



Big Data Viz (and much more!) with
Apache Zeppelin

About me



Bruno Bonnin @_bruno_b_ - 17 juin






Architecte logiciel / Développeur @MyScript

#Java #JavaScript #Python
#Elasticsearch #MongoDB # NoSQL
#Hadoop #Spark #Storm #BigData
#HTML5 #AngularJS #VueJS #NodeJS

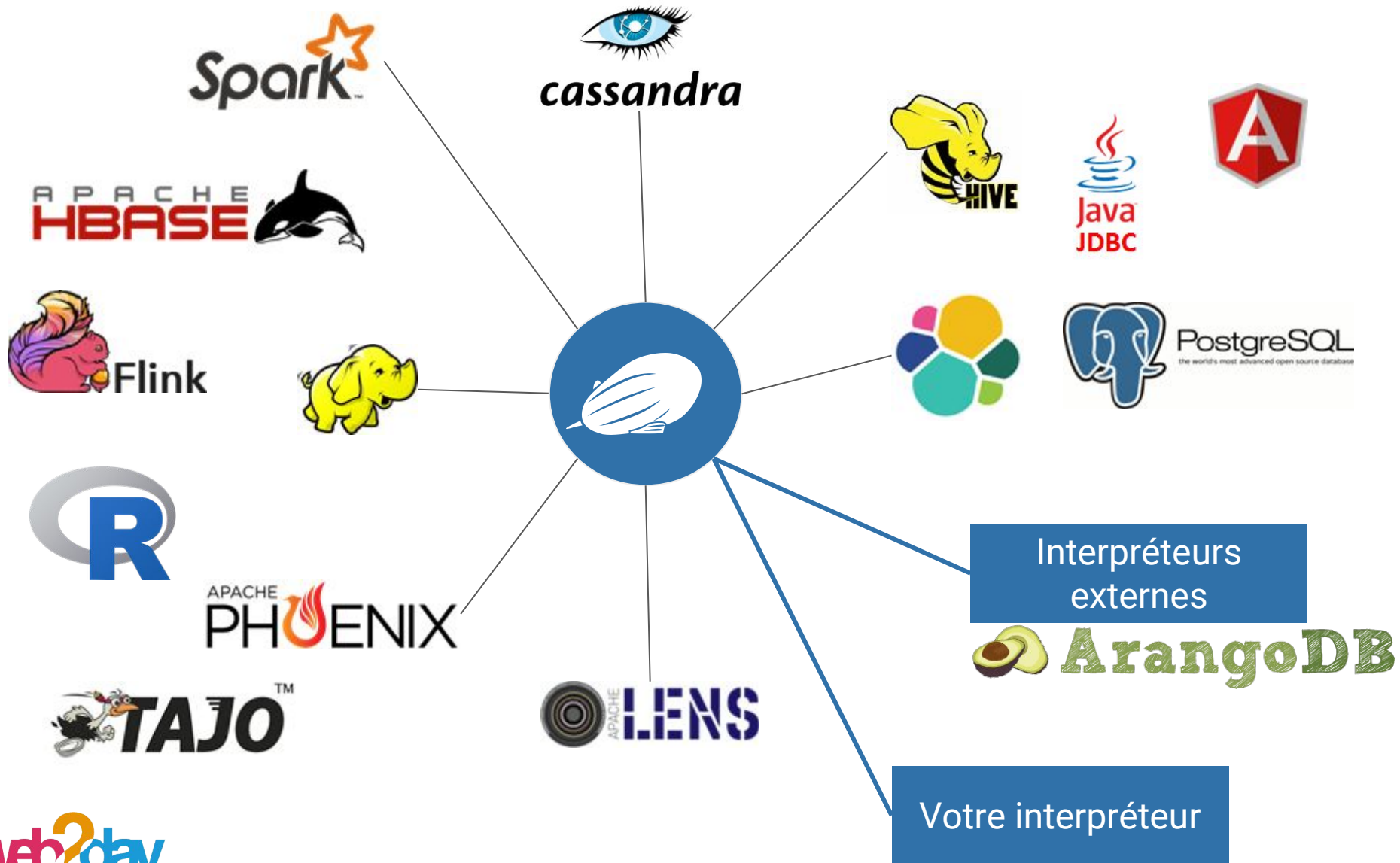


Apache Zeppelin: mais qu'est-ce donc ?

“The one interface for all your big data needs”

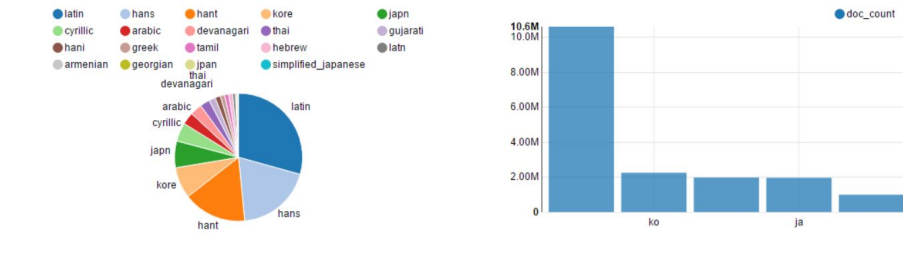
-  Exploration des données...
-  ... avec visualisation graphique
-  Création de documents interactifs...
-  ... facilitant le partage
-  Et tout ça, dans un navigateur !

Apache Zeppelin: multi-languages, multi-backends

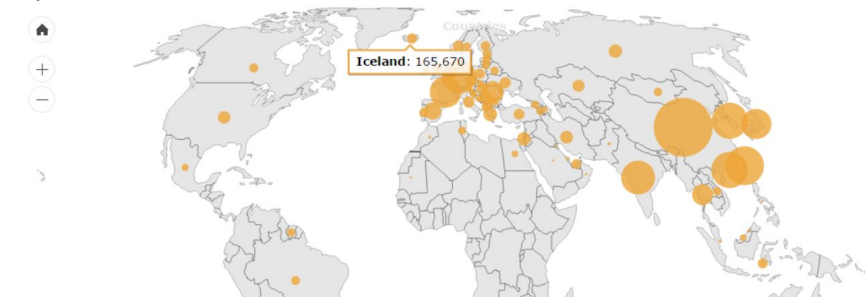


Apache Zeppelin: visualisation graphique

Demo



Pays

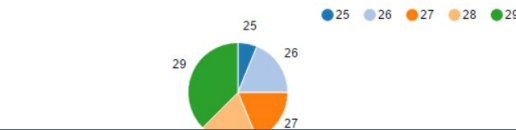
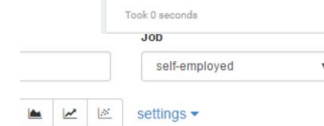


```
select * from bank
where age >= 30 and age < 40
```



ir âge

```
int(1) as cou
age[10-30] or
}
order by age
```



The Zeppelin notebook interface shows a job configuration for a query. The job is named "J00" and is configured to run on a "self-employed" dataset. The settings are displayed in a dropdown menu. Below the job configuration, there is a table of available files:

Fichiers disponibles	Taille
/data/bank.csv	461474
/data/bank-full.csv	4610348

Apache Zeppelin: d'où ça vient ?

- Origine: **NFLabs** (<http://www.nflabs.com/>)
 - Plusieurs essais de faire un env. pour l'analyse de données depuis 2012
 - Au départ, produit commercial
 - Puis, décision de le proposer à la communauté **Apache**
- Projet "incubator" à partir de décembre 2014

Zeppelin est Top Level Project depuis le 25 mai 2016 !



Démo

(<http://localhost:8000/#/>)

Apache Zeppelin: User Interface



Demo #2

Toolbar icons: play, stop, refresh, save, share, and others.

Help, settings, and user profile icons.

Paragraphe

Répartition selon âge

```
%jdbc
SELECT age, count(1)
FROM bank
WHERE age < ${maxAge=30}
GROUP BY age
ORDER BY age
```

Interpréteur utilisé (sql, spark, sh, md, jdbc, ...)

FINISHED

Texte à interpréter (SQL, Spark, sh, html, ...)

Toolbar du paragraphe (start, ...)

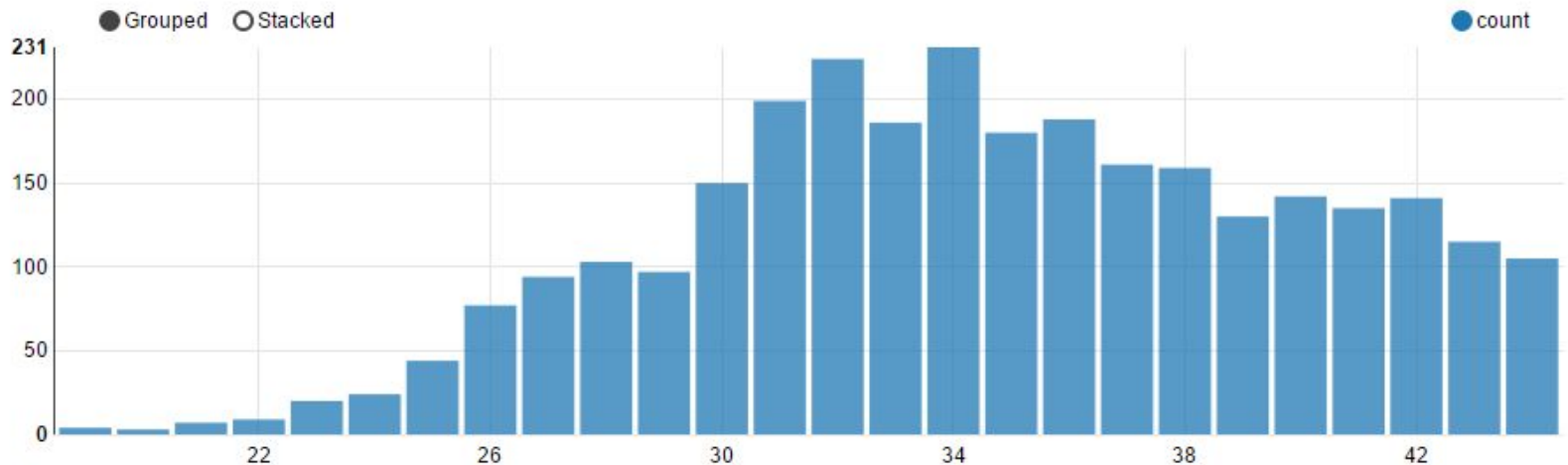
maxAge

45

Formulaire généré à partir de la requête

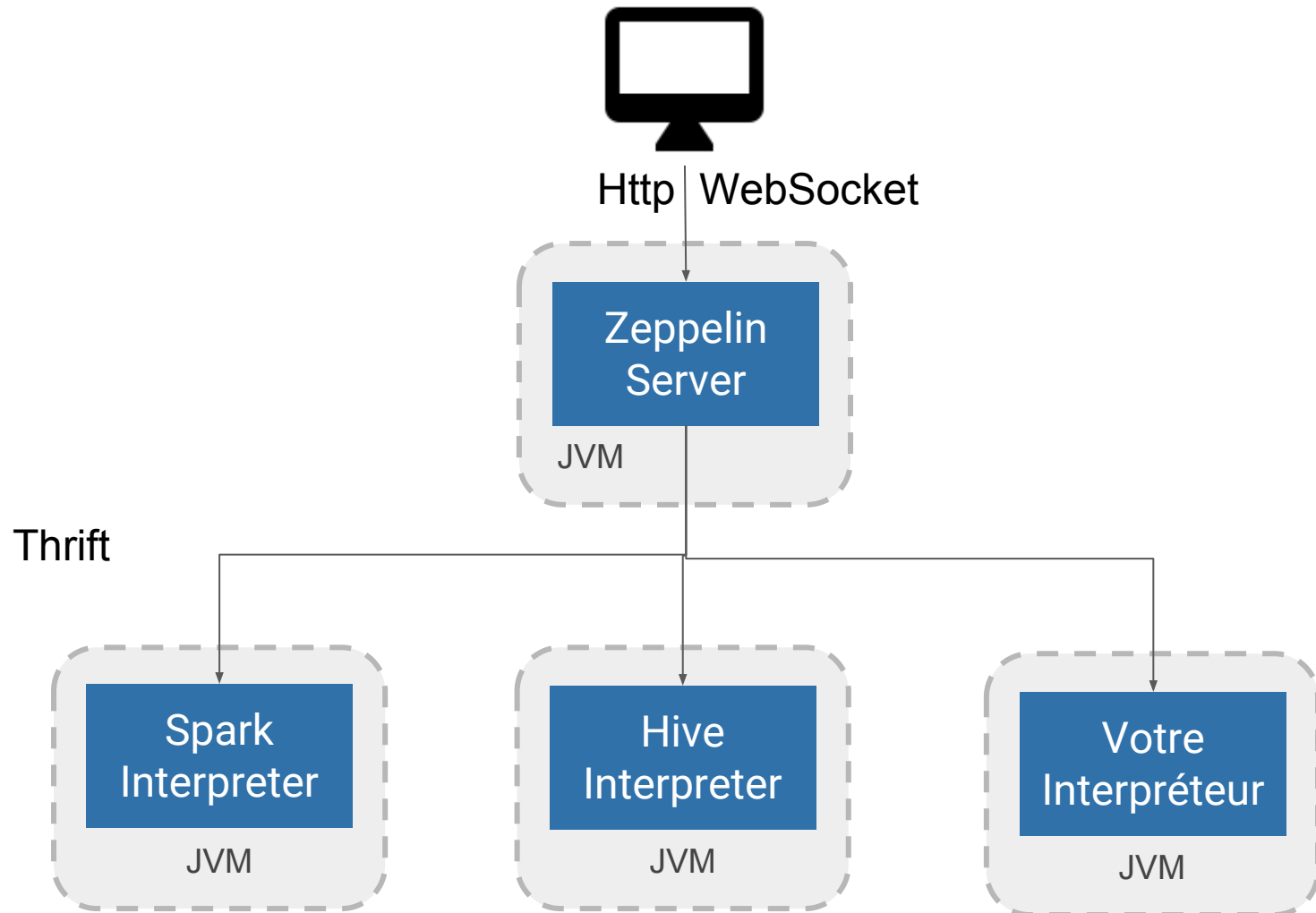


Choix de l'affichage

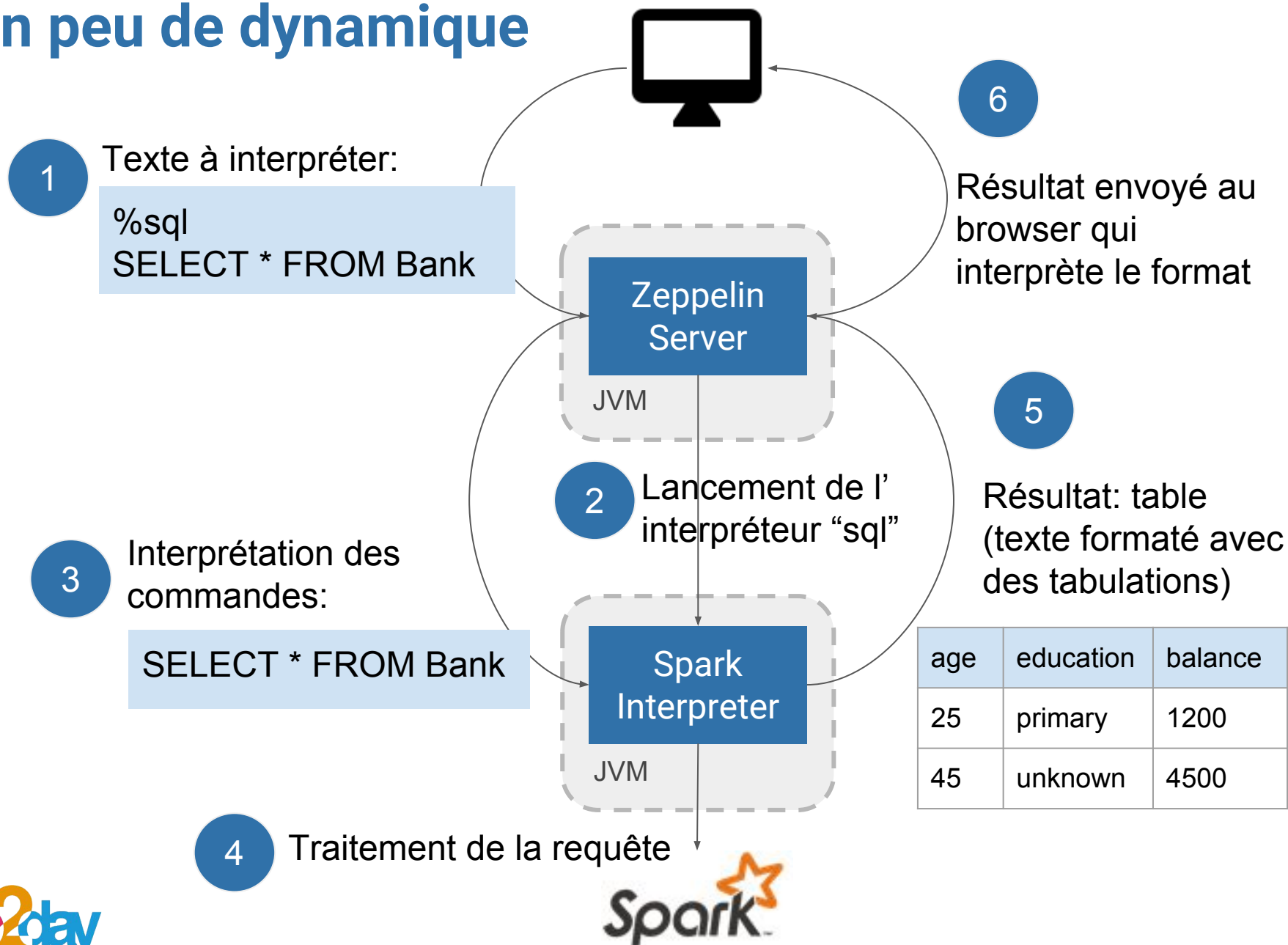


Résultat

Architecture



Un peu de dynamique



Systèmes d'affichage: la base

Il existe plusieurs systèmes d'affichage:

- **texte**: output par défaut, sans formatage particulier
- **html** : si l'output commence par **%html**
 - Affichage du html contenu dans la sortie
- **table** : si l'output commence par **%table**
 - La sortie doit contenir des lignes avec chaque cellule séparée par des `\t`
 - **Accès automatiquement à des visualisations graphiques** de vos données dans ce cas !!

Le format peut être émis par l'interpréteur (exemple: SQL)

Display: text

FINISHED ▶ 🗖 ⚙

```
%sh
echo 'Fichiers disponibles:'
stat -c "%n %s" /data/tripadvisor/json/993*.json
```

```
Fichiers disponibles:
/data/tripadvisor/json/99302.json 621797
/data/tripadvisor/json/99307.json 1480367
/data/tripadvisor/json/99321.json 287815
/data/tripadvisor/json/99332.json 646289
/data/tripadvisor/json/99333.json 330618
/data/tripadvisor/json/99357.json 215297
/data/tripadvisor/json/99365.json 142078
/data/tripadvisor/json/99368.json 225732
/data/tripadvisor/json/99371.json 271849
/data/tripadvisor/json/99387.json 162388
```

Display: html

FINISHED ▶ 🗖 ⚙

```
%sh
echo '%html <table class="table table-striped table-condense
d"><tr><th>Fichiers disponibles</th><th>Taille</th></tr>'
for csv in $(ls /data/tripadvisor/json/993*.json)
do
  csv_size=$(stat -c%s "$csv")
  echo "<tr><td>$csv</td><td>$csv_size</td></tr>"
done
echo "</table>"
```

Fichiers disponibles

Taille

/data/tripadvisor/json/99302.json	621797
-----------------------------------	--------

Display: table

FINISHED ▶ 🗖 ⚙

```
%sh
echo "%table"
echo -e "Fichiers disponibles\tTaille"
for csv in $(ls /data/tripadvisor/json/993*.json)
do
  csv_size=$(stat -c%s "$csv")
  echo -e "$csv\t$csv_size"
done
```



Fichiers disponibles

Taille

/data/tripadvisor/json/99302.json	621,797
/data/tripadvisor/ison/99307.json	1.480.367

Systèmes d'affichage: formulaires

Génération automatique de formulaires (input text, select) si présence de **`${...}`** dans le code à interpréter

%sql

```
SELECT * FROM hotels WHERE avgRating >= ${note_minimum} AND country = '${country=France,England|France|Italy|Spain|USA}'
```

FINISHED ▶ ⌵ 📖 ⚙

country

Italy ▼

note_minimum

4.5



name	street	locality	postalCode	region	country	avgRating	room	cleanliness
UNA Hotel Venezia	Ruga Do Pozzi 4173	Venice	30,121		Italy	4.538461538461538	4.641025641025641	4.8023255813953485

Systèmes d'affichage: Angular

Permet d'avoir accès à des formulaires et des affichages plus évolués/sympa/... (mais ça demande du code)

- Côté client: il existe un interpréteur "angular"
 - Le paragraphe contient le code html/javascript pour interpréter pour réaliser l'affichage du paragraphe
- Zeppelin fournit aussi une API permettant de partager des données entre les paragraphes Spark et Angular, accessible via une variable : **Z**
 - `z.angularBind (nom, valeur)`
 - `z.angularUnbind (nom)`
 - `z.angular(nom)`
 - `z.runParagraph(id paragraphe)`

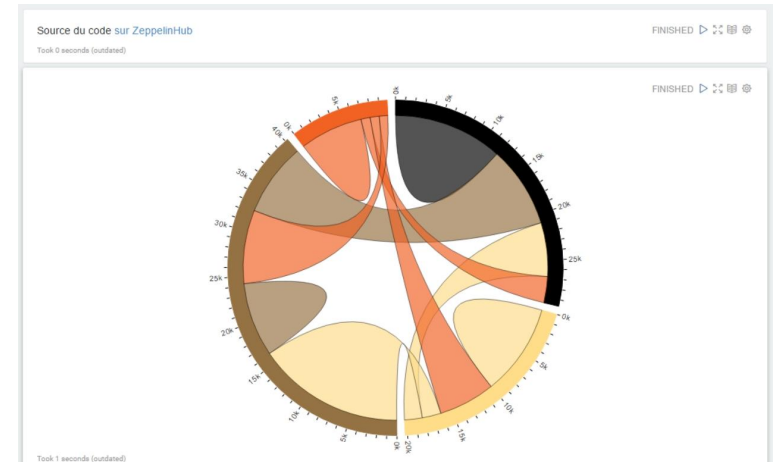


Démo

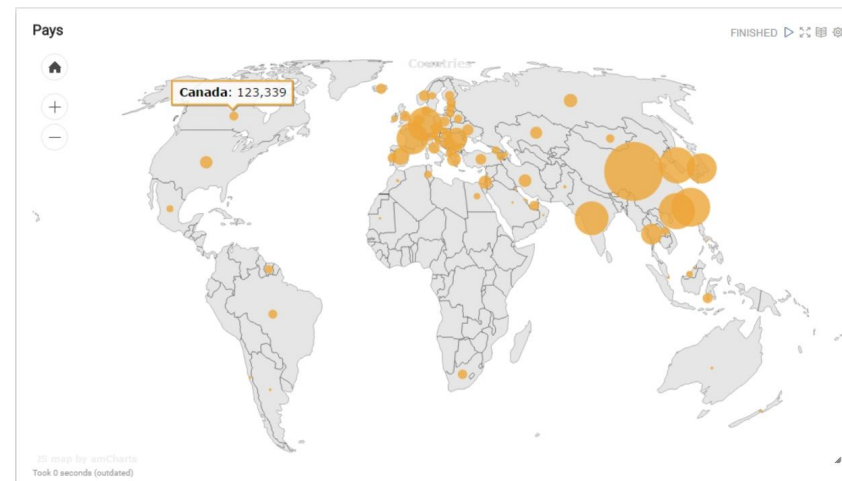
(<http://localhost:8000/#/notebook/2BNCV88E6>)

Systèmes d'affichage: extensions

1 Zeppelin embarque **D3.js**, on peut donc l'utiliser (en codant un peu) pour proposer des visualisations top moutantes !



2 Libs graphiques externes: référencées dans le code (<script src="">)





Démo

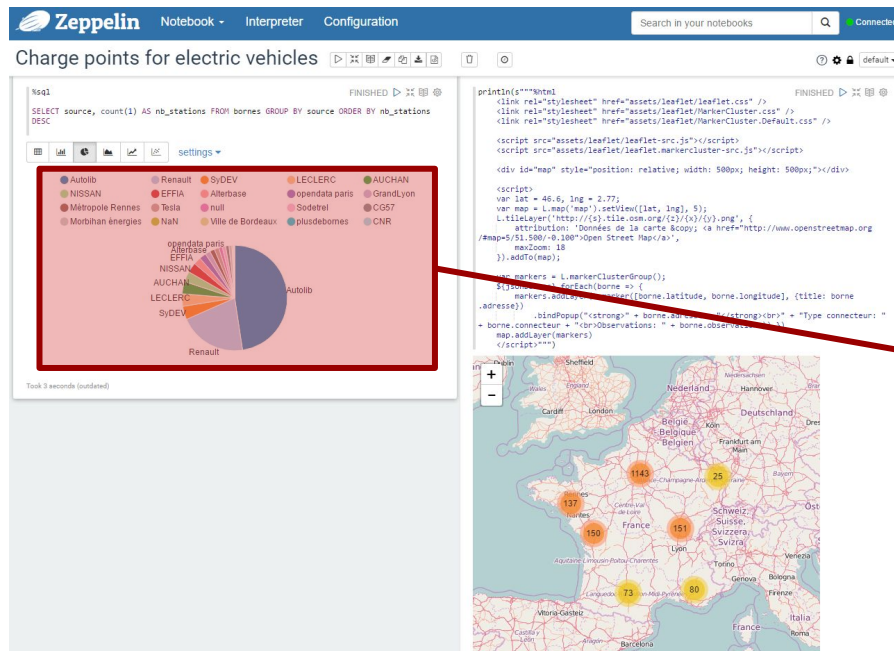
(<http://localhost:8000/#/notebook/2BMXSE5D2>)

Publication des résultats

Les résultats peuvent être réutilisés dans d'autres pages (via `<iframe>`)

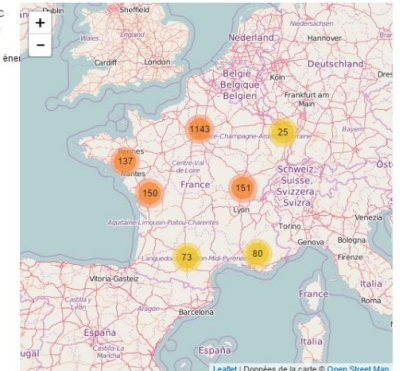
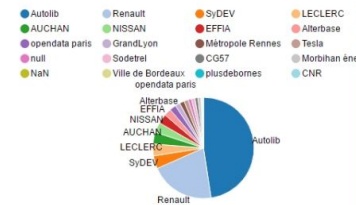
Analyse et construction dans Zeppelin

Intégration dans votre site



Charge points: Providers

Charge points: Location



Étendre Zeppelin: développer son interpréteur

```
public class MyInterpreter extends Interpreter {
    static {
        Interpreter.register("interp_name", "interp_group", MyInterpreter.class.getName());
    }

    public InterpreterResult interpret(String cmds, InterpreterContext ctx) {
        String result = ...
        return new InterpreterResult(ResultCode.SUCCESS, Type.HTML, result);
    }

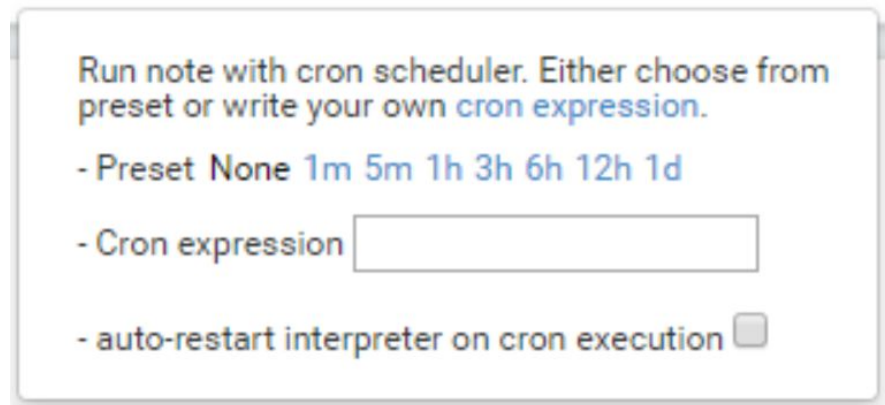
    public void open() {}
    public void close() {}

    public void cancel(InterpreterContext ctx) {}
    public int getProgress(InterpreterContext ctx) {}

    public List<String> completion(String cmd, int i) {}
    public FormType getFormType() {}
}
```

Apache Zeppelin: y a quoi d'autres ?

- **Scheduler**



Run note with cron scheduler. Either choose from preset or write your own cron expression.

- Preset None 1m 5m 1h 3h 6h 12h 1d

- Cron expression

- auto-restart interpreter on cron execution ☐

- Export / Import

- Gestion de versions

- **Sécurité**: indispensable pour passer du stade de PoC à un vrai système en prod (mais ça vient toujours en dernier)
 - Authentification avec Shiro, Autorisation au niveau Notebook, ...

En résumé...

Zeppelin, c'est:

- Open source (<https://zeppelin.apache.org/>)
- **Ouvert** (on peut l'adapter à ses besoins, son contexte, via le dev d'interpréteurs, l'utilisation de libs pour les visualisations)
- **Plein de fonctionnalités** déjà présentes ou à venir:
 - Nouveaux interpréteurs (scalding,...)
 - Visualisation de maps
 - Améliorations internes / UI
 - ...

Votre futur environnement pour vos futurs besoins autour de vos futurs (méga-) données

Merci !

 @_bruno_b_

<https://github.com/bbonnin/web2day2016>

Liens Zeppelin

Site officiel:

- <https://zeppelin.incubator.apache.org/>

Docs:

- <https://zeppelin.incubator.apache.org/docs/latest/>

Code source:

- <https://github.com/apache/incubator-zeppelin>

Exemples:

- <https://www.zeppelinhub.com/viewer>

Exemple d'interpréteur: ArangoDB Interpreter

- <https://github.com/bbonnin/zeppelin-arangodb-interpreter>

Source des données

Bornes de recharge:

- <https://www.data.gouv.fr/s/resources/fichier-consolide-des-bornes-de-recharge-pour-vehicules-electriques-irve/20151008-182813/IRVE-201510.csv>

NASA:

- <https://data.nasa.gov/>

TripAdvisor:

- <http://times.cs.uiuc.edu/~wang296/Data/>

Bank:

- <http://archive.ics.uci.edu/ml/machine-learning-databases/00222/bank.zip>

Exemple D3.js

- <https://www.zeppelinhub.com/viewer/notebooks/bm90ZTovL2Rjb3JuZWV1L1BlcnNvbWFsLU5vdGVib29rcy80NzU5L25vdGUuanNvbG>