

# APPLICATION WEB DE LOCATION IMMOBILIÈRE AVEC REACT

# KASA

Projet 7 Développeur Web OpenClassRooms  
Magali BERNARD 22-11-2022

# PRESENTATION DU PROJET

## MAQUETTE FIGMA DES 4 PAGES DU SITE:

- Respect de la disposition des éléments en version desktop et mobile
- Respect des dimensions et des couleurs en version desktop et mobile
- Utilisation du logo et des images

=> MOBILE FIRST

=> Dimensions des écrans de référence d'après largeur des footer figma:

- 375px mobile
- 1440px desktop
- point de rupture 767px

## OUTILS ENVIRONNEMENT DE TRAVAIL:

- REACT: composants et React Router
- Sass pour le style
- Create React App =>
  - squelette d'application React
  - embarque certains outils préconfigurés (Webpack, Babel, ESLint),
- appli hébergée en local sur le port 3000 : <http://localhost:3000/>
- gestionnaire de paquets yarn (yarn start pour démarrer la compilation)

# ARCHITECTURE APPLICATION

## PUBLIC

Index.html dont le body contien la div "root"

## SRC>ASSETS

images et logo

## SRC>COMPONENTS

composants React, réutilisables dans plusieurs pages

## SRC>PAGES

composants "page"

## SRC>STYLES

fichiers sass liés aux composants en jsx

## SRC>APP.JSX : ROUTAGE

## SRC>INDEX.JSX : LIE L'APPLICATION A INDEX.HTML

## SRC>DATA.JSON : DONNEES FIGURANT LE BACK

# PRESENTATION DES PAGES

## ACCEUIL

### COMPOSANTS:

- page:Acceuil.jsx=>aceuill.scss
  - Masque.jsx
- 

- Card.jsx=>card.scss

### LIENS:

- chaque card vers page logement

## LOGEMENT

### COMPOSANTS:

- page:Logement.jsx=>logement.scss
  - Masque.jsx
- 

- Gallery.jsx=>gallerie.scss
  - Tags.jsx=>tags.scss
  - RateAndProprio.jsx=>scss
  - divMasqueble.jsx=>divDeroulante.scss
- 

## A PROPOS

### COMPOSANTS:

- page:Apropos.jsx=>apropos.scss
  - LMasque.jsx
- 

- divMasqueble.jsx=>divDeroulante.scss
- 

## ERREUR

### COMPOSANTS:

- page:Erraur.jsx=>erreur.scss
  - Masque.jsx
- 

### LIENS:

- lien supplémentaire vers accueil

# REACT ROUTER

NAVLINK permet un routage plus léger, moins gourmand en ressource que les liens html `<a>` car ne se recharge que ce qui change dans la pge au click sur un lien.

Le composant Masque.jsx comprend le logo, le menu de navigation et le footer. Il est rendu en route "parente" des autres routes vers les pages qui le nécessitent grâce à Outlet importé de react-router-dom:

```
import './styles/footer.scss';
import { Outlet } from "react-router-dom";
import React from "react";
import Nav from "../components/Nav";
import Footer from "../components/Footer";
import LOGO from "../assets/LOGO.png";
export default class Masque extends React.Component {
  render() {
    return (
      <div className="masque">
        <div className="ligneDuHaut">
          <img src={LOGO} alt="Logo Kasa" className="ligneDuHaut_logo" />
          <Nav />
        </div>
        <Outlet />
        <Footer />
      </div>
    );
  }
}
```

```
import Accueil from "../pages/Accueil";
import Apropos from "../pages/Apropos";
import Logement from "../pages/Logement";
import Erreur from "../pages/Erreur";
import Masque from "../pages/Masque";

import { BrowserRouter, Routes, Route } from "react-router-dom";

import React from "react";

const App = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Masque />} />
        <Route path="" element={<Accueil />} />
        <Route path="Apropos" element={<Apropos />} />
        <Route path="Logement/:id" element={<Logement />} />
        <Route path="*" element={<Erreur />} />
      </Routes>
    </BrowserRouter>
  );
};

export default App;
```



# PAGE ACCEUIL

```
class Accueil extends React.Component {
  render() {
    return (
      <div className="accueil">
        <div className="cote">
          <img src={COTE} alt="cote" className="cote_image" />
          <p className="cote_texte"> Chez vous, partout et ailleurs</p>
        </div>

        <div className="espaceCartes">
          /* creation des cards à partir du json */
          {data.map((eachcard) => (
            <Card
              idLogement={eachcard.id}
              titreLogement={eachcard.title}
              coverLogement={eachcard.cover}
            />
          ))}
        </div>
      </div>
    );
  }
}
```

Responsive:

- mobile first, rupture à 786px
- changement de police du texte dans la bannière
- changement de mode flex dans la disposition des cartes

Lien vers la page logement correspondante au logement cliquer:

- l'id du logement est passée en props de accueil.jsx vers card.jsx
- au click sur le Link "carteLogement" l'appli redirige vers l'url: /Logement/\${props.idLogement}
- le BrowserRouter dans App.jsx reconnaît le path: "/Logement/:id" et rend le composant Logement.jsx

```
import React from "react";
import { Link } from "react-router-dom";
function Card(props) {
  return (
    <div>
      <Link className="carteLogement" to={` /Logement/${props.idLogement}`}>
        <p className="carteLogement_titre">{props.titreLogement}</p>
        <img
          src={props.coverLogement}
          alt="cover"
          className="carteLogement_image"
        />
      </Link>
    </div>
  );
}
export default Card;
```



# PAGE LOGEMENT: DONNEES

```
import { useParams } from "react-router-dom";
import Erreur from "../Erreur";
export default function Logement() {
  const idLogement = useParams().id;
  let ceLogement = data.find((card) => card.id === idLogement);
  if (!ceLogement) {
    return <Erreur />;
  }
  const listpEquip = ceLogement.equipments.map((equip) => <p>{equip}</p>);
  return (
    <div className="logement">
      <Gallery images={ceLogement.pictures} altImages={ceLogement.title} />
      <div className="TitresTagsRateProprio">
        <div className="TitresTagsRateProprio_TitresTags">
          <div className="titres">
            <p className="titres_grosTitre">{ceLogement.title}</p>
            <p className="titres_petitTitre">{ceLogement.location}</p>
          </div>
          <Tags tags={ceLogement.tags} />
        </div>
        <RateAndProprio etoilles={ceLogement.rating} hote={ceLogement.host} />
      </div>
      <div className="DeroulanteContainer">
        <DivMasquable
          title="Description"
          render={() => <p className="pLogement">{ceLogement.description}</p>}
        />
        <DivMasquable
          title="Equipement"
          render={() => <div className="pLogement">{listpEquip}</div>}
        />
      </div>
    </div>  </div>  );}
```

- l'id du logement est passée en Paramètre dans l'url de la page
- grâce à l'import de useParams on récupère l'id.
- on trouve le logement tel que l'id de l'url soit le même que l'id dans le fichier json de données "data.json".
- s'il existe ce logement est stocké dans une constante "ceLogement" pour pouvoir accéder à ses différentes propriétés.
- on appelle les composants de la page

NB: appel du composant Gallery avec en props la liste d'images et le titre du logement .

# PAGE LOGEMENT: GALERIE D'IMAGES

```
import React from "react";
import { useState } from "react";
import "../styles/gallerie.scss";
function Gallery(props) {
  const [currentIndex, setCurrentIndex] = useState(0);

  const tabImages = props.images.map((eachimg, keyImg) => (
    <img
      src={eachimg}
      alt={props.altImages}
      key={keyImg}
      className="gallery_image"
    />
  ));
  return (
    <div className="gallery">
      {tabImages.length > 1 ? (
        <div className="btnContainer">
          <div
            className="btnContainer_btnPrev"
            onClick={() => {
              if (currentIndex === 0) {
                return setCurrentIndex(tabImages.length - 1);
              }
              return setCurrentIndex(currentIndex - 1);
            }}
          >
            <i class="fa-solid fa-chevron-left"></i>
          </div>
```

```
          <div className="btnContainer_btnNext"
            onClick={() => {
              if (currentIndex === tabImages.length - 1) {
                return setCurrentIndex(0);
              }
              return setCurrentIndex(currentIndex + 1);
            }}
          >
            <i class="fa-solid fa-chevron-right"></i>
          </div>
        </div>
      ) : (
        <div className="btnContainer"></div>
      )}
      <img
        src={props.images[currentIndex]}
        alt="cover"
        className="gallery_image"
      />
      {tabImages.length > 1 ? (
        <div className="affichIndex">
          {currentIndex + 1}/{tabImages.length}
        </div>
      ) : (
        <div className="affichIndex"> </div>
      )}
    </div>
  );}export default Gallery;
```

- on déclare une constante currentIndex d'état initial = 0, assortie d'un setCurrentIndex qui permettra de changer d'état grâce à l'import de useState
- on crée un tableau de balises img, une par image de la liste props.images, grâce à la méthode de tableau .map, avec chacune une clé propre.
- on affiche la balise image du tableau d'image dont l'index = currentIndex.
- s'il y a plus d'une image, on affiche la div de numéro d'index et les chevrons Prev et Next
- au click sur l'un d'eux, on change currentIndex en conséquence grâce à setCurrentIndex. sans oublier de gérer le passage de la dernière image à la première et inversement.



# PAGE LOGEMENT: COLLAPSE

```
export default function Logement() {
  ...
  const listpEquip = ceLogement.equipments.map((equip) => <p>{equip}</p>);
  return (
    <div className="logement">
      ...
      <div className="DeroulanteContainer">
        <DivMasquable
          title="Description"
          render={() => <p className="pLogement">{ceLogement.description}</p>}
        />

        <DivMasquable
          title="Equipement"
          render={() => <div className="pLogement">{listpEquip}</div>}
        />
      </div>
    </div>
  );
}
```

- on déclare une constante showDivMasquable d'état initial = false en utilisant UseState.
- au click sur le bandeau, on change l'état de showDivMasquable .
- si l'état est false, on affiche le chevron down dans le bandeau et rien d'autre
- si l'état est true, on affiche la div masquable contenant le rendu programmé dans la props render lors de l'appel du composant DivMasquable.

- on appelle le composant divMasquable en lui passant les props title et render

NB: render est une "render props" i.e une propriété "fonction" au lieu d'une propriété "élément", ce qui permet de modifier le rendu du composant à chaque appel.

```
import React, { useState } from "react";
function DivMasquable(props) {
  const [showDivMasquable, setShowDivMasquable] = useState(false);
  return (
    <div className="divDeroulante">
      <div className="divDeroulante_bandeau">
        <p>{props.title}</p>
        <div
          className="divDeroulante_bandeau_down"
          onClick={() =>
            setShowDivMasquable((showDivMasquable) => !showDivMasquable)
          } >
          {showDivMasquable ? (
            <i class="fa-solid fa-chevron-up"></i>
          ) : (
            <i class="fa-solid fa-chevron-down"></i>
          )}
        </div>
      </div>
      {showDivMasquable ? (
        <div className="divDeroulante_masquable">{props.render()}</div>
      ) : null}
    </div> );}
export default DivMasquable;
```

# PAGE LOGEMENT: RATEANDPROPRIO

```
function RateAndProprio(props) {
  console.log(props.etoiles);
  console.log(props.hote);
  const wordsHote = props.hote.name.split(" ");
  const firstWord = wordsHote[0];
  const lastWord = wordsHote[1];
  const etoilesGrises = 5 - props.etoiles;
  const etoiles = [];
  for (let i = 0; i < props.etoiles; i++) {
    etoiles.push(
      <div className="etoileOrange">
        <i className="etoileOrange fa-solid fa-star"></i>
      </div>
    );
  }
  for (let i = 0; i < etoilesGrises; i++) {
    etoiles.push(
      <div className="etoileGrise">
        <i className="etoileGrise fa-solid fa-star"></i>
      </div>
    );
  }
  return (
    <div className="TitresTagsRateProprio_rateAndProprio">
      <div className="TitresTagsRateProprio_rateAndProprio_rate">{etoiles}</div>
      <div className="TitresTagsRateProprio_rateAndProprio_hote">
        <div className="TitresTagsRateProprio_rateAndProprio_hote_name">
          <p>{firstWord}</p>
          <p>{lastWord}</p>
        </div>
        <div className="TitresTagsRateProprio_rateAndProprio_hote_imgHote">
          <img src={props.hote.picture} alt={props.hote.name} />
        </div>
      </div>
    </div>
  );
}
export default RateAndProprio;
```

- Props: <RateAndProprio etoiles={ceLogement.rating} hote={ceLogement.host} />
- on utilise la méthode .split des tableaux de chaîne pour scinder la chaîne hote.name en prénom et nom.
- on déclare un tableau etoiles qu'on remplit d'étoiles oranges jusqu'au chiffre entre 0 et 5 passé en props.etoiles (-1 car on commence à 0)
- on ajoute le nombre d'étoiles grises = (5-nombre d'étoiles oranges)
- on retourne pour affichage:
  - les 5 étoiles personnalisées
  - les nom et prénom dans 2 <p> séparés pour la mise en page
  - la photo du propriétaire passée en props.hote.picture

# PAGE A PROPOS

On utilise le même composant DivMasquable que pour les collapses de la page logement.

# PAGE D'ERREUR

- path="\*" : toute url différente de celles déclarées dans le router.
- ajout d'un NavLink supplémentaire pour retourner vers l'accueil

**MERCI POUR VOTRE ATTENTION !**