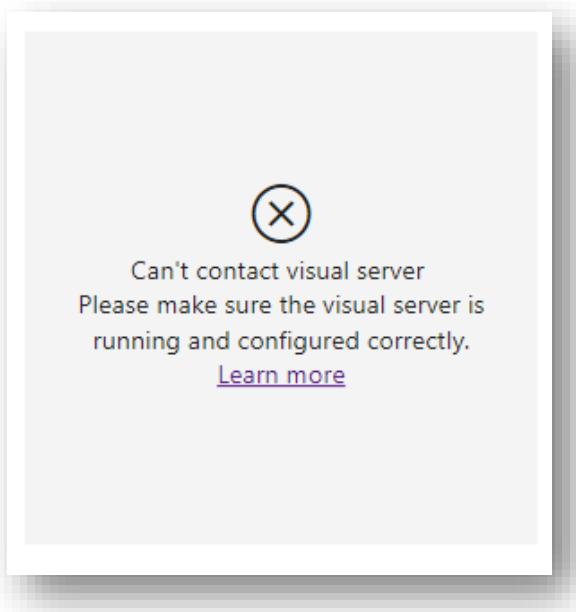
The screenshot shows the Power BI Service settings interface. On the left, there's a sidebar with options like Home, Favorites, Recent, Apps, Shared with me, Workspaces, and My workspace. The main area has tabs for General, Alerts, Subscriptions, Dashboards, Datasets, and Workbooks. The General tab is selected. A sub-menu under 'Developer' is open, showing 'Enable custom visual debugging using the developer visual' with a checked checkbox, a 'Learn more' link, and 'Apply' and 'Discard' buttons. To the right, there's a 'Visualizations' pane displaying various chart icons, and a 'Fields' pane showing fields like Date, Product, Region, Sales, and State. A 'Developer Visual' icon is highlighted with a cursor, indicating it's been activated.

- Un nouveau visuel apparaît



# Test d'activation sur Power BI Service

- ❖ Après activation du paramètre vu précédemment, nous obtenons le message d'erreur suivant :
  - Cela indique que le portail Power BI Service n'est pas en mesure de détecter un serveur HTTPS sécurisé sur le poste client
  - Pour remédier au problème, on doit d'abord installer un **certificat SSL** à partir de la commande `pbviz`.
  - Une fois le certificat SSL installé, nos composants visuels pourront être chargés dans notre **navigateur local**.



# Vérification du bon fonctionnement PBIVIZ CLI

- ❖ Vérifiez que le package Power BI Visual Tools a été installé
  - Il faut lancer la commande PowerShell suivante : pbviz

```
PS      pbviz

+syso+/
oms/+osydhso/
ym/     /+oshddhsy/
ym/     /+oyhddhyo/
ym/     /osyhdho
ym/           sm+
ym/     yddy   om+
ym/     shho /mmmm/ om+
/     oys/ +mmmm /mmmm/ om+
oso  ommmh +mmmm /mmmm/ om+
ymmy smmmh +mmmm /mmmm/ om+
ymmyy smmmh +mmmm /mmmm/ om+
ymmyy smmmh +mmmm /mmmm/ om+
+dmd+ smmmh +mmmm /mmmm/ om+
/hmdo +mmmm /mmmm/ /so//ym/
/dmhh /mmmm/ /osyhy/
//    dmmid
++
```

PowerBI Custom Visual Tool

Usage: pbviz [options] [command]

Commands:

```
new [name]      Create a new visual
info            Display info about the current visual
start           Start the current visual
package         Package the current visual into a pbviz file
validate [path]  Validate pbviz file for submission
update [version] Updates the api definitions and schemas in the current visual. Changes the version if specified
help [cmd]       display help for [cmd]
```

Options:

```
-h, --help      output usage information
-V, --version   output the version number
--create-cert   Create new localhost certificate
--install-cert  Install localhost certificate
```



# Création et installation du certificat

## ❖ Sous Windows

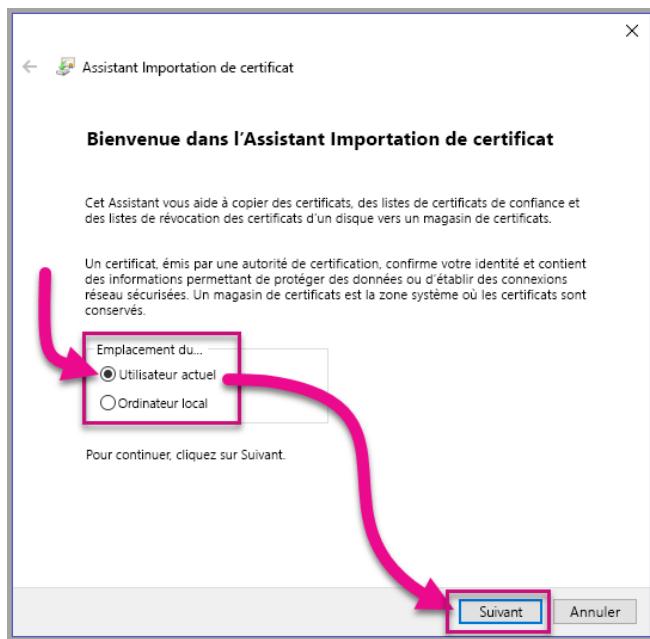
- 1. Pour créer et installer un certificat, lancer la commande  
➤ `pbviz --install-cert`

```
PS C:\Users\maghan> pbviz --create-cert
info Certificate generated. Location is C:\Users\maghan\AppData\Roaming\npm\node_modules\powerbi-visuals-tools\certs\
PowerBICustomVisualTest_public.pfx. Passphrase is '15105661266553327'
PS C:\Users\maghan>
```

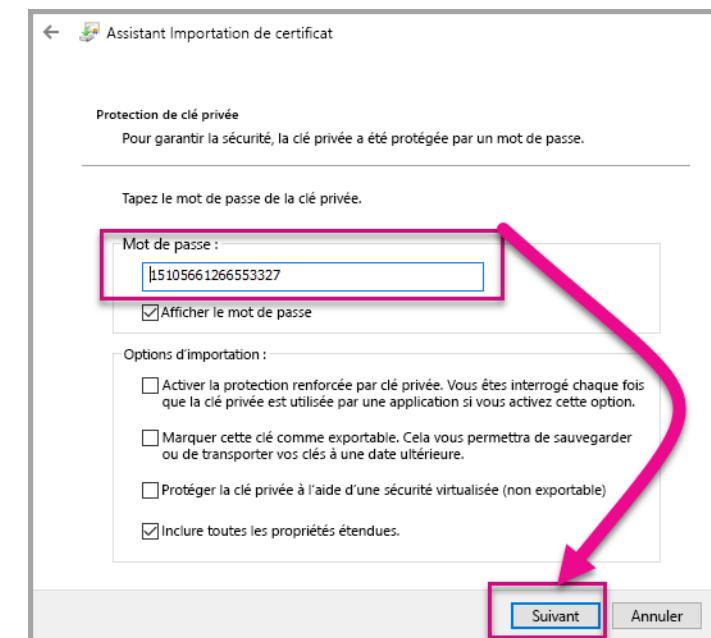
- Elle retourne un résultat qui produit une phrase secrète. Dans ce cas, la phrase secrète est **15105661266553327**. Elle démarre également l'Assistant Importation de certificat.



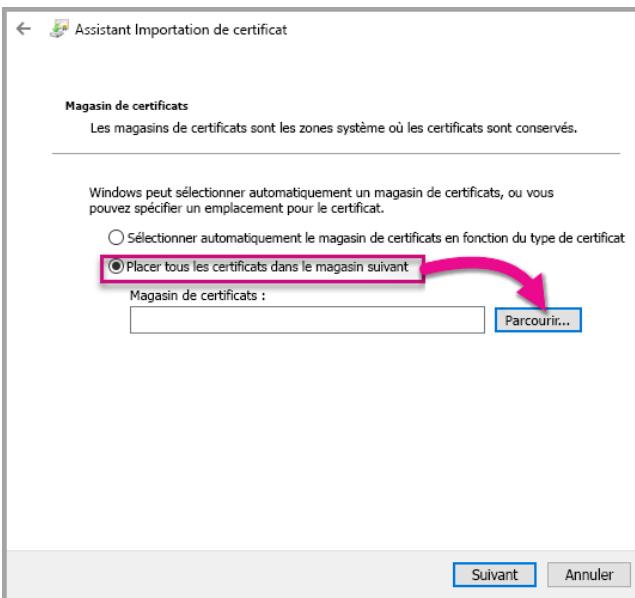
# Création et installation du certificat



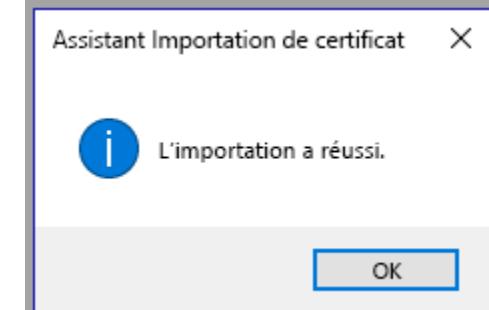
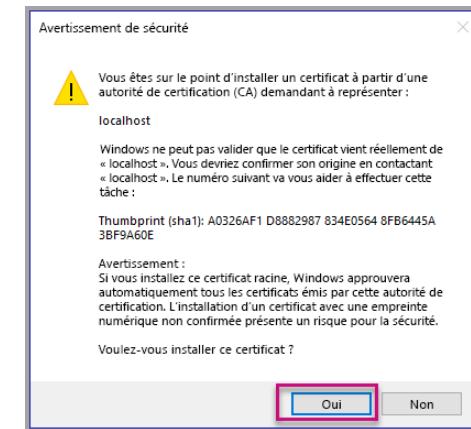
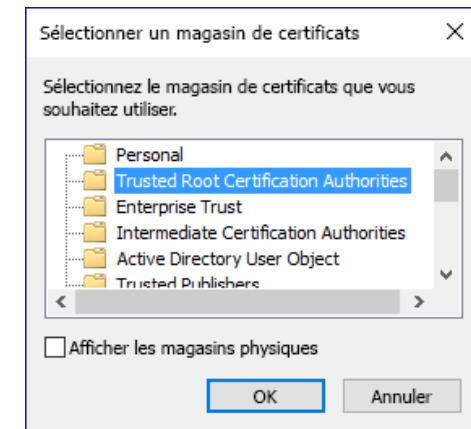
- Dans l'Assistant Importation de certificat, vérifiez que l'emplacement du magasin est défini sur l'utilisateur actuel. Sélectionnez ensuite *Suivant*.
- À l'étape **Fichier à importer**, sélectionnez *Suivant*.
- À l'étape **Protection de clé privée**, dans la zone du mot de passe, collez la phrase secrète reçue lors de la création du certificat.
  - Dans ce cas également, c'est **1510566126655327** .



# Création et installation du certificat



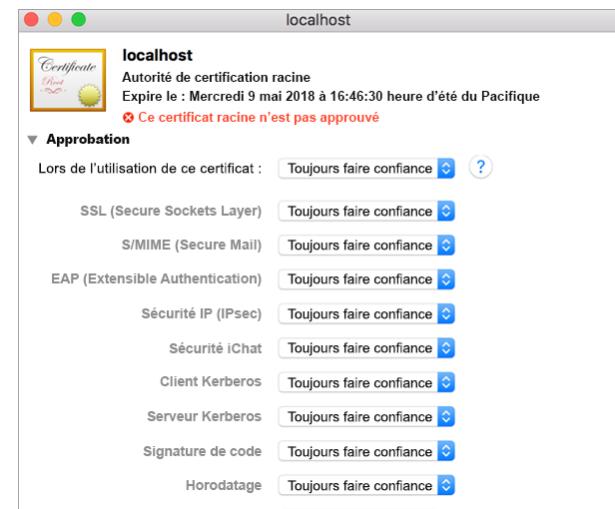
- À l'étape **Magasin de certificats**, sélectionnez l'option **Placer tous les certificats dans le magasin suivant**. Sélectionnez ensuite *Parcourir*.
- Dans la fenêtre **Sélectionner un magasin de certificats**, sélectionnez **Autorités de certification racines de confiance**, puis OK. Sélectionnez ensuite *Suivant* sur l'écran **Magasin de certificats**.
- Pour terminer l'importation, sélectionnez **Terminer**.
  - Si vous recevez un avertissement de sécurité, sélectionnez **Oui**.
- Lorsque vous êtes informé que l'importation a réussi, sélectionnez **OK**.



# Création et installation du certificat

## ❖ OSX

- Si le verrou dans le coin supérieur gauche est verrouillé, sélectionnez-le pour le déverrouiller. Recherchez *localhost*, puis double-cliquez sur le certificat.
- Sélectionnez **Toujours faire confiance**, fermez la fenêtre.
- Entrez vos nom d'utilisateur et mot de passe.
  - Sélectionnez **Mettre à jour les paramètres**.



- Fermez les navigateurs que vous avez ouverts.



# Création d'un projet Custom Visual

- Pour créer un projet de visuels personnalisés, entrez la commande suivante. **CircleCard** est le nom du projet.

» `pbiviz new CircleCard`

```
2.1.0  
PS C:\> pbiviz new CircleCard  
info Creating new visual  
info Installing packages...  
info Installed packages.  
done Visual creation complete
```

- Vous créez le projet à l'emplacement actuel de l'invite de commande (à modifier si besoin)
- Accédez au dossier du projet.

» `cd CircleCard`
- Démarrez le visuel personnalisé. Le visuel **CircleCard** est maintenant exécuté tout en étant hébergé sur votre ordinateur.

» `pbiviz start`

```
PS C:\> cd CircleCard  
PS C:\CircleCard> pbiviz start  
info Building visual...  
done build complete  
  
info Starting server...  
info Server listening on port 8080.
```

- ❖ Ne fermez pas la session Windows PowerShell
- ❖ Grâce à ces commandes, cela permet de générer un nouveau modèle de projet en quelques secondes et à lancer les tests.
- ❖ L'ossature du projet existe mais il faut tout de même développer !



# Installation des bibliothèques/modules

## ❖ Ajout de la bibliothèque D3.js

- `npm i d3@^5.0.0 --save`

```
PS C:\Users\ErwanBIZERAY\OneDrive - AZEO\PowerBI\custom> npm i d3@^5.0.0 --save
npm WARN          ENOENT: no such file or directory, open 'C:\Users\ErwanBIZERAY\package.json'
npm WARN          ENOENT: no such file or directory, open 'C:\Users\ErwanBIZERAY\package.json'
npm WARN ErwanBIZERAY No description
npm WARN ErwanBIZERAY No repository field.
npm WARN ErwanBIZERAY No README data
npm WARN ErwanBIZERAY No license field.

+ d3@5.0.0
added 36 packages from 4 contributors and audited 103 packages in 7.343s
found 0 vulnerabilities
```

## ❖ Installation des types D3.js

- `npm i @types/d3@^5.0.0 --save`

```
PS C:\Users\ErwanBIZERAY\OneDrive - AZEO\PowerBI\custom> npm i @types/d3@^5.0.0 --save
npm WARN          ENOENT: no such file or directory, open 'C:\Users\ErwanBIZERAY\package.json'
npm WARN          ENOENT: no such file or directory, open 'C:\Users\ErwanBIZERAY\package.json'
npm WARN ErwanBIZERAY No description
npm WARN ErwanBIZERAY No repository field.
npm WARN ErwanBIZERAY No README data
npm WARN ErwanBIZERAY No license field.

+ @types/d3@5.0.0
added 33 packages from 12 contributors and audited 255 packages in 9.451s
found 0 vulnerabilities
```

## ❖ Installation de core.js

- `npm i core-js@3.2.1 --save`

```
PS C:\Users\ErwanBIZERAY\OneDrive - AZEO\PowerBI\custom> npm i core-js@3.2.1 --save
> core-js@3.2.1 postinstall C:\Users\ErwanBIZERAY\node_modules\core-js
> node scripts/postinstall || echo "ignore"

Thank you for using core-js ( https://github.com/zloirock/core.js ) for polyfilling JavaScript standard library!
The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock

Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job ~

npm WARN          ENOENT: no such file or directory, open 'C:\Users\ErwanBIZERAY\package.json'
npm WARN          ENOENT: no such file or directory, open 'C:\Users\ErwanBIZERAY\package.json'
npm WARN ErwanBIZERAY No description
npm WARN ErwanBIZERAY No repository field.
npm WARN ErwanBIZERAY No README data
npm WARN ErwanBIZERAY No license field.

+ core-js@3.2.1
added 1 package and audited 302 packages in 9.335s
found 0 vulnerabilities
```

## ❖ Installation Power BI Visual API

- `npm i powerbi-visuals-api --save-dev`

```
PS C:\Users\ErwanBIZERAY\OneDrive - AZEO\PowerBI\custom\circleCard> npm i powerbi-visuals-api --save-dev
npm notice    powerbi-visuals-api is being moved from dependencies to devDependencies
npm WARN visual@ No description
npm WARN visual@ No repository field.
npm WARN visual@ No license field.

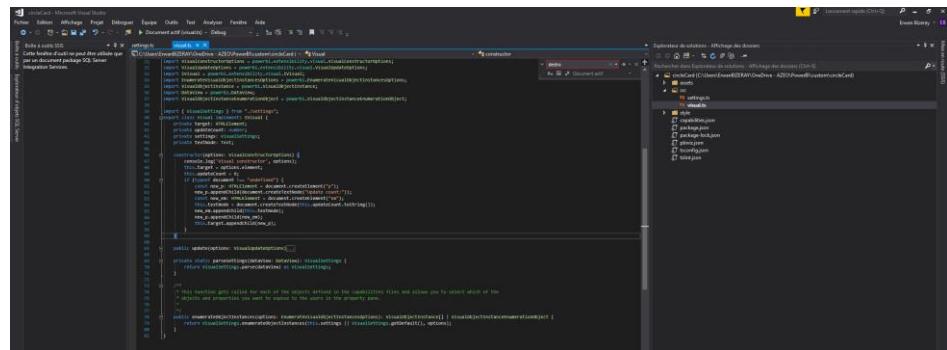
+ powerbi-visuals-api@2.6.1
updated 1 package and audited 1102 packages in 6.788s
found 0 vulnerabilities
```



# Utilisation de Visual Studio Code

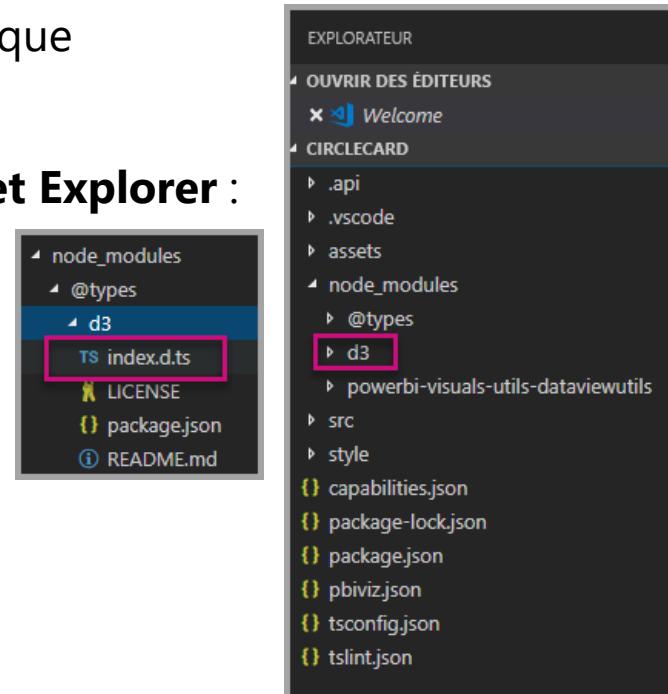
## ❖ Lancer IDE Visual Studio Code

- Depuis le chemin du visuel :
  - code .
- Ne pas oublier le point



## ❖ Vérification des modules installés

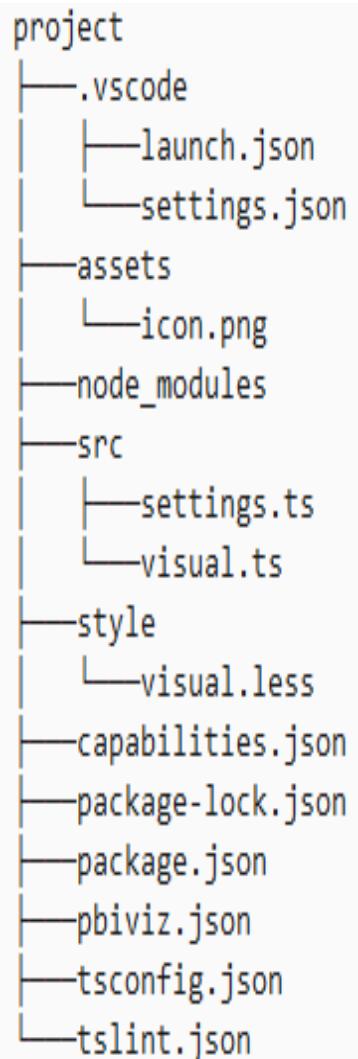
- Dans le **volet Explorateur**, développez le dossier **node\_modules** pour vérifier que la **bibliothèque d3** a été installée
- Assurez-vous que le fichier **index.d.ts** a été ajouté en développant dans le **volet Explorer** :
  - node\_modules > @types > d3



## ❖ Développez maintenant le code attendu dans votre visuel



# Architecture d'un projet custom visual



- ❖ « **.vscode/** »
  - Ce dossier contient les paramètres du projet VS Code. Pour configurer votre espace de travail, modifiez le fichier « **.vscode/settings.json** »
- ❖ « **assets/** »
  - Ce dossier est utilisé pour stocker des éléments graphiques tels que des images, icônes
- ❖ « **node\_modules/** »
  - Ce répertoire contient l'ensemble des sources des modules/paquets déployés à partir de commandes « **npm** »
- ❖ « **src/** »
  - Ce dossier est utilisé pour stocker tout code source du visuel au format TypeScript et JavaScript. Power BI crée les 2 fichiers suivants :
    - « **visual.ts** » : code source principal du visuel
    - « **settings.ts** » : code des paramètres du visuel. Les classes du fichier fournissent une interface permettant de définir les propriétés de votre visuel.
- ❖ « **style/** »
  - Ce dossier contient le fichier principal de style « less » nommé « **visual.less** », qui définit les styles du visuel.
  - Possibilité d'importer des fichiers de style externes : **@import (less) "style/external.css"**
- ❖ « **dist/** »
  - Ce répertoire contient le fichier **pbviz** qui est généré automatiquement lorsque la commande « **pbviz package** » est exécutée
- ❖ **capabilities.json**
  - Fichier de configuration utilisé pour définir les capacités relatifs, propriétés, paramètres principaux (ou fonctionnalités), mapping de vues de données pris en charge du visuel  
[**dataRole**, **dataViewMappings**, **object**]
- ❖ **package-lock.json**
  - Ce fichier est généré automatiquement pour toutes les opérations où **npm** modifie l'arborescence **node\_modules** ou le fichier **package.json**.
- ❖ **package.json**
  - Ce fichier permet d'effectuer la gestion des modules [**node\_modules**] à partir de commande **npm**.
- ❖ **pbviz.json**
  - Principal fichier de configuration du projet. Il contient les informations générales du visuel comme le nom, l'auteur, les fichiers ExternalJS,...
- ❖ **tsconfig.json**
  - Paramètres du compilateur de fichiers TypeScript. Il contient notamment la liste des fichiers .ts nécessaires (incluant les types)
- ❖ **tslint.json**
  - Ce fichier contient la configuration TSLint
- ❖ **Fichiers // dossiers facultatifs**
  - **.gitignore** : Manifeste indiquant à l'outil **git** les fichiers à ignorer, à ne pas prendre en charge
  - « **api/** » : Ce répertoire contient des informations relatives à la version.

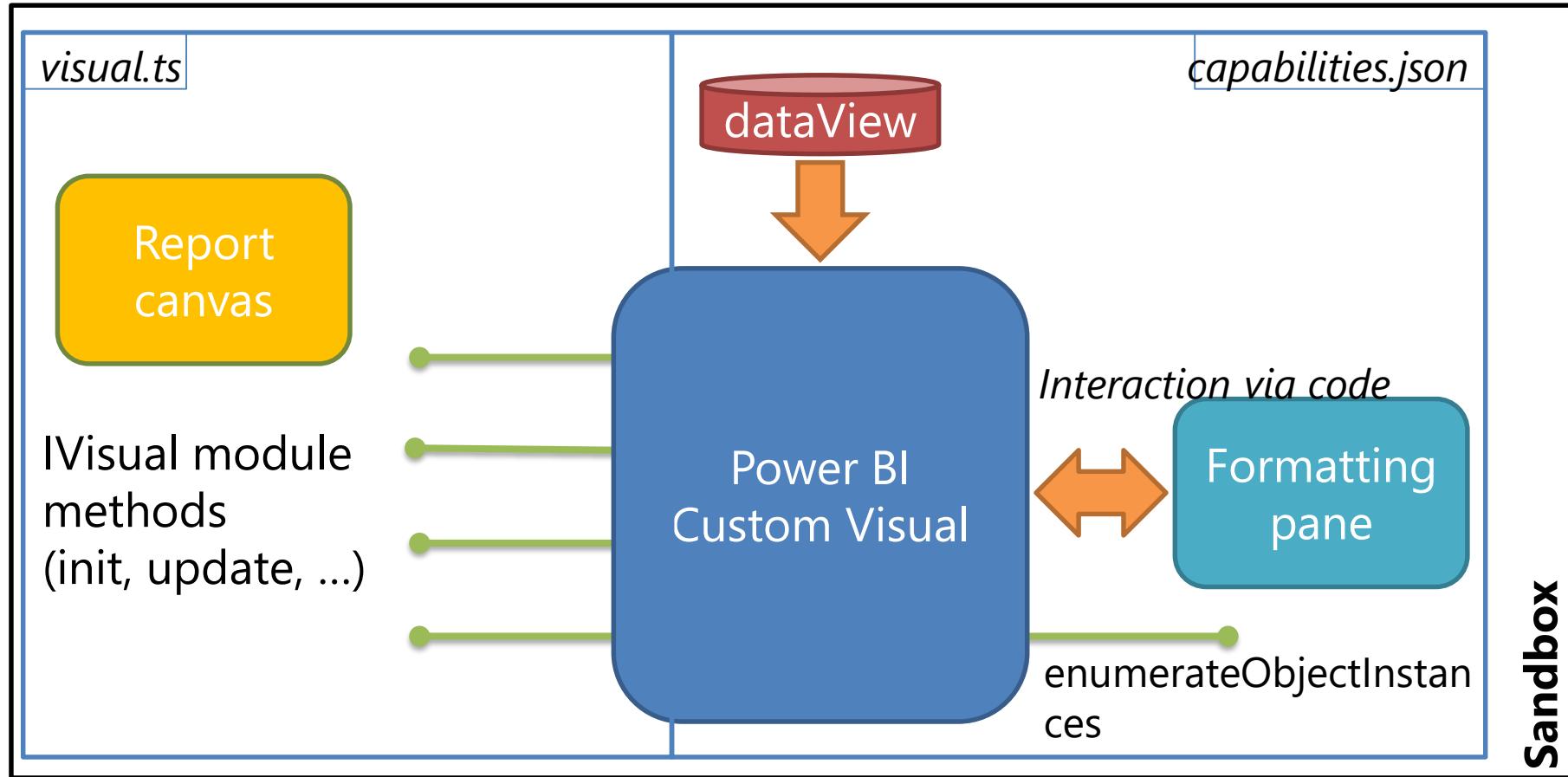


# Custom visuals : concepts de base



# Les concepts de base

## ❖ Schéma global de l'API Power BI Custom Visual



# Les concepts de base : IVisual

❖ L'interface IVisual est l'organe principal de Power BI Visuals que toute classe de composant visuel « visual.ts » doit implémenter.

❖ L'interface IVisual est exécutée en 3 temps :

➤ Constructor : `constructor(options: VisualConstructorOptions) {}`

- Cette méthode est appelée une seule fois; lorsque le visuel est initialisé pour la première fois au chargement.

➤ Update : `update(options: VisualUpdateOptions) {}`

- Cette méthode est appelée chaque fois qu'une mise à jour est effectuée sur le composant visuel. Que ce soit pour un changement au niveau des paramètres des propriétés, des modifications des données, ou bien un repositionnement du visuel par l'utilisateur.

➤ Destroy : `destroy(){} // implicite non généré`

- Cette méthode (implicite) est appelée lorsque le visuel est sur le point d'être « unloaded ». Power BI retire généralement toute l'Iframe sans appeler cette méthode.



# Les concepts de base : capabilities.json

- ❖ L'ajout de fichiers **capabilities.json** permet une meilleure séparation des éléments et leurs contrôles.
- ❖ Apparition des notions :
  - Dataroles & Mappings
  - Objects
  - Dataview

The diagram illustrates the mapping between the JSON configuration in `capabilities.json` and the corresponding settings in the Power BI Data View interface.

**capabilities.json (Left):**

```
1  "dataRoles": [
2    {
3      "displayName": "My Category",
4      "name": "category",
5      "kind": 0
6    },
7    {
8      "displayName": "My Measure",
9      "name": "measure",
10     "kind": 1
11   }
12 ],
13 "dataViewMappings": [
14   {
15     "conditions": [
16       {
17         "category": { "max": 1 },
18         "measure": { "max": 1 }
19       }
20     ],
21     "categorical": {
22       "categories": {
23         "for": { "in": "category" },
24         "dataReductionAlgorithm": { "top": {} }
25       },
26       "values": {
27         "select": [ { "wind": { "to": "measure" } } ]
28       }
29     }
30   }
31 ],
32   public static converter(dataView, colors: IDataColorPalette): ViewModel {
33   178   var series = dataView.categorical.values;
34   179   return { value: series[0].values[series[0].values.length - 1], ... }
35   180 }
36   181 }
```

**Power BI Data View (Right):**

The Data View interface shows the following settings:

- Letter Frequencies:** A data view containing:
  - frequency**: A measure.
  - letter**: A category.
- My Category:** A data role mapping for the `category` role.
- My Measure:** A data role mapping for the `measure` role.
- frequency**: A selected measure.
- letter**: A selected category.

Arrows from the JSON code point to specific sections of the Data View interface, indicating how each part of the JSON configuration is reflected in the tool's interface.



# Les concepts de base : **dataView**

- ❖ Un **dataView** est un modèle d'objet JavaScript pour les représentations canoniques des données. Il ne s'agit pas vraiment d'un format JSON mais plutôt d'une forme d'arborescence ou graphe d'objet avec des références croisées supportant des vues multiples, simples et canoniques des mêmes données, et laissant aux visualisations la liberté de choisir leur modèle de représentations préféré.
  - Les structures canoniques **dataView** sont :
    - Single, Categorical, Table, Matrix, Tree, Metadata
  - Chaque **mapping** valide produit une vue de données. En général, vous n'avez qu'une seule vue de données, toutefois, vous pouvez fournir plusieurs mappings de données dans certaines conditions.
  - Power BI crée un mappings à une vue de données SI et SEULEMENT SI le mappage est renseigné dans *dataViewMappings*.

```
JSON  
  
"dataViewMappings": [  
  {  
    "conditions": [ ... ],  
    "categorical": { ... },  
    "single": { ... },  
    "table": { ... },  
    "matrix": { ... }  
  }  
]
```



# Les concepts de base : dataView

## ❖ DataView – Single

- Le mapping de données unique est la forme la plus simple du mapping de données. Il accepte un **champ de mesure unique** et donne une agrégation. Si le champ est numérique, vous obtenez la somme. Dans le cas contraire, vous obtenez un nombre de valeurs uniques.

The image shows a comparison between JSON configurations and TypeScript code for a single value visual in Power BI.

**JSON Configuration:**

```
{
  "dataRoles": [
    {
      "displayName": "Y",
      "name": "Y",
      "kind": "Measure"
    }
  ],
  "dataViewMappings": [
    {
      "conditions": [
        {
          "Y": {
            "max": 1
          }
        }
      ],
      "single": {
        "role": "Y"
      }
    }
  ]
}
```

**Power BI Data View Configuration:**

```
{
  "dataView": [
    {
      "single": {
        "value": 94163140.3560001
      }
    ]
  ]
}
```

**TypeScript Code:**

```
"use strict";
import powerbi from "powerbi-visuals-api";
import DataView = powerbi.DataView;
import DataViewSingle = powerbi.DataViewSingle;
// standard imports
// ...

export class Visual implements IVisual {
  private target: HTMLElement;
  private host: IVisualHost;
  private valueText: HTMLParagraphElement;

  constructor(options: VisualConstructorOptions) {
    // constructor body
    this.target = options.element;
    this.host = options.host;
    this.valueText = document.createElement("p");
    this.target.appendChild(this.valueText);
    // ...
  }

  public update(options: VisualUpdateOptions) {
    const dataView: DataView = options.dataViews[0];
    const singleDataView: DataViewSingle = dataView.single;

    if (!singleDataView ||
      !singleDataView.value) {
      return;
    }

    this.valueText.innerText = singleDataView.value.toString();
  }
}
```

**Power BI Visual Preview:**

A screenshot of the Power BI desktop interface showing a single value visual. The visual displays the value "896113". The Power BI ribbon is visible at the top, and the "Visuals" tab is selected. The "Visual level filters" pane on the right shows a filter for "Country" set to "(All)".



# Les concepts de base : dataView

## ❖ DataView – Categorical

- Un mapping de données par catégorie permet de récupérer un ou deux regroupements indépendants de données.



```
TypeScript
Copier
use strict;
import powerbi from "powerbi-visuals-api";
import DataView = powerbi.DataView;
import DataViewCategoricalSingle = powerbi.DataViewCategorical;
import DataViewValueColumnGroup = powerbi.DataViewValueColumnGroup;
import PrimitiveValue = powerbi.PrimitiveValue;
// ...
// standard imports
// ...

export class Visual implements IVisual {
  private target: HTMLElement;
  private host: IVisualHost;
  private categories: HTMLElement;

  constructor(options: VisualConstructorOptions) {
    // constructor body
    this.target = options.element;
    this.host = options.host;
    this.categories = document.createElement("pre");
    this.target.appendChild(this.categories);
    // ...
  }

  public update(options: VisualUpdateOptions) {
    const dataView = options.dataViews[0];
    const categoricalDataView: DataViewCategorical = dataView.categorical;

    if (!categoricalDataView || !categoricalDataView.categories || !categoricalDataView.categories[0] || !categoricalDataView.values) {
      return;
    }

    // Categories have only one column in data buckets
    // If you want to support several columns of categories data bucket, you should iterate categoricalDataView.categories[0].values array
    const categoryFieldIndex = 0;
    // Measure has only one column in data buckets.
    // If you want to support several columns on data bucket, you should iterate years.values array in map
    const measureFieldIndex = 0;
    let categories: PrimitiveValue[] = categoricalDataView.categories[categoryFieldIndex].values;
    let values: DataViewValueColumnGroup[] = categoricalDataView.values.grouped();

    let data = [];
    // iterate categories/countries
    categories.map((category: PrimitiveValue, categoryIndex: number) => {
      data[category.toString()] = {};
      // iterate years/years
      values.map((years: DataViewValueColumnGroup) => {
        if ((data[category.toString()])[years.name] && years.values[measureFieldIndex].values[categoryIndex]) {
          data[category.toString()][years.name] = [];
        }
        if (years.values[0].values[categoryIndex]) {
          data[category.toString()][years.name].push(years.values[measureFieldIndex].values[categoryIndex]);
        }
      });
    });

    this.categories.innerText = JSON.stringify(data, null, 2);
    console.log(data);
  }
}
```



# Les concepts de base : dataView

## ◆ DataView – Table

- La vue de données de table est un mappage de données simple. En gros, il s'agit d'une liste de points de données, où les points de données numériques peuvent être agrégés.

The diagram illustrates the mapping process from JSON data roles to a final Power BI visual:

- JSON Data Roles:**

```
"dataRoles": [ { "displayName": "Column", "name": "column", "kind": "Measure" }, { "displayName": "Value", "name": "value", "kind": "Measure" } ]
```
- JSON DataView Mappings:**

```
"dataViewMappings": [ { "table": { "rows": { "select": [ { "for": { "in": "column" } }, { "for": { "in": "value" } } ] } } } ]
```
- LESS Styling:**

```
table { display: flex; flex-direction: column; } tr, th { display: flex; flex: 1; } td { flex: 1; border: 1px solid black; }
```
- JSON Table Structure:**

```
{ "table": { "columns": [...], "rows": [ [ "Canada", 2014, 630 ], [ "Canada", 2015, 490 ], [ "Mexico", 2013, 645 ], [ "UK", 2014, 831 ], [ "USA", 2015, 650 ], [ "USA", 2016, 350 ] ] }}
```
- TypeScript Class:**

```
"use strict"; import "./style/visual.less"; import powerbi from "powerbi-visuals-api"; // ... import DataViewMetadataColumn = powerbi.DataViewMetadataColumn; import DataViewTable = powerbi.DataViewTable; import DataViewTableRow = powerbi.DataViewTableRow; import PrimitiveValue = powerbi.PrimitiveValue; // other imports // ...
```

```
export class Visual implements IVisual { private target: HTMLElement; private host: IVisualHost; private table: HTMLParagraphElement; constructor(options: VisualConstructorOptions) { // constructor body this.target = options.element; this.host = options.host; this.table = document.createElement("table"); this.target.appendChild(this.table); // ... } public update(options: VisualUpdateOptions) { const dataView: DataView = options.dataViews[0]; const tableDataView: DataViewTable = dataView.table; if (!tableDataView) { return } while(this.table.firstChild) { this.table.removeChild(this.table.firstChild); } // draw header const tableHeader = document.createElement("thead"); tableDataView.columns.forEach(column: DataViewMetadataColumn => { const tableHeaderColumn = document.createElement("th"); tableHeaderColumn.innerText = column.displayName; tableHeader.appendChild(tableHeaderColumn); }); this.table.appendChild(tableHeader); // draw rows tableDataView.rows.forEach((row: DataViewTableRow) => { const tableRow = document.createElement("tr"); row.forEach((columnValue: PrimitiveValue) => { const cell = document.createElement("td"); cell.innerText = columnValue.toString(); tableRow.appendChild(cell); }) this.table.appendChild(tableRow); }); }
```
- Power BI Visual:** A screenshot of the Power BI visual showing a table with columns Country, Year, and Sales. The data includes Canada (2014: 630, 2015: 490), Mexico (2013: 645), UK (2014: 831), USA (2015: 650, 2016: 350). A total row shows 3596.



# Les concepts de base : dataView

## Dataview – Matrix

- Le mappage de données de matrice est semblable au mapping de données de table, mais les lignes sont présentées de façon **hiérarchique**. Toutes les valeurs de rôle de données peuvent être utilisées comme valeur d'en-têtes de colonnes.

```
JSON
{
  "dataRoles": [
    {
      "name": "Category",
      "displayName": "Category",
      "displayNameKey": "Visual_Category",
      "kind": "Grouping"
    },
    {
      "name": "Column",
      "displayName": "Column",
      "displayNameKey": "Visual_Column",
      "kind": "Grouping"
    },
    {
      "name": "Measure",
      "displayName": "Measure",
      "displayNameKey": "Visual_Values",
      "kind": "Measure"
    }
  ],
}
```



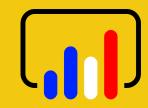
```
JSON
{
  "dataViewMappings": [
    {
      "matrix": {
        "rows": [
          {
            "for": {
              "in": "Category"
            }
          },
          {
            "columns": [
              {
                "for": {
                  "in": "Column"
                }
              },
              {
                "values": [
                  {
                    "select": [
                      {
                        "for": {
                          "in": "Measure"
                        }
                      }
                    ]
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Power BI Custom Visual API



```
JSON
{
  "matrix": {
    "rows": [
      {
        "level": 0,
        "levelValues": [...],
        "value": "Parent1",
        "identity": {...}
      },
      {
        "level": 1,
        "levelValues": [...],
        "value": "Child1",
        "identity": {...}
      },
      {
        "level": 2,
        "levelValues": [...],
        "value": "Grand child1",
        "identity": {...}
      },
      {
        "level": 1,
        "levelValues": [...],
        "value": "Child2",
        "identity": {...}
      },
      {
        "level": 2,
        "levelValues": [...],
        "value": "Grand child2",
        "identity": {...}
      },
      ...
    ],
    "columns": [
      {
        "level": 0,
        "levelValues": [...],
        "value": "Col1",
        "identity": {...}
      },
      {
        "level": 0,
        "levelValues": [...],
        "value": "Col2",
        "identity": {...}
      }
    ],
    "valueSources": [...]
  }
}
```

| Parents      | Col1 | Col2 | Total |
|--------------|------|------|-------|
| Parent1      | 24   | 31   | 55    |
| Child1       | 12   | 14   | 26    |
| Grand child1 | 5    | 6    | 11    |
| Grand child2 | 7    | 8    | 15    |
| Child2       | 12   | 17   | 29    |
| Grand child3 | 5    | 3    | 8     |
| Grand child4 | 4    | 9    | 13    |
| Grand child5 | 3    | 5    | 8     |
| Parent2      | 18   | 16   | 34    |
| Child3       | 18   | 16   | 34    |
| Grand child6 | 1    | 2    | 3     |
| Grand child7 | 7    | 1    | 8     |
| Grand child8 | 10   | 13   | 23    |
| Total        | 42   | 47   | 89    |



# Les concepts de base : dataView

- ❖ Pour visualiser ses données, lors des tests en ligne sur l'environnement Power BI Service, il faut cliquer sur le bouton d'affichage du DataView au format JSON.

The screenshot shows the Power BI Service interface with a JSON representation of a DataView. The JSON structure includes:

- dataRoles**:
  - My Category (Kind: 0)
  - My Measure (Kind: 1)
- dataViewMappings**:
  - conditions**:
    - category: max: 1
    - measure: max: 1
  - categorical**:
    - categories**:
      - for: category
      - dataReductionAlgorithm: top: {}
    - values**:
      - select: bind: to: measure
- values**:
  - source**:
    - 765
    - 468
    - 398
    - 173
    - 138
    - 76
    - 63
    - 29
  - minLocal**: 29
  - maxLocal**: 765
- tree**:
  - table**:
    - matrix**:
      - ...



# Compilation et tests du visuel



@ClubPowerBI

# Compiler et tester le visuel custom

- ❖ Pour compiler le code du projet, il faut lancer la commande

- pbviz start

- Cette commande va établir, via le certificat SSL, le lien entre le portail Power BI Service, le rapport utilisant la console développeur et Visual Studio Code.

```
PS C:\Users\ErwanBIZERAY\OneDrive - AZEO\PowerBI\custom\circleCard> pbviz start
                                         Power BI Custom Visual API

info  Starting server...
info  Start preparing plugin template
i @wds@: SSL Certificate is more than 30 days old. Removing.
i @wds@: Generating SSL Certificate
(node:16076) DeprecationWarning: Tapable.plugin is deprecated. Use new API on `hooks` instead
i @wds@: Project is running at https://localhost:8080/webpack-dev-server/
i @wds@: webpack output is served from /assets
i @wds@: Content not from webpack is served from C:\Users\ErwanBIZERAY\OneDrive - AZEO\PowerBI\custom\circleCard\.tmp\drop
info  Finish preparing plugin template
info  Start packaging...
info  Finish packaging

DONE Compiled successfully in 21934ms                                         15:41:05
```



# Quelques astuces pour débugger votre visuel

## ❖ Utilisation des points d'arrêts

- Étant donné que le JavaScript du visuel est entièrement rechargeé à chaque mise à jour du visuel, tous les points d'arrêt que vous ajoutez seront perdus lors de l'actualisation du visuel de débogage.
- Pour contourner ce problème, utilisez des instructions de débogage dans votre code.

```
public update(options: VisualUpdateOptions) {  
    console.log('Visual update', options);  
    debugger;  
    this.target.innerHTML = `<p>Update count: <em>${{this.updateCount++}}</em></p>`;  
}
```

- Il est recommandé de désactiver le chargement automatique lors de l'utilisation du débogueur dans votre code.



# Affichage des exceptions

- ❖ Lorsque vous travaillez sur votre visuel, vous remarquerez que toutes les erreurs sont «mangées» par le service Power BI. Il s'agit d'une fonctionnalité intentionnelle de Power BI pour éviter que les visuels qui se comportent mal entraînent l'instabilité de l'ensemble de l'application.
- ❖ Pour contourner ce problème, vous devez ajouter du code pour intercepter et consigner par programme vos exceptions ou simplement définir votre débogueur pour «casser les exceptions interceptées»

- **Consigner les exceptions par programmation**

- Pour consigner les exceptions dans votre visuel, ajoutez le code suivant à votre visuel pour définir un décorateur de consignation des exceptions:

```
module powerbi.extensibility.visual {
    export function logExceptions(): MethodDecorator {
        return function (target: Object, propertyKey: string, descriptor: TypedPropertyDescriptor<Function>)
            : TypedPropertyDescriptor<Function> {

            return {
                value: function () {
                    try {
                        return descriptor.value.apply(this, arguments);
                    } catch (e) {
                        console.error(e);
                        throw e;
                    }
                }
            }
        }
    }
}
```

- Ensuite, vous pouvez utiliser ce décorateur sur n'importe quelle fonction pour voir la journalisation des erreurs comme ceci:

```
@logExceptions()
public update(options: VisualUpdateOptions) {
```

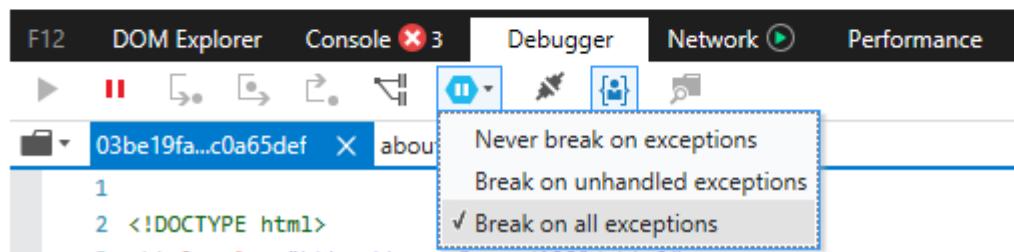


# Pause sur les exceptions

- ❖ Vous pouvez également configurer le navigateur pour qu'il interrompt les exceptions interceptées, ce qui arrêtera l'exécution de code chaque fois qu'une erreur se produit et vous permettra de déboguer à partir de là.

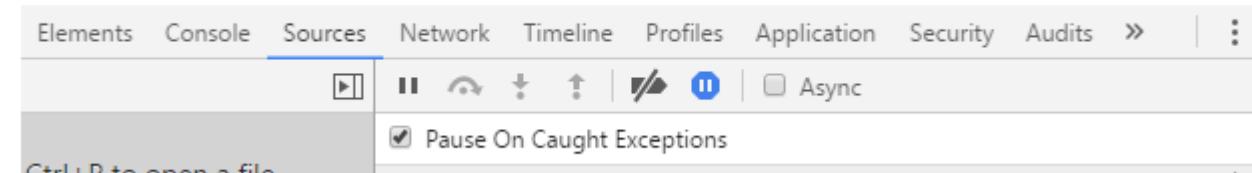
## Edge

- Open developer tools (F12)
- Go to the **Debugger** tab
- Click the **break on exceptions** icon (hexagon with a pause symbol)
- Select **Break on all exceptions**

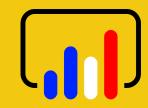


## Chrome

- Open developer tools (F12)
- Go to the **Sources** tab
- Click the **break on exceptions** icon (stop sign with a pause symbol)
- Check the **Pause On Caught Exceptions** checkbox



démoc



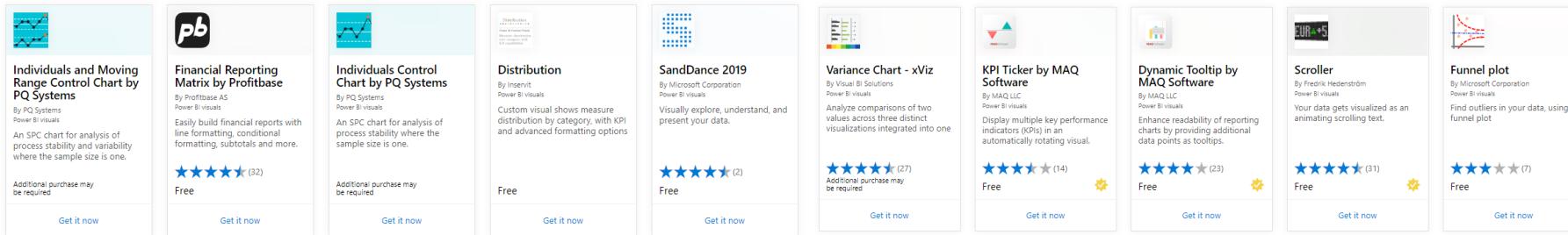
@ClubPowerBI

# **Méthodes de diffusion d'un visuel custom**

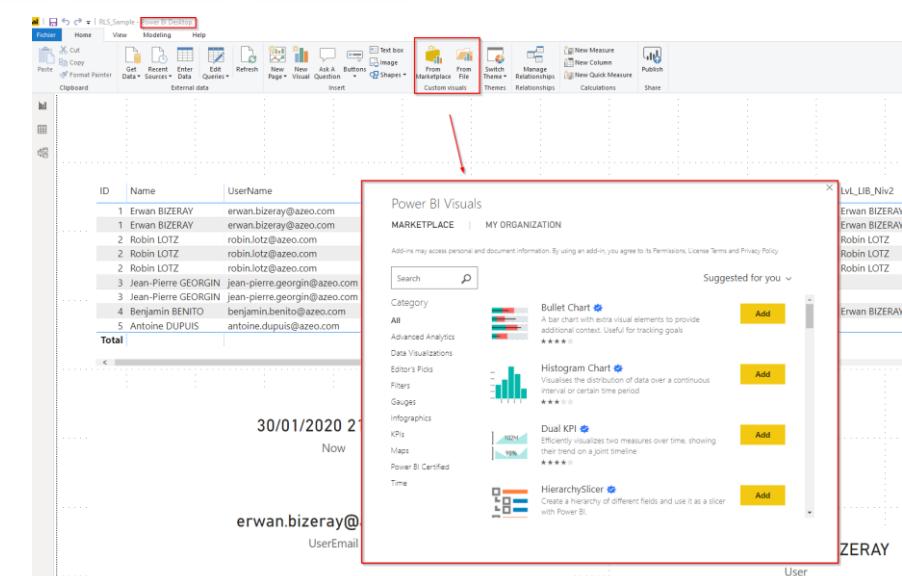


# Ressources existantes

- Suite à sa dépréciation du **Power BI Visual CORE**, certains d'entre eux ont été migrés vers **Power BI Visuals CLI** et sont dorénavant disponibles via le site « Place du marché » directement à partir de l'[AppSource](#)



- Soit directement à partir de l'outil **Power BI Destkop** dans le menu **Custom Visuals**



# Magasin d'organisation & AppSource

## ❖ L'import d'un visuel se fait depuis différents sources :

- **Fichier .pbviz en local**
- **Magasin d'organisation**

➤ Visuels validés au sein de votre organisation

### ■ **Visuels Power BI AppSource dont Visuels Power BI certifiés**

➤ Crées par Microsoft et les partenaires Microsoft

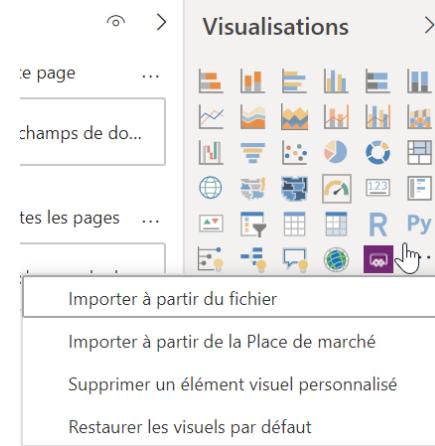
➤ testés et validés par l'équipe de validation AppSource

- Ces visuels sont testés pour vérifier qu'ils n'accèdent pas à des services ou ressources externes, et qu'ils suivent les modèles et instructions de codage sécurisé.

- Un visuel **Power BI certifié offre** davantage de fonctionnalités

- » **exporter vers PowerPoint**

- » **afficher le visuel dans les e-mails reçus** quand un utilisateur s'abonne aux pages du rapport.



# Magasin d'organisation

- ❖ Les administrateurs Power BI approuvent et déploient des visuels Power BI dans leur organisation.
  - Les auteurs de rapports peuvent ainsi facilement détecter, mettre à jour et utiliser ces visuels Power BI.
  - Les administrateurs peuvent facilement gérer ces visuels avec des actions telles que la mise à jour de versions, la désactivation et l'activation de visuels Power BI.



# Magasin d'organisation

- ❖ Les administrateurs Power BI approuvent et déploient des visuels Power BI dans leur organisation.
  - Les auteurs de rapports peuvent ainsi facilement détecter, mettre à jour et utiliser ces visuels Power BI.
  - Les administrateurs peuvent facilement gérer ces visuels avec des actions telles que la mise à jour de versions, la désactivation et l'activation de visuels Power BI.
- ❖ Accès au magasin d'organisation:
  - Dans le volet *Visualisation*, cliquez sur **Importer un élément visuel personnalisé**, sélectionnez **Importer à partir de la Place de marché** et, en haut de la fenêtre *Visuels Power BI*, sélectionnez l'onglet **Mon organisation**.
- ❖ Processus de déploiement
  - Le schéma suivant montre le processus par lequel des visuels Power BI d'organisation dans Power BI passent de l'administrateur à l'analyste des données via le développement et la maintenance.



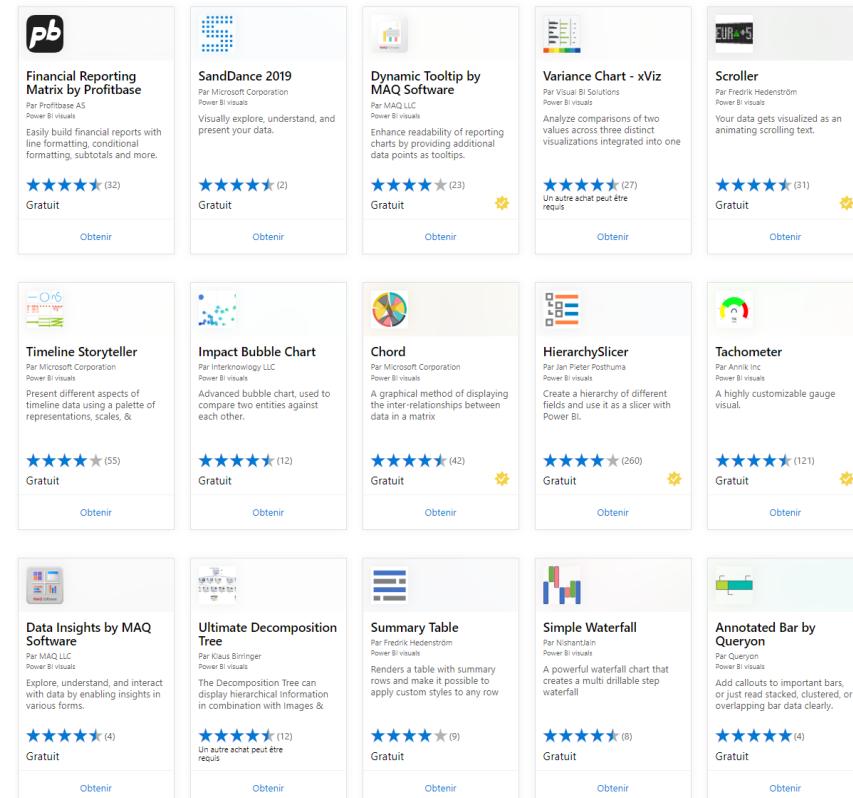
# Visuels Power BI sur l'AppSource

❖ AppSource est l'endroit où figurent des applications, des compléments et des extensions pour vos logiciels Microsoft.

- Les membres de la communauté et Microsoft mettent à disposition du public leurs visuels Power BI et les publient sur AppSource.

## ❖ Visuels Power BI certifiés

- Les visuels Power BI certifiés sont des visuels qui sont conformes à certaines exigences de codes spécifiées, celles-ci ayant été testées et approuvées par l'équipe Microsoft Power BI.
- Les tests sont conçus pour vérifier que le visuel n'a pas accès à des services ni à des ressources externes.



# Critères de certification pour AppSource

## ❖ Conditions générales

- Office Seller Dashboard // Partners + AppSource hosting

## ❖ Exigences relatives au référentiel de code

- Approbation PBI Team + GitHub + Certified Branch + Code Only

## ❖ Exigences relatives aux fichiers

- Dernière version API + Package des fichiers spécifiques

## ❖ Exigences relatives aux commandes

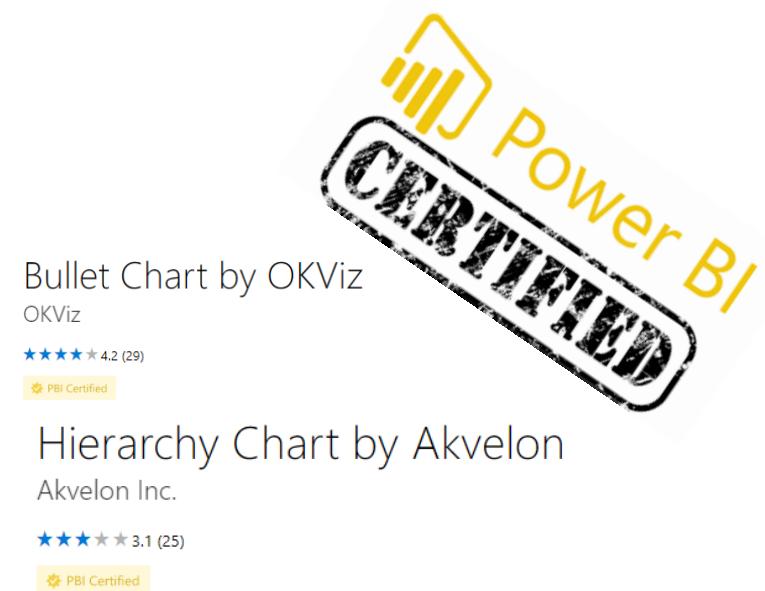
- Vérification des commandes sans erreur (npm+pbviz package+npm Audit + TSLint)

## ❖ Exigences relatives à la compilation

- Dernière version de [powerbi-visuals-tools](#)

## ❖ Exigences relatives au code source

- Respect des stratégies de certification supplémentaires des visuels Power BI.
  - Si votre envoi ne suit pas ces instructions, l'e-mail de rejet de l'Espace partenaires comportera les numéros de stratégie indiqués dans ce lien.
- Vérification que votre code est conforme aux stratégies de certification Power BI pour les éléments obligatoires et/ou Non autorisé

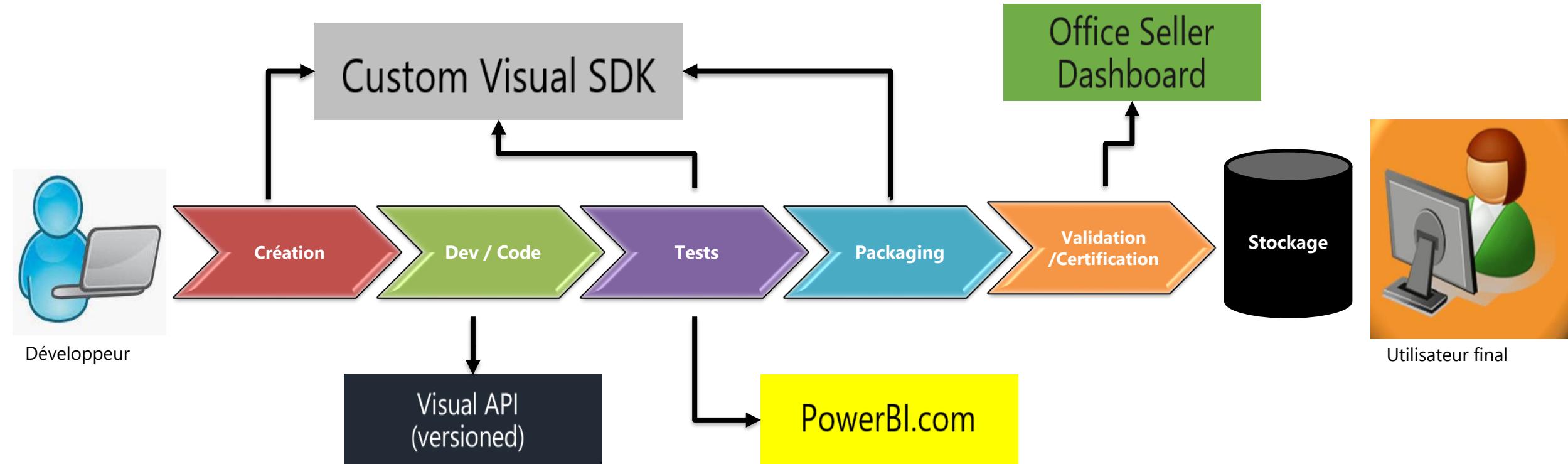


# Soumission d'un visuel pour certification

- ❖ Le processus de certification Power BI peut prendre du temps. Il est recommandé de publier votre visuel Power BI via l'Espace partenaires avant de demander une certification.
- ❖ Connectez-vous à l'Espace partenaires.
  - Sur la **page Vue d'ensemble**, choisissez votre visuel Power BI, puis accédez à la page de configuration du **produit**.
  - Cochez la case **Demander la certification Power BI**.
  - Sur la page **Vérifier et publier**, dans la zone de texte **Remarques pour la certification**, fournissez un lien vers le code source et les informations d'identification requises pour y accéder.



# Cycle de développement d'un visuel custom



# **REX SNCF Gares & Connexions**

Elise NICOLLEAU – Arnaud LUTELLIER



# ECHO

ANALYSE DE LA PERFORMANCE DES ANNONCES SONORES PAR LA RECONNAISSANCE VOCALE



GARES &  
CONNEXIONS

SNCF

# SOMMAIRE

CONTEXTE

VIDEO DE PRESENTATION

STRUCTURATION

DEMO

# CONTEXTE

## SITUATION ACTUELLE

Les annonces sonores en gare constituent l'un des principaux canaux d'information des clients, notamment lors des situations perturbées. La performance en termes de qualité de l'Information Voyageur associée à ces annonces apparaît comme un levier essentiel d'augmentation de la satisfaction client.

2 types d'annonces sont diffusées en gare



Annonces réalisées par les agents  
Annonces sonores numérisées « Simone »

**ECHO**  
Ecoutes à CHaud Opérationnelles

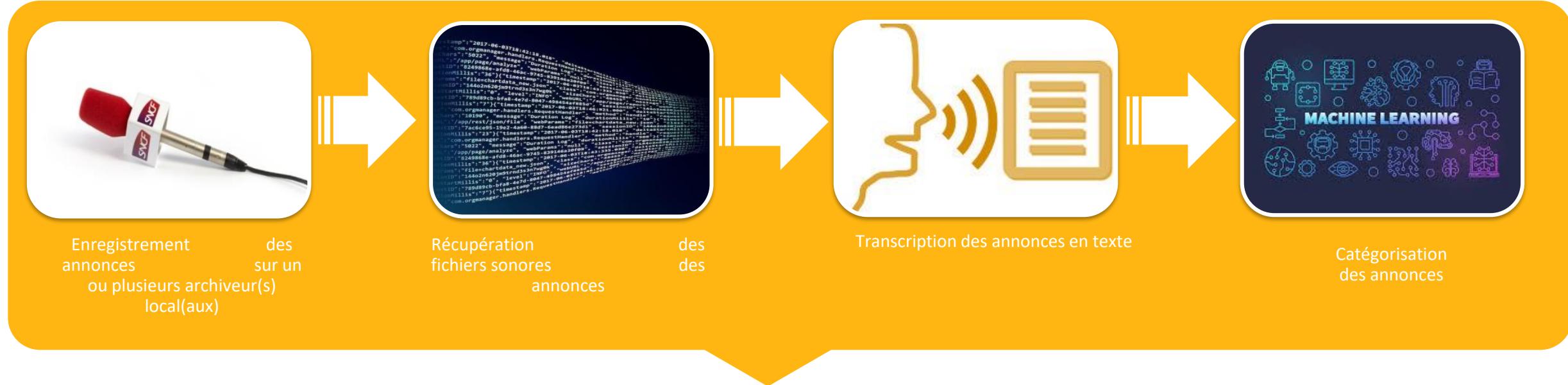
Mise en place d'un outil  
de réécoute et d'analyse des annonces sonores

# PRESSENTATION VIDEO

-

# STRUCTURATION

## CHAINÉE DE TRAITEMENT



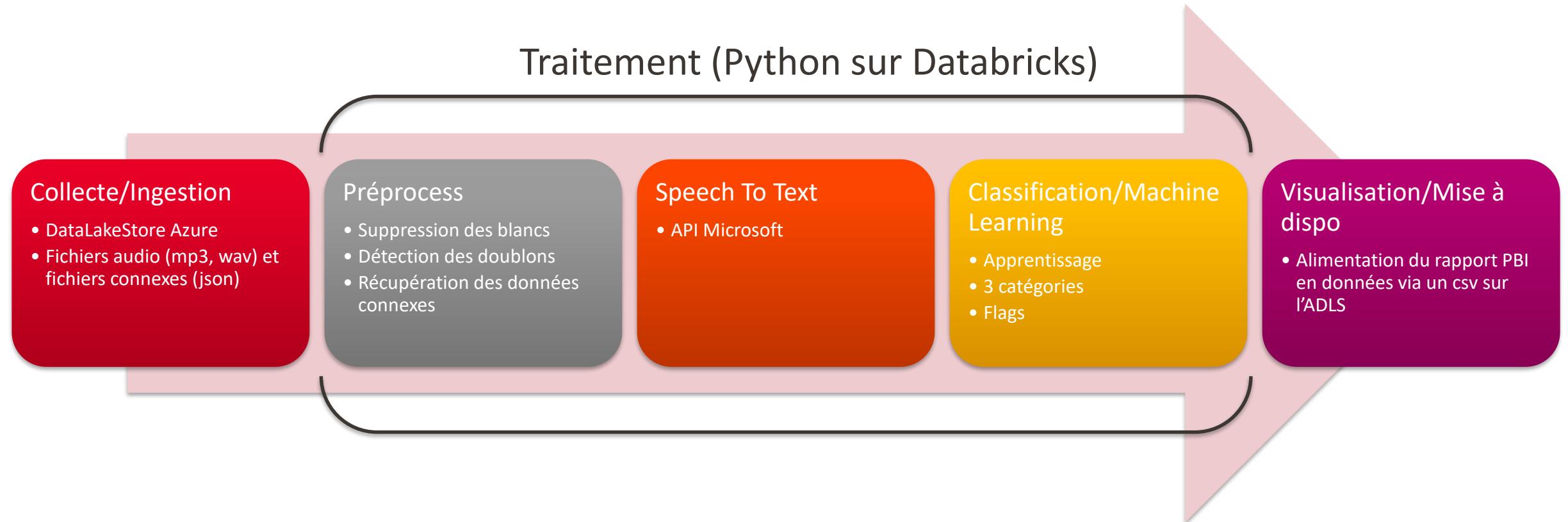
POWER BI Interface manager

GARES &  
CONNEXIONS



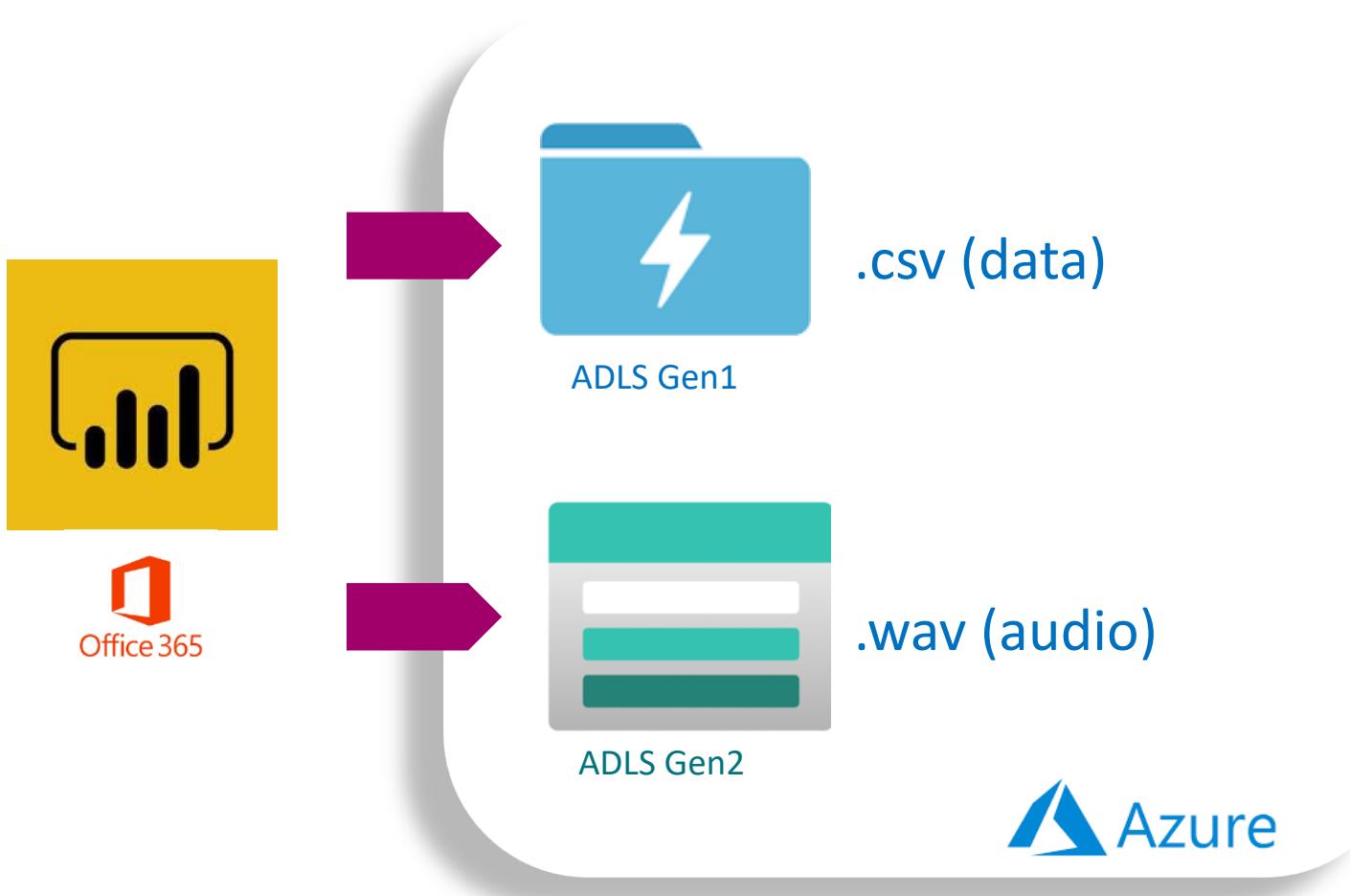
# STRUCTURATION

## CHAINTE TECHNIQUE

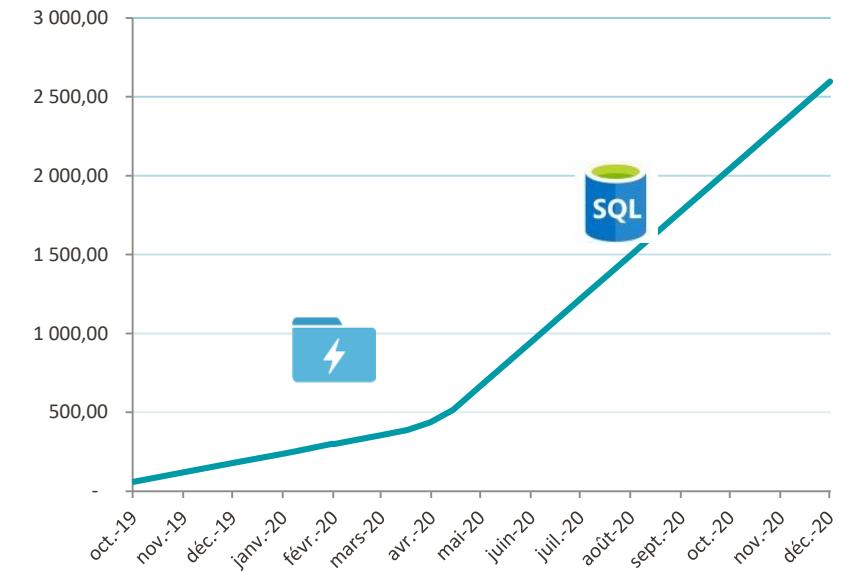


# STRUCTURATION

## SOURCES



Taille fichier data en Mo



# STRUCTURATION

## CUSTOM VISUALS

Filtre par date/heure

This screenshot shows a user interface for filtering announcements. At the top, there are two dropdown menus: "Date/heure" containing "01/02/2020 15:54" and "02/02/2020 15:54", and "Groupe" with "Tout" selected. Below these are two large buttons: a green one labeled "2 261 annonces en situation normale" and an orange one labeled "404 annonces en situation perturbée pro". A red arrow points to the "Date/heure" dropdown. The main area is titled "annonces diffusées" and contains a table with columns: Gare, Date/heure, Catégorie, and Annonce. The table lists four entries:

| Gare                          | Date/heure          | Catégorie | Announce   |
|-------------------------------|---------------------|-----------|--|
| Paris Gare de Lyon Souterrain | 02/02/2020 15:53:14 | SN        | Soyons attentifs ensemble pour la sécurité. Veuillez nous suivre sur les écrans de surveillance. Veuillez nous suivre sur les écrans de surveillance. Veuillez nous suivre sur les écrans de surveillance. |
| Lagny - Thorigny              | 02/02/2020 15:50:50 | SN        | Madame, monsieur, je vous rappelle de bien vouloir tenir éloigné de la bordure des quais.  |
| Changis - Saint-Jean          | 02/02/2020 15:50:49 | SN        | Mesdames, messieurs, je vous rappelle de bien vouloir tenir éloigné de la bordure des quais. Merci de votre attention.   |
| Chézy-sur-Marne               | 02/02/2020 15:50:49 | SN        | Mesdames, messieurs, je vous rappelle de bien vouloir tenir éloigné de la bordure des quais.   |

Filtre par mots-clés

This screenshot shows a user interface for filtering announcements by keywords. At the top, there are three dropdown menus: "Gare" with "Tout" selected, "Catégorie" with "Tout" selected, and "Mots-clés" with "travaux esbly" entered. Below these is a title "Ecoute et Transcription" with a progress bar showing "0" for programmed disturbances and "102" for unexpected disturbances. A red arrow points to the "Mots-clés" input field. The main area displays a table of announcements with columns: Announce, Durée (en s), Débit (en mots/min), and Signaler erreur. The table lists two entries:

| Announce   | Durée (en s) | Débit (en mots/min) | Signaler erreur |
|--|--------------|---------------------|-----------------|
| au 14 février, sauf samedi et dimanche à partir de 21h10, les navettes seront remplacées par des bus. Pour obtenir plus d'informations, je vous invite à consulter le site transilien.com où l'application SNCF.   | 23,05        | 130                 |                 |
| Accès esbly, crécy-la-chapelle mesdames, messieurs. Des travaux vont avoir lieu sur l'axe esbly crécy-la-chapelle du 20 janvier au 14 février, sauf samedi et dimanche à partir de 21h10, les navettes seront remplacées par des bus. Pour obtenir plus d'informations, je vous invite à consulter le site transilien.com où l'application SNCF. | 21,80        | 138                 |                 |

# DEMO

-

# Annonce : création d'un GitHub

Dédié au Club Power BI et à sa communauté



# GitHub pour des ressources communautaires

## ❖ <https://github.com/clubpowerbi>

- Les diapositives des anciens meetups

## ❖ <https://github.com/teamclubpowerbi/>

- Les contributions de la communauté
  - Scripts M
  - Custom visuals

The screenshot shows the GitHub profile for the 'Club Power BI' organization. It features a yellow icon with a bar chart. The repository list includes:

- meetups**: Supports de présentation utilisés lors des meetups du Club Power BI. Updated 4 hours ago.
- customviz**: Visuels personnalisés. Updated 11 days ago.
- M-functions**: Fonctions M pour Power Query. Updated 20 days ago.

On the right, there's a sidebar for 'People' showing 8 members and an 'Invite someone' button.

The screenshot shows the GitHub repository page for 'clubpowerbi' under the 'meetups' branch. It features a yellow icon with a bar chart. The repository details are:

- meetups**: Supports de présentation utilisés lors des meetups du Club Power BI. Updated 4 hours ago.

At the bottom, there's a 'Set status' button and the GitHub logo.

# Place au networking

