Mathias JANIN

Azad LUCAS

BUT SD 2

Projet Rshiny

ADEME x BAN

Documentation technique (orientée développeur)

+

Documentation fonctionnelle (orientée utilisateur)

Année universitaire 2024-2025

Documentation technique :

Pour mener à bien ce projet, nous avons eu besoin de 2 tables pour construire notre structure relationnelle. La première qui s'appelle « adresses-55 » et qui contient les informations utiles des logements du département de la Meuse (code postal, nom de commune, coordonnées géographiques, ...) disponible grâce à la Base des Adresses Nationales « BAN » (https://adresse.data.gouv.fr/) qui met à disposition les données de chaque département français.

Pour créer la deuxième, nous avons dû découper le travail en deux. Grâce aux données disponibles sur l'API de l'ADEME (https://data.ademe.fr/) concernant les données des logements neufs et existants, nous avons pu créer deux bases de données temporaires nommées « existants_55 et neufs_55 ». Celles-ci ont pu nous apporter des informations supplémentaires concernant les DPE des logements concernés (ici ceux de la Meuse). Une fois cela fait, nous avons liés simplement les deux (grâce à la fonction rbind()) pour créer la base de données nommée « df_logement ».

Ainsi, grâce à ces deux tables (et la fonction merge()), nous avons pu lier les deux tables pour obtenir la base de données finale qui allait nous être utile pour créer notre fichier RMarkdown et l'application RShiny. Celle-ci se nomme très simplement « df_final ».

Explication du code :

Concernant l'application RShiny, nous avons opté pour un code unique qui contient le Server, c'est-à-dire le code pour faire l'application, ainsi que le visuel (l'UI).

Pour lancer l'application, les packages sont d'abord installés. Ici, ils ne le sont que s'ils ne sont pas déjà présents. La fonction « if » permet de vérifier si les packages ne sont pas déjà installé sur l'ordinateur et cela permet donc de gagner du temps sur l'exécution du code. Les packages sont donc ensuite chargés sur R Studio. Ensuite, nous avons mis le code pour obtenir la base de données « df_final » qui contient toutes les données afin de la charger. Suite à quoi en découle tout le code pour exécuter le serveur et l'interface utilisateur.

Les packages:

shiny : Créer des applications web interactives en R.

shinythemes : Ajoute des thèmes personnalisés à une appli Shiny.

ggplot2 : Génère des visualisations de données avec syntaxe puissante et flexible.

DT: Créer des tableaux interactifs et dynamiques dans Shiny.

dplyr: Manipule et transforme efficacement des données structurées.

leaflet: Génère des cartes interactives directement dans R.

shinyWidgets: Fournit des widgets supplémentaires et personnalisés pour Shiny.

base64enc: Encode et décode des données en base64, surtout pour des images.

httr: Gère les requêtes http pour interagir avec des API.

jsonlite: Pour la conversion entre les données R et le format JSON.

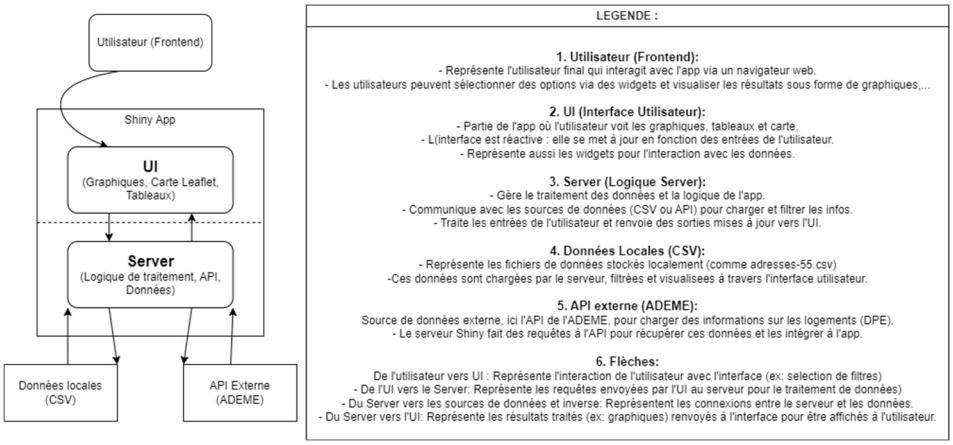
Pour en savoir plus sur le code :

L'interface utilisateur est structurée grâce à navbarPage(), qui permet une navigation fluide entre plusieurs onglets. La page « Accueil » présente un aperçu du département et propose un mode nuit (activé via switchInput()).

Concernant les graphiques, ils permettent de visualiser des données sous forme de graphiques interactifs, grâce au package ggplot2. Les utilisateurs peuvent filtrer par classe DPE, type de logement, et télécharger les résultats sous forme de PNG. Pour la carte, on utilise leaflet pour afficher une carte interactive des logements, avec des marqueurs colorés selon leur classe énergétique. La fenêtre contexte possède un tableau interactif permettant de visualiser l'ensemble des données avec des filtres avancés. Les utilisateurs peuvent exporter ces données en CSV via un bouton de téléchargement.

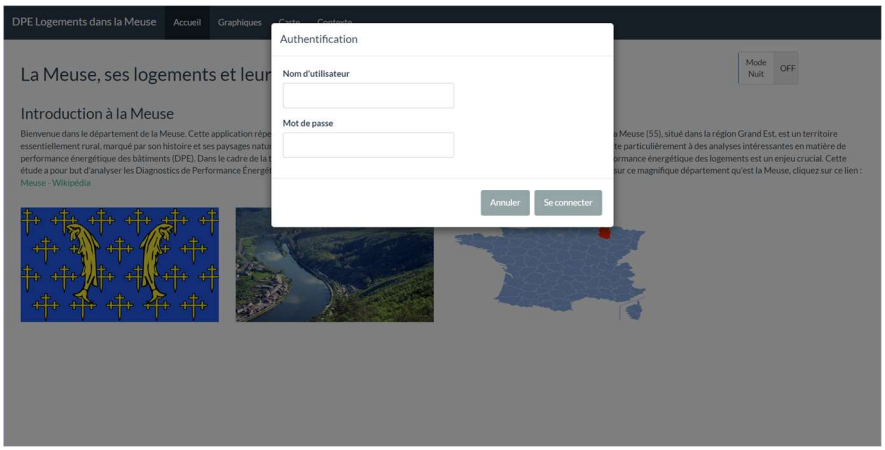
La partie serveur, gérée par server(), orchestre toutes les interactions et mises à jour dynamiques. Les utilisateurs doivent d'abord s'authentifier grâce à une simple vérification de username et password. (username : asardell ; password : le_meilleur). Ensuite, les différents filtres appliqués par l'utilisateur (comme la sélection de la classe DPE ou le type de logement) déclenchent la mise à jour des graphiques et des cartes en temps réel via des fonctions comme renderPlot() et renderLeaflet(). Une fonctionnalité importante est la possibilité de télécharger les graphiques et les données. Cette option est activée via downloadHandler(), permettant aux utilisateurs de sauvegarder leurs analyses sous forme de fichiers PNG ou CSV.

Schéma de l'architecture de l'application



Documentation fonctionnelle :

Cette partie sert à l'utilisateur de l'application. Elle lui permettra de connaitre les fonctionnalités que celle-ci permet de visualiser.

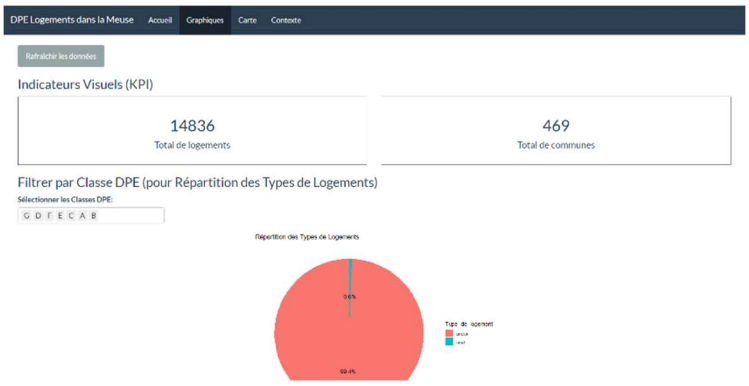


Dès que le code est lancé, la page demandera une authentification. L'identifiant est : asardell , et le mdp est : le_meilleur .

Une fois sur la page d'accueil, celui-ci nous plonge dans le département de la Meuse. On peut y retrouver leur célèbre étendard, leur fleuve, ou encore leur situation géographique. En haut à droite, une fonctionnalité permet d'activer le mode nuit ou non, afin d'assombrir l'écran si besoin. Si l'utilisateur souhaite avoir plus de renseignement sur le département 55, il peut utiliser le lien cliquable qui l'emmènera sur Wikipédia à la page en question.

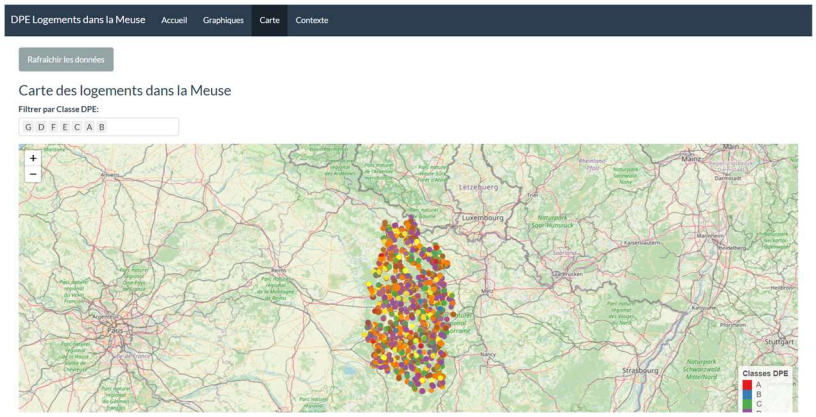
On peut également retrouver 3 onglets sur le haut de la page (comme ci-dessus). On y retrouve l'onglet :

- ***Graphique*** : Permettant d'afficher des visualisations et analyses concernant les logements de la Meuse.
- ***Carte*** : Permettant de visualiser de façon interactive et dynamique les différents logements selon leur position géographique.
- ***Contexte*** : Permettant de retrouver l'entièreté des données, accessible au téléchargement.



Dans l'onglet « Graphique », on peut retrouver deux KPIs, montrant le nombre total de logements ainsi que de communes. Plusieurs graphiques se retrouvent sur cette même page, à savoir un graphique de la répartition des DPE, le TOP 3 des communes avec le plus de logements, suivi de la distribution des classes DPE par type de logements ainsi qu'un nuage de points avec régression linéaire (avec choix des X et Y).

On retrouve deux fonctions majeures sur cette page, à savoir le rafraîchissement de données et le téléchargement en PNG. Celles-ci permettent de suivre en temps réel les dernières données concernant la Meuse, ainsi que le téléchargement direct pour des analyses personnelles. Les graphiques sont également liés à des filtres dynamiques pour des analyses plus ciblées.



Concernant l'onglet « Carte », la carte interactive permet d'observer la disparité des logements selon les classes DPE dans le territoire de la Meuse. Les données sont également rafraîchissables car directement issues de l'API et filtrées selon les notes. Pour finir, l'onglet « Contexte » présente un tableau de l'entièreté des données. On peut également filtrer de plusieurs manières comme ci-après. On peut également sélectionner n'importe quelles données afin de les télécharger au format csv.