

TER-12 : Outil pour exploration d'une base de connaissance musicale

El motacime Omar - Ewen Hanni -
Ivo Costa Cunha - Wajdi Gaiech



Encadrants : Michel BUFFA,
Aline MENIN, Maroua TIKAT

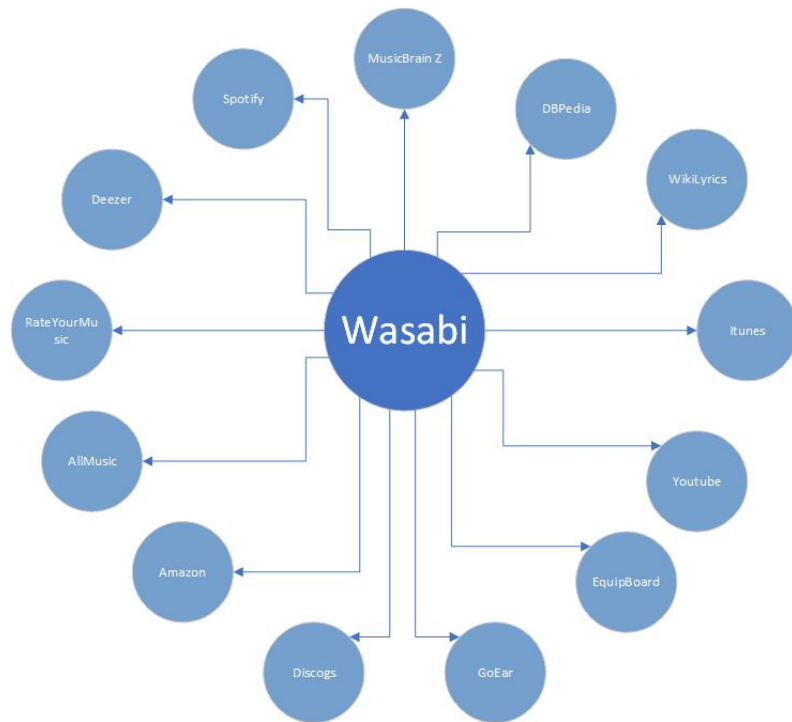
Responsable : Adrien Richard

Sommaire:

- Sujet et problématiques
- Démonstration
- Technologies utilisées
- Notre approche
- Récupération de données
- Présentation des fonctionnalités
- Organisation du projet
- Conclusion

Sujet et problématiques

- Volume de données: 2M de chansons
- Données intégrées à partir de plusieurs sources sur le web + à partir de l'analyse audio + à partir de l'analyse des paroles



Data quality problems categorization : WASABI

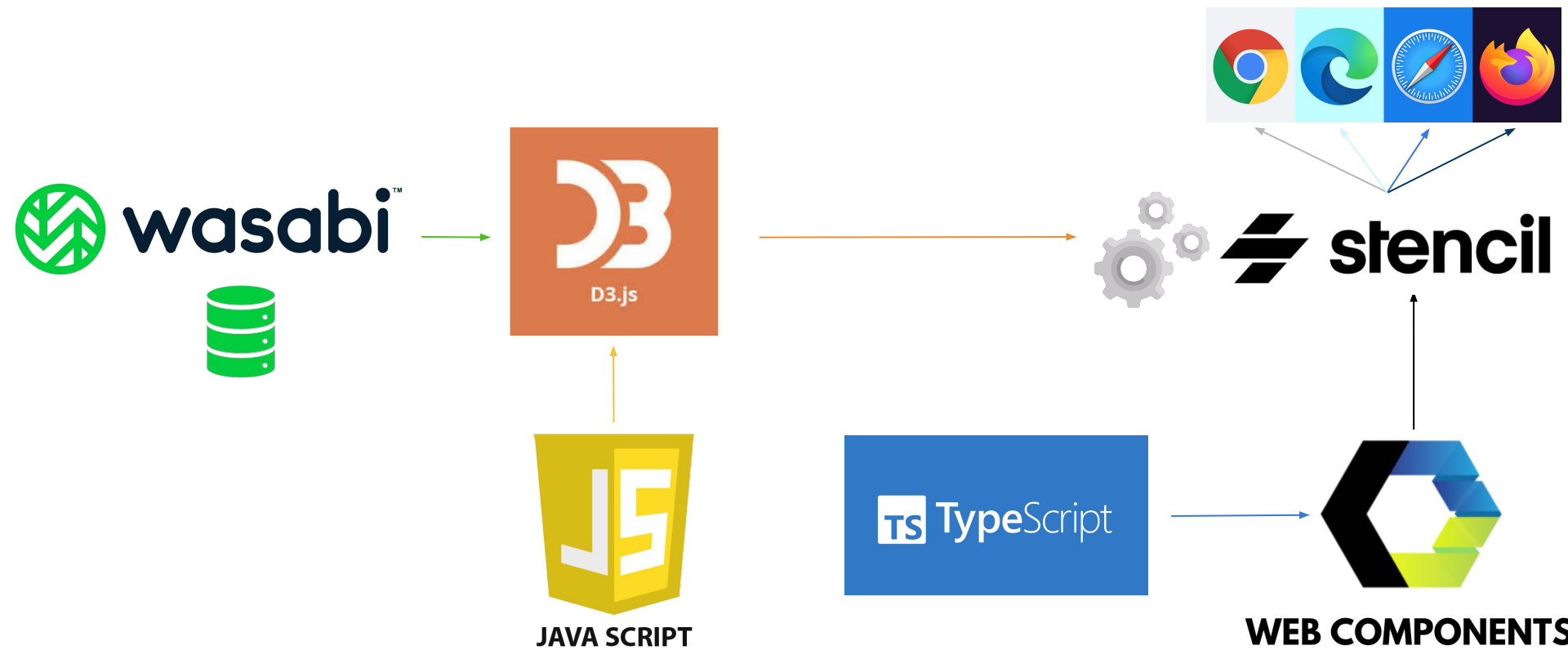
Laranjeiro, Nuno, Seyma Nur Soydemir, and Jorge Bernardino. "A survey on data quality: classifying poor data." 2015 IEEE 21st Pacific rim international symposium on dependable computing (PRDC). IEEE, 2015.

In **WASABI**, we detected data quality issues related to all quality dimensions; **Accessibility, accuracy, completeness, consistency, currency**

Problem types		Data quality problems	Accessibility	Accuracy	Completeness	Consistency	Currency
Source	Level						
Single	Instance	Missing data		•	•		
		Incorrect data		•			
		Misspellings					
		Ambiguous data	•	•			
		Extraneous data	•				
		Outdated temporal data		•			•
		Misfielded values	•	•	•	•	
		Incorrect references		•			
		Duplicates	•				
	Schema	Domain violation		•			
		Violation of functional dependency		•			
		Wrong data type	•			•	
		Referential integrity violation	•	•	•	•	
		Uniqueness violation		•			
Multiple	Instance	Structural conflicts	•			•	
		Different word orderings	•	•		•	
		Different aggregation levels	•	•		•	
		Temporal mismatch		•		•	•
		Different units	•			•	
		Different representations	•	•		•	
	Schema	Use of synonyms	•				
		Use of homonyms	•				
		Use of special characters	•				
		Different encoding formats	•			•	

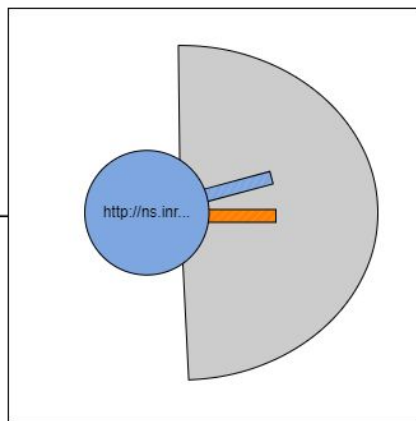
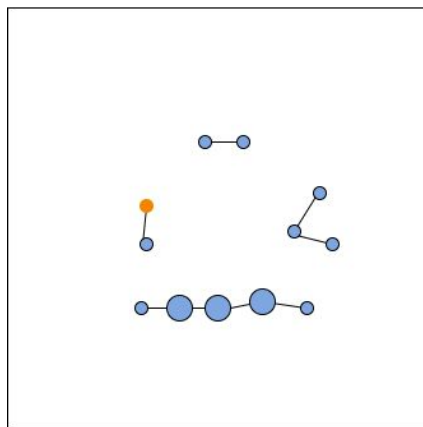
Démonstration

Technologies utilisées



Notre approche : maquettes

Données manquantes



Titres
inconnu

date publication 4
inconnu

lien:

Attention, cette/ces chanson(s) semble(nt) avoir des données manquante

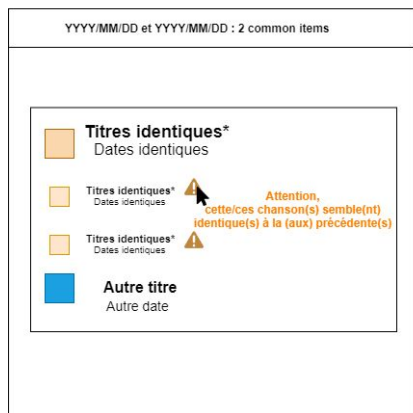
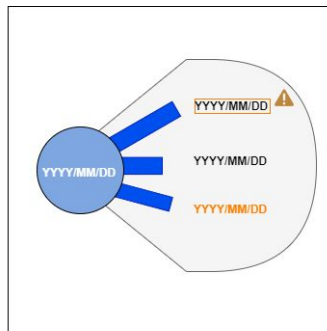
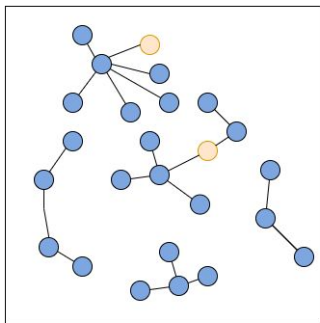
Titres
inconnu

date publication 4
inconnu

lien:

Attention, cette/ces chanson(s) semble(nt) avoir des données manquante

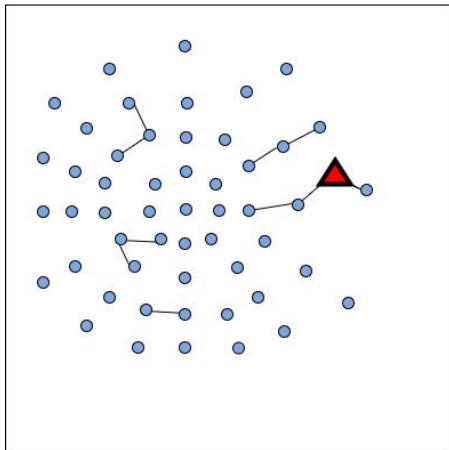
Notre approche : maquettes



*Ou quasi identiques, ils correspondent à la même chanson.

Duplications

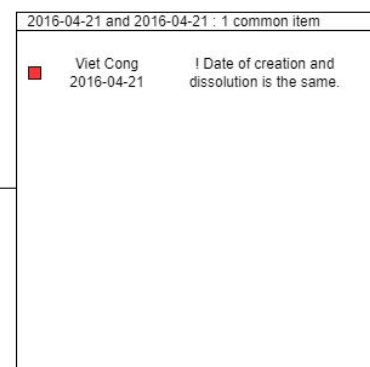
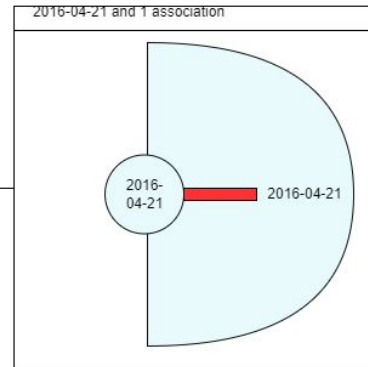
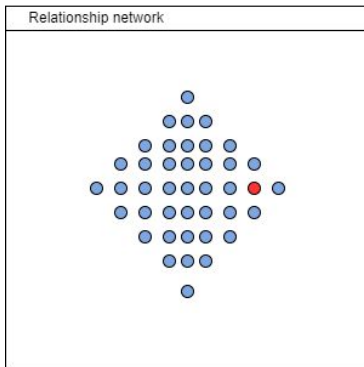
Notre approche : maquettes



Il s'agit ici d'une reprise. Les deux artistes n'ont pas collaboré car un des deux n'est plus en activité (Michael Jackson qui est mort et Justin Timberlake qui reprend une de ses œuvres).

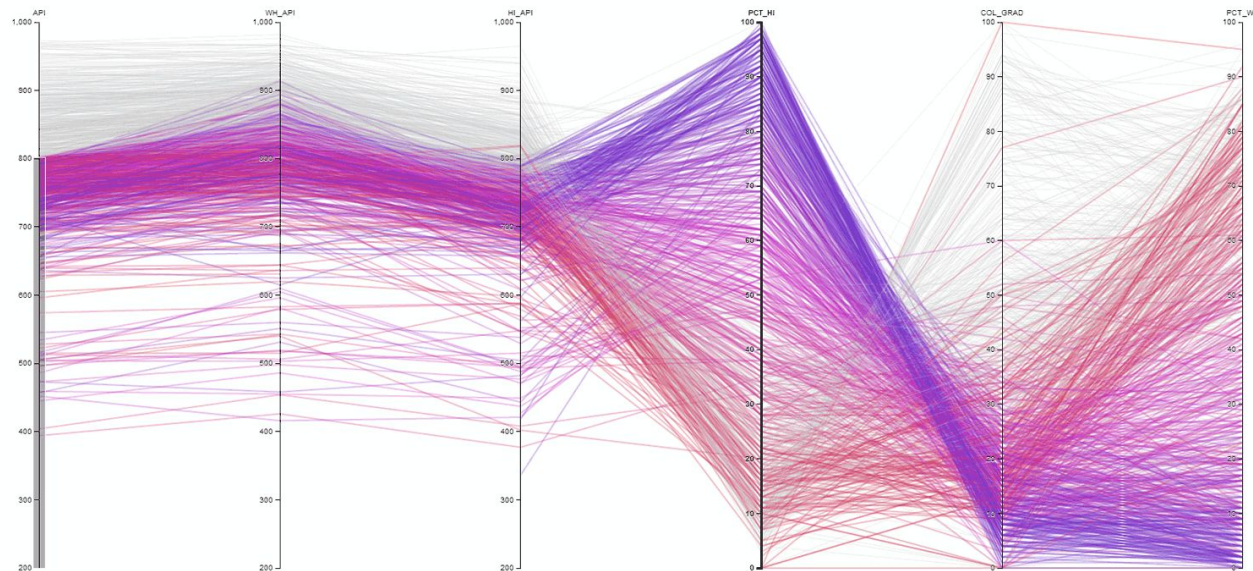
Les reprises seront marquées par un triangle rouge. Le triangle est ici sélectionné ce qui va nous donner plus d'information sur la reprise

Incohérences



Notre approche : l'heure du choix

Choix final :

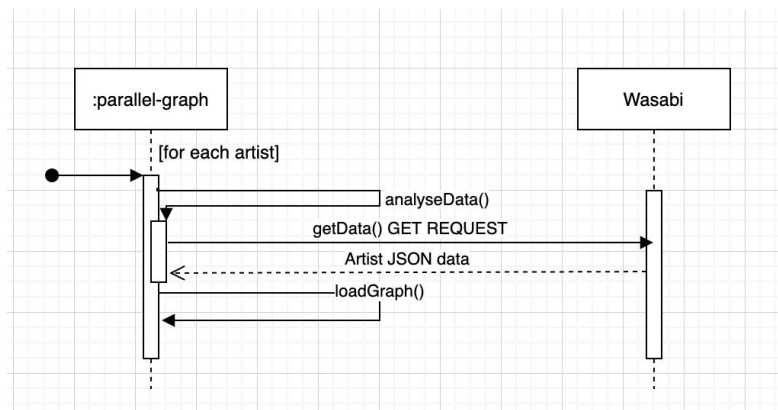


Graphique de coordonnées parallèle

Récupération et traitement de donnée

A l'initialisation du web component

On composantDidLoad()



Formulaire de recherche d'artiste

```
render() {  
  return (  
    <Host>  
      <div class="search-zone">  
        <h1>Parallel graph</h1>  
  
        <form onSubmit={(e) => this.handleSubmit(e)}>  
          <label>  
            Search artist name: <input class="input-search" type="text" value={this.newArtist} onChange={(event) => this.handleChange(event)} />  
          </label>  
          <input type="submit" value="Search" />  
        </form>  
        <div class="log"> No problems </div>  
        <div class="current-artist" >Current artist : {this.currentArtist}</div>  
        <h2>Stats (TODO) </h2>  
        <p>Missing data : </p>  
      </div>  
  
      <div class="tool">  
        <svg class="chart" />  
      </div>  
      <div class="add">  
      </div>  
    </Host>  
  )  
}
```

Diagramme de séquence du formulaire

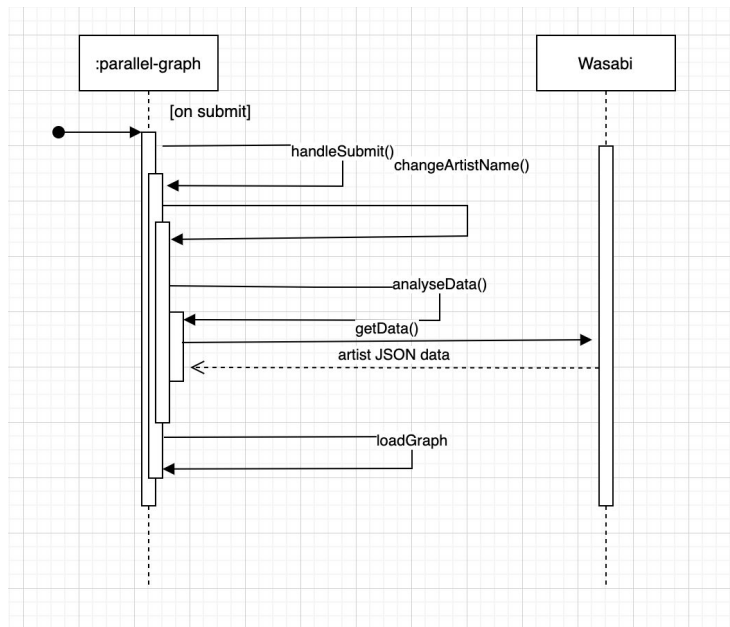
A chaque recherche d'artiste

Variables globales

```
@State() newArtist: string;
@State() currentArtist: string;

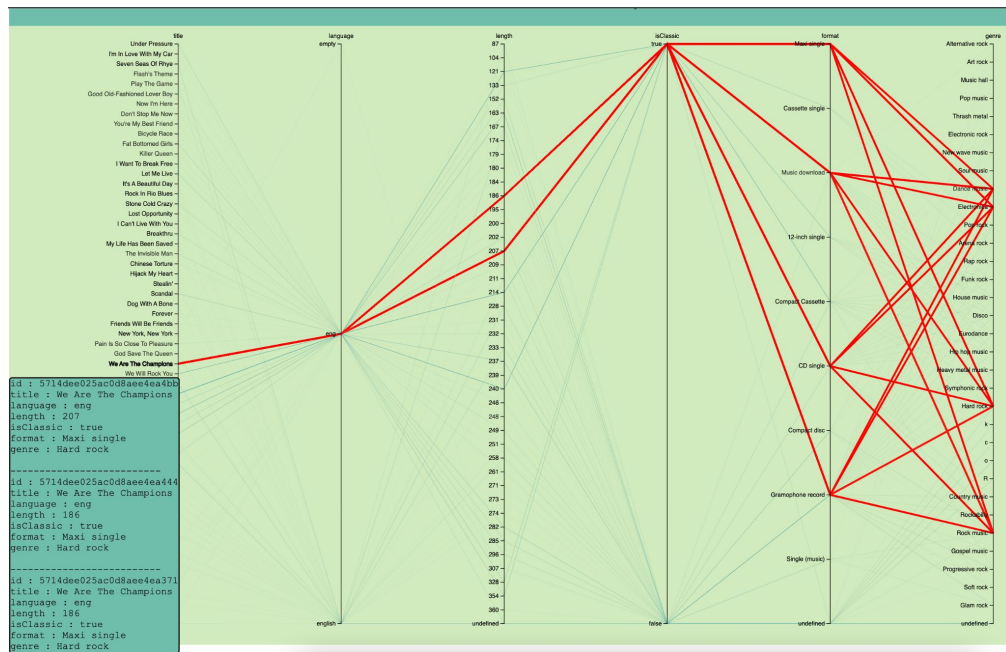
// Constrain valuable information to use in the parallel graph
@State() dataObj : { artist: string; } = { artist: "" };
@State() dataObjInt;
@State() svg;
@State() data1;
@State() dimensions;
@State() categories;
@State() group;
@State() dropdownButton;
public chartData: any;
```

On form submit



Fonctionnalités de la visualisation

- Afficher des valeurs manquantes
- Afficher les multiples valeurs d'une chanson (format, genre)
- Survol des attributs
- Tooltip affichant les informations détaillées sur une chanson
- Afficher dynamiquement les données d'un artiste donné à partir de Wasabi
- Ajouter des attributs dynamiquement
- Afficher les différentes valeurs d'un attribut
- Permuter les attributs sur le graphique



Organisation du projet

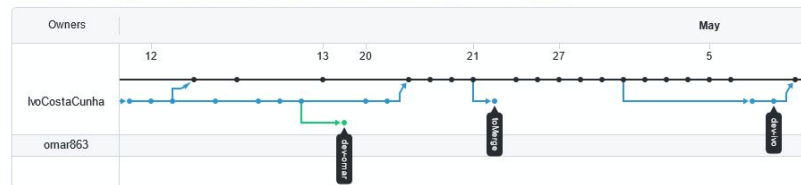
Tâche / Semaine	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Introduction web components															
Introduction shadow dom															
Introduction stenciljs															
Intro à D3JS															
Introduction au projet															
Introduction au type d'erreurs + determiner une bonne visualisation															
Mise en place des requêtes à Wasabi															
Mise en place d'une version initiale du parallel-graph															
Récuperer les données spécifiques à certains attributs des chansons															
Ajouter au graphe ces attributs															
Repliquer les données afin de tracer correctement les lignes sur le graphe															
Mettre en evidence une chanson sur le graphe															
Corrections sur les données															
Trier le graphe															
Requêtes dynamiques à Wasabi															
Création d'un tooltip qui affiche les données d'une chanson															
Rajouts dynamiques d'axes															
Finalisation															
Redaction rapport															

Début de projet	21/02/2022
Fin de projet	06/06/22



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Conclusion

Merci de votre attention !