

DOCUMENTATION PROJET DASHBOARD

By Théo Capelle & Christophe Tiet

SOMMAIRE

I – Présentation du projet

II – Choix des technologies

III – Conception et architecture

1 – Architecture générale

2 – Architecture du back

3 – Architecture du front

4 – Architecture du docker

IV – UX et UI

1 – UI

2 - UX

V – Démarrer notre projet

I – Présentation du projet

Le but du projet est de développer une application Web de dashboard ressemblant à Netvibes.

II – Choix des technologies

La décision concernant la stack technologique MEAN à utiliser fut plutôt rapide. Theo connaissait Angular et Christophe avait déjà travaillé sur Nodejs. Nous nous sommes donc réparti le travail avec l'un sur le front et l'autre sur le back. Concernant la base de données, nous avons fait le choix d'utiliser MongoDB pour toucher un peu au NoSQL.

III – Conception et Architecture.

1 - Architecture Générale

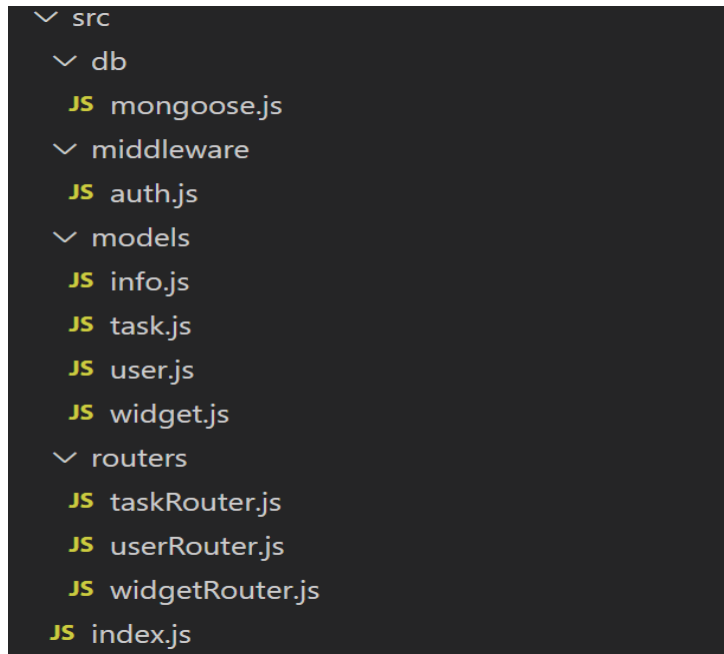
Back

- Authentification JWT
- Routes pour gérer la persistance des utilisateurs, des widgets et des taches de la to-do list.
- Base de données MongoDB

Front

- Gridster pour pouvoir drag'n'drop les widgets
- Appel vers le back et API tierce
- Implémentation de l'UI et de l'UX

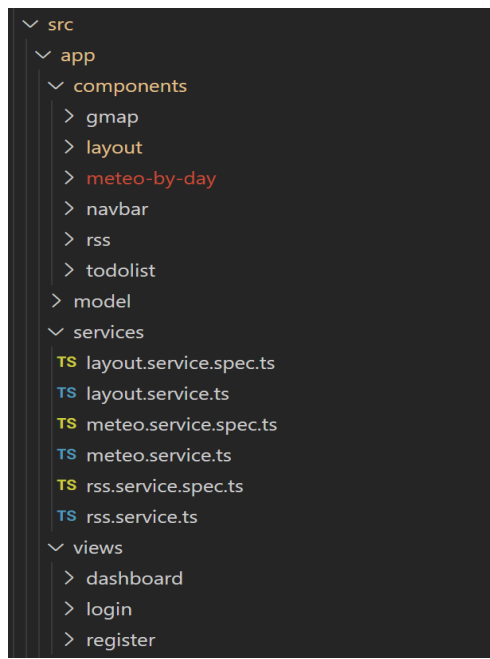
2 - Architecture du back



Le back se décompose en 4 dossiers :

- db contient les fichiers permettant de se connecter à la BDD via le module mongoose
- middleware contient le middleware d'authentification JWT
- models contient les schémas utilisés.
- routers contient tous nos routes CRUD pour le front avec le module Express

3 – Architecture du front



Le front se compose en plusieurs parties :

- La partie components, avec tous les widgets ainsi que leur données
- La partie views, qui représente donc les différentes pages du site (login, register, dashboard)
- La partie service, permettant de gérer toute les traitements de données du front

4 – Architecture du docker

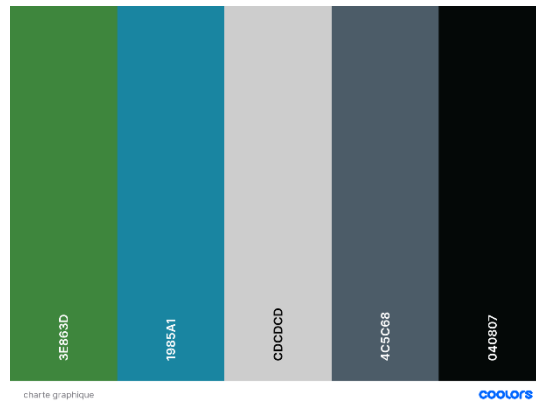
Le docker-compose exécute trois containers :

- Un container pour le serveur angular exposé sur le port 4200
- Un container pour le server nodeJS exposé sur le port 8080
- Un container pour MongoDB exposé sur le port 27017

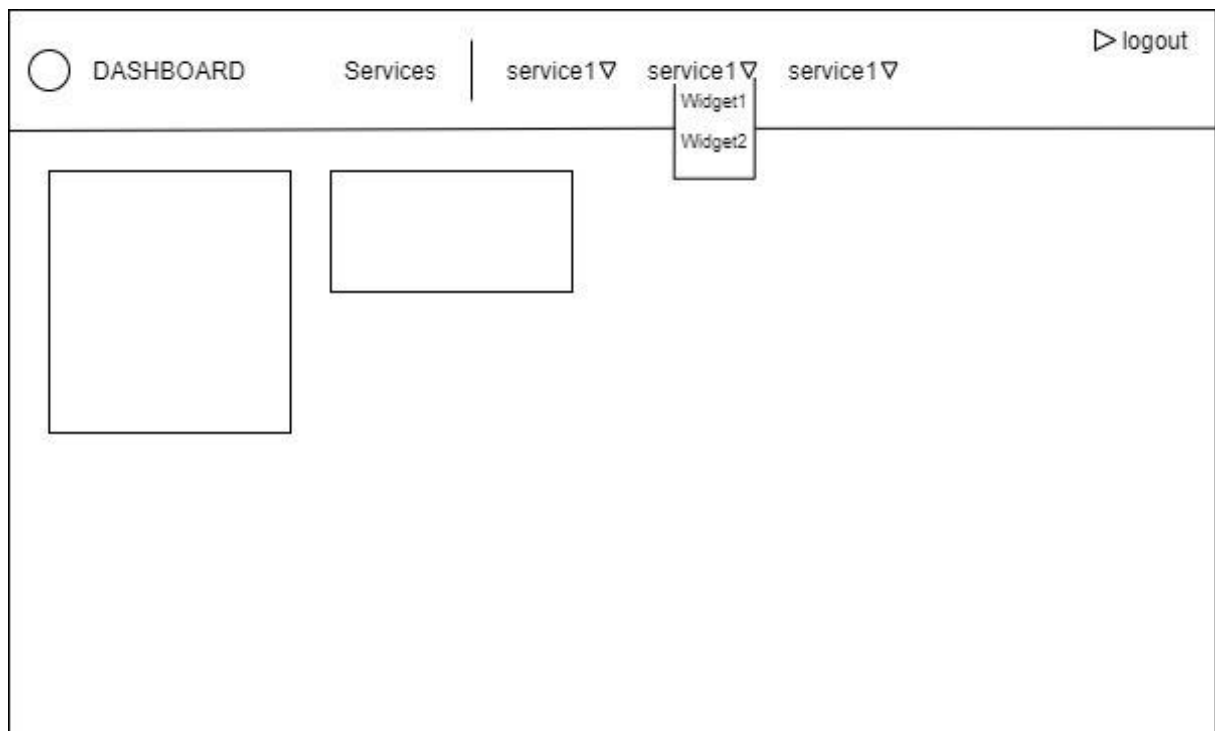
IV – UX et UI

1 – UI

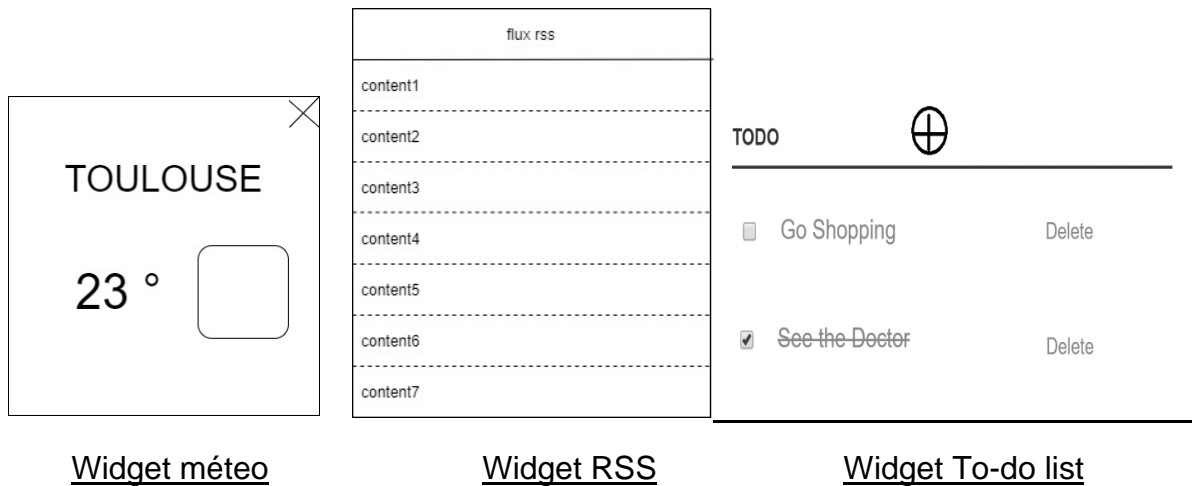
Charte graphique



Mock-ups



Site principal



2 – UX

Afin d'avoir une expérience utilisateur la plus agréable possible, nous voulions que l'utilisateur puisse tout faire de la même page : s'abonner aux services et display ses widgets. Nous avons donc épuré le design au maximum pour que seules les infos essentielles soit visibles : less is more.

Toute la gestion des services et l'ajout de widget se fait depuis la nav bar. L'édition des widgets se fait directement sur le widget.

V – Démarrer notre projet

Lancer docker

A l'aide du terminal, une fois dans le dossier DEV_dashboard_2019, taper la commande : docker-compose up –build

Ouvrir le browser vers l'adresse : <http://localhost:4200>