Université de Pau et des pays de l'Adour

Rapport projet DMN

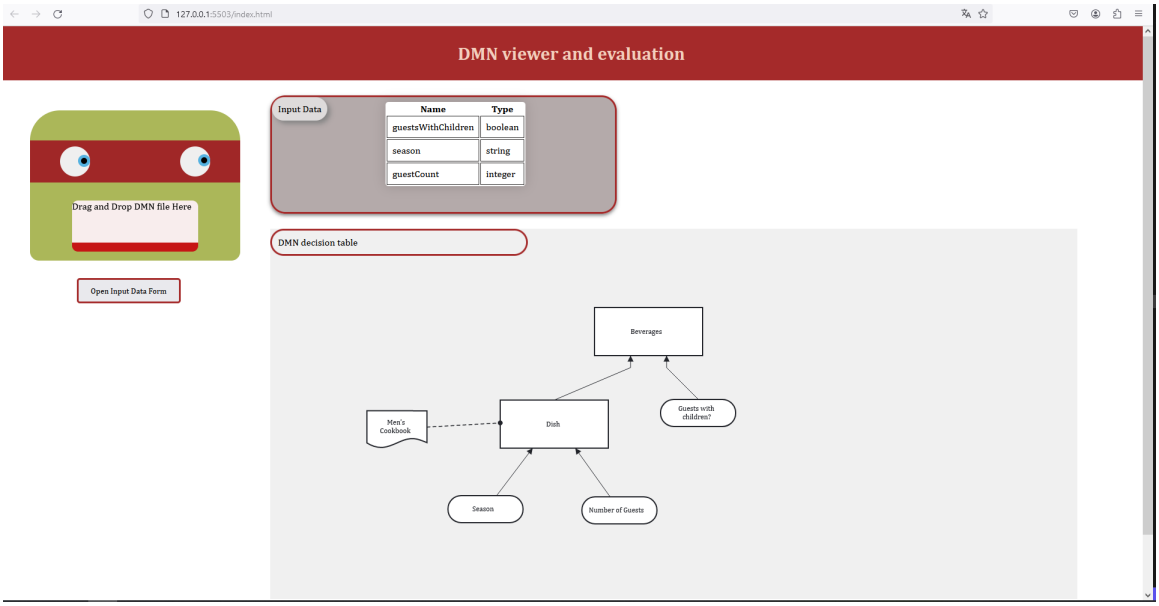
Author:
Corentin Calmels

Décembre 2023

Contents

1 Introduction

L'objectif de ce projet était de pouvoir afficher et évaluer un fichier DMN. Ce rapport explique l'architecture du projet ainsi que les choix de conception, les fonctionnalités principales, et les librairies utilisées pour atteindre ces objectifs. Le code source est également disponible ici C21111222/DMN_project.



2 Organisation du code

2.1 Structure des dossiers

Le projet est constitué de plusieurs dossiers et sous-dossiers, suivant la structure suivante:

- `ts/`: Le dossier contenant tous les fichiers Typescript.
  - `animation/`: Fonctions d'animation de la page (e.g., $\texttt{eyes}_{m}ovements)\texttt{models/}$: $Classesdemodèlesdedonnées(e.g., \texttt{DMNModel})$
  - `utils/`: Fonction et classes utilitaire (e.g., $\texttt{migrate}_{D}MN)$
  - `view/`: Classes et fonctions pour l'affichage des données (e.g., `DataDisplay`)
  - `main.ts`: Fichier main du projet, importation des différentes fonctions et données, prise en charge des fichiers drag and drop.
- `css/`: Dossier pour les fichiers css
- `@types/`: Dossier pour la déclaration de types Typescript nécessaire au build
- `DMN_file`: Dossier contenant les fichiers DMN d'exemple

2.2 Convention de nommage

- **Fichier**: Snake-case (e.g., `decision_table.ts`).
- **Classes & Interfaces**: PascalCase (e.g., `DMNModel`).
- **Méthodes & Variables**: camelCase (e.g., `defineInputData`).
- **Constantes**: UPPER_SNAKE_CASE (e.g., `MAX_TIMEOUT`).

2.3 Détail des classes et fonctions

Le détail des classes, leurs méthode ainsi que les fonction du projet sont consultable dans la documentation, pour cela se référer à la section associé dans "Installation et utilisation" .

3 Librairies et Frameworks

Le projet se base sur plusieurs librairies pour fonctionner:

- **DMN-JS**: Regroupe des objets JavaScript pour "lire" les fichiers DMN
- **SweetAlert2**: Permet d'affciher simplement des notifications.
- **Xmldom** Permet de parser des fichiers xml
- **DMN-Migrate**: J'ai utiliser cette librairie (lien github) en l'adaptant en TypeScript, permettant la migration de fichier DMN en version 1.3, à partir de versions antérieures.
- **Feelin**: Permet d'évaluer les expressions du langage feel, qui correspond aux règles dans un fichier DMN.
- **TypeDoc**: Permet de générer une documentation.

4 Fonctionnement général

Le principe de l'application est simple : on fait glisser et déposer un fichier DMN. On peut ainsi le visualiser et l'évaluer. Ensuite, on peut observer les données en sortie.

4.1 Fonctionnalités annexes

- **Migration**: Grace à la librairie migrate-dmn l'application permet de prendre en charge les fichiers DMN en version 1.1 et 1.2.
- **Visualisation des sous-tables**: Lorsque le fichier MDN contient plusieurs décisions, l'affichage correspond à l'ensemble de celles-ci. Pour visualiser en détail les sous-tables, on peut cliquer sur l'objet le représentant dans le diagramme.
- **Formulaire**: Pour évaluer un fichier DMN, on peut au choix déposer un fichier JSON contenant les données requises, soit utiliser le formulaire généré automatiquement en cliquant sur "Open Input Data form".
- **Documentation**: La génération de la documentation de chaque fichier est disponible dans

5 Installation et utilisation

5.1 Pré-requis

- **npm (Node Package Manager)**: Le gestionnaire de paquets Node.js est nécessaire pour le téléchargement des dépendances et la construction de l'application.
- **Live serveur**: Un live serveur est nécessaire pour ouvrir le fichier HTML du projet.

5.2 Build

1. (depuis github uniquement) Cloner le dépot.
2. Accédez au répertoire du projet.
3. Installez les dépendances nécessaires en exécutant "npm install".
4. Construisez le projet en exécutant "npm run build".

5.3 Utilisation

1. Ouvre index.html avec un live serveur dans un navigateur web moderne.
2. Glissez et déposez un fichier DMN dans la zone prévue.
3. Consultez la table de décision et les données d'entrée.
4. Évaluez la logique de décision en fournissant un json ou en utilisant le formulaire fourni
5. Observez la sortie.

5.4 Documentation

1. Accédez au répertoire du projet.
2. Exécutez npm run doc.
3. Ouvrez docs/index.html dans un navigateur web.

6 Limites

Plusieurs points sont à améliorer dans ce projet;

- Gestion des erreurs incomplète
- Prise en charge des fichiers DMN aberrants, par exemple dans un fichier comportant plusieurs décisions non reliées (flight rebooking notamment)
- Robustesse du code