Documentation technique

DASHBOARD

Sommaire

Application Mobile :

1. Dashboard.js.................................................4
2. Login.js..........................................................5
3. Register.js.....................................................6
4. SettingsWidget.js..........................................7
5. App.js............................................................8

Dashboard

Dashboard est un site web permettant à l’utilisateur d’avoir accès à un tableau de bord personnalisable en fonction des services qu’il utilise.

Dashboard utilise l’api material-ui dans la plupart des fichiers. L’api sera utilisée dans toutes la partie affichage du site.

1. Dashboard.js :

Le fichier dashboard.js contient en lui-même la page principale du site internet.

class Dashboard extends React.Component; Cette classe affiche la page principale du site.

Une balise <AppBar/> affiche la base de l’interface de la page.

La class contient aussi une balise <Drawer/> qui contient elle-même les balises :

<WidgetWeather/>

<WidgetGmail/>

<WidgetGithub/>

<WidgetCoinranking/>

<CovidSettings/>

Chacune de ces balises permettent de configurer un widget.

Il faudra donc rajouter à la suite de ces balises une nouvelle configurations à chaque nouveaux widgets.

1. Login.js :

Le fichier login.js permet d’afficher la page login.js.

export default function SignIn();

Contient les 4 balises <TextField/> pour que l’utilisateur puisse se connecter au site.

Les 2 balises <MyButton/> permettent de se connecter au site ou d’aller à la page regiser.

1. Register.js :

Le fichier contient la page register.

export default function SignUp();

Les 4 balises <TextField/> permettent à l’utilisateur de se connecter.

Les balises <MyButton/> permettent de s’enregistrer, se connecter au site ou d’aller à la page login.

1. SettingsWidget :

Le dossier SettingsWidget contient tous les widgets du site.

Par exemple la fonction export default function WidgetWeather(props); permet à l’utilisateur de configurer un widget se basant sur le service météo proposé.

La fonction contient des balises <TextField/> <Checkbox/> afin que l’utilisateur puisse configurer son widget.

Ainsi pour configurer un nouveau widget tiré d’un nouveau service, il faut rajouter un fichier du nom du service. Ce fichier devra contenir des balises de types <TextField/> <Checkbox/> correspondant aux différentes possibilités tiré du service.

1. App.js :

La page contient le router qui permet d’aller de page en page.

export default function App();

La fonction contient le router et permet d’aller de la page login à register ou dashboard.

La fonction permet aussi d’utiliser l’history pour revenir en arrière ou en avant.