

Soutenance KASA

**Créez une application web de location immobilière
avec React**

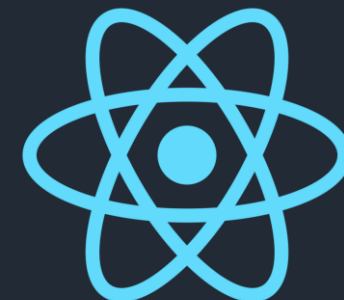


Contexte

- Kasa vous recrute en tant que développeur front-end en freelance pour développer sa nouvelle plateforme web.
- Le site de Kasa a été codé il y a maintenant plus de 10 ans en ASP.NET avec un code legacy important.
- La CTO a donc lancé une refonte totale pour passer à une stack complète en JavaScript avec NodeJS côté back-end, et React côté front-end, ce qui va être notre rôle ici.

Étape 1 : Initialisez le projet

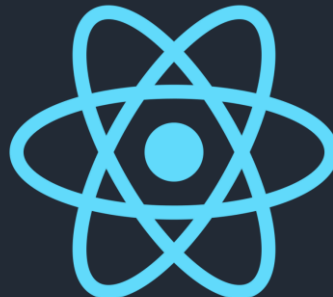
- **Création du projet avec create React App**
- **Lien du repo GitHub: <https://github.com/lanPelina/kasa>**
- **Ajout du fichier Json au projet**



Étape 2 : Ajoutez React Router

- Ajout de React Router et de ses dépendances

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



Étape 3 : Créez le routeur et les différentes pages de l'application

1. Routage de l'application fonctionnel

2. Composant à part pour le routeur

```
export default function Router() {  
  return (  
    <Routes>  
      <Route path="/" element={<Home />} />  
      <Route path="/apropos" element={<About />} />  
      <Route path="/logements/:id" element={<Housing />} />  
      <Route path="*" element={<Error />} />  
    </Routes>  
  )  
}
```



Étape 4 : Créez la page d'accueil

Composant Banner

```
export default function Banner({ children }) {  
  return (<div className="banner about-banner">{children}</div>)  
}
```

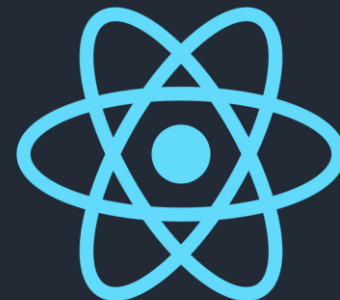
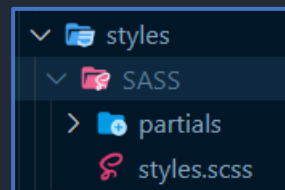
```
<Banner>  
  <div className="banner-container">  
    <img className='banner__img' src={Img} alt="paysage" />  
    <p className="banner__tagline" >Chez vous, partout et ailleurs</p>  
  </div>  
</Banner>
```

```
<Banner>  
  <img className='banner__img about-img' src={Img} alt="paysage" />  
</Banner>
```

Composant Card

```
export default function Card({ logement }) {  
  
  return (  
    <Link to={`/logements/${logement.id}`} className="card">  
      <img src={logement.cover} alt={logement.cover} />  
      <h3>{logement.title}</h3>  
    </Link>  
  );  
}
```

Utilisation de Sass pour le style



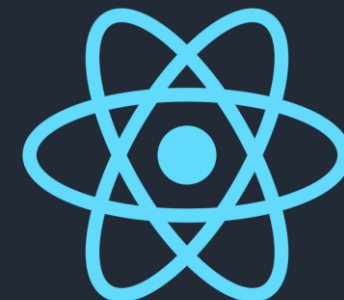
Étape 5 : Créez la page À propos

Composant Accordion

```
export default function Accordion({ title, description }) {  
  
  const [active, setActive] = useState(false);  
  
  const toggle = () => {setActive(!active);};  
  
  const content =  
    Array.isArray(description) === true ? (  
      <ul>  
        {description.map((item, index) => (  
          <li key={index}>{item}</li>  
        ))}  
      </ul>  
    ) : (  
      description  
    );  
  
  return (  
    <div className={`accordion ${active ? "fall" : ""}`}>  
      <div className="accordion__title" key={Accordion.id} onClick={toggle}>  
        {title}<img className={`accordion__icon ${active ? "rotate" : ""}`} src={AccordionLogo} alt="arrow" />  
      </div>  
      <div className={`accordion__content ${active ? "accordion-show" : ""}`}>{content}</div>  
    </div>  
  );  
}
```

Page A Propos

```
export default function Apropos() {  
  return(  
    <div>  
      <div>  
        <Banner>  
          <img className='banner__img about-img' src={Img} alt="paysage" />  
        </Banner>  
      </div>  
      <div className="accordions">  
        {AccordionData.map((Accordion) => (  
          <Accordions  
            key={Accordion.id}  
            title={Accordion.title}  
            description={Accordion.description}  
          />  
        ))}  
      </div>  
    </div>  
  );  
}
```



Étape 6 : Créez la page d'erreur

```
<Route path="*" element={<Error />} />
```

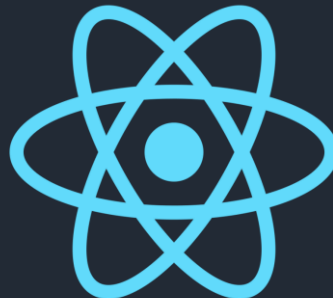
```
export default function Error() {  
  return (  
    <div className="error-page">  
      <div className="error-logo">  
        <img src={ErrorLogo} alt="error"/>  
      </div>  
      <div className="error-and-link">  
        <p className="error-msg">Oops! La page que vous demandez n'existe pas.</p>  
        <Link className="home-link" to="/">Retourner sur la page d'accueil</Link>  
      </div>  
    </div>  
  )  
}
```



Étape 7 : Créez le carrousel de la page de logement

```
export default function Carrousel({pictures}) {  
  
  const [currentPicture, setCurrentPicture] = useState(0);  
  
  const getClassName = (i) => {  
    if (i === currentPicture) return "show";  
    return "";  
  };  
  
  const nextSlide = () => {  
    setCurrentPicture((currentPicture + 1) % pictures.length);  
  };  
  
  const previousSlide = () => {  
    const newCurrentPicture = currentPicture - 1;  
    if (newCurrentPicture < 0) {  
      setCurrentPicture(pictures.length - 1)  
      return  
    };  
    You, hier • housing+responsive(manque underline liens / ...  
    setCurrentPicture(currentPicture - 1);  
  }  
}
```

```
return(  
  <div className="apartment_img">  
    {pictures.length > 1 && (  
      <div className="arrows">  
        <img className="show arrow left" src={left} onClick={previousSlide} />  
        <img className="show arrow right" src={right} onClick={nextSlide} />  
      </div>)}  
    {pictures.map((picture, i) => (  
      <img key={picture} src={picture} className={getClassName(i)}></img>  
    ))}  
  </div>  
)  
}
```

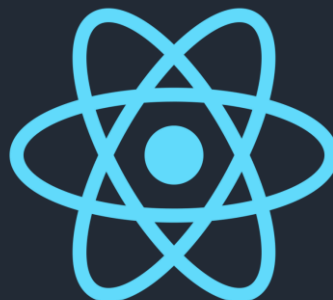


Étape 8 : Finalisez la page de logement

```
export default function Details({logement}) {  
  
  return (  
    <div className="apartment__infos">  
      <div className="apartment__description">  
        <h1>{logement.title}</h1>  
        <p>{logement.location}</p>  
        <div className="apartment__tags">  
          {logement.tags.map(tag => <div className="apartment__tag">{tag}</div>)}  
        </div>  
      </div>  
      <div className="apartment__user">  
        <div className="text-and-pics">  
          <div className="text">  
            <p>{logement.host.name}</p>  
            <p>{logement.host.surname}</p>  
          </div>  
          <div className="pics">  
            <img alt="img" src={logement.host.picture} />  
          </div>  
        </div>  
        <div className="rating">  
          { Array(Number(logement.rating)).fill(<div className="star"><img alt="img" src={full}/></div>) }  
          { Array(5- Number(logement.rating)).fill(<div className="star"><img alt="img" src={empty}/></div>) }  
        </div>  
      </div>  
    </div>  
  )  
}
```

```
export default function Collapse({ title, description }) {  
  
  const [active, setActive] = useState(false);  
  
  const toggle = () => {setActive(!active);};  
  
  const content =  
    Array.isArray(description) === true ? (  
      <ul>  
        {description.map((item, index) => (  
          <li key={index}>{item}</li>  
        ))}  
      </ul>  
    ) : (  
      description  
    );  
  
  return (  
    <div className={`housing-accordion ${active ? "housing-fall" : ""}`}>  
      <div className="housing-accordion__title" onClick={toggle}>  
        {title}<img className={`accordion__icon ${active ? "rotate" : ""}`} src={AccordionLogo} alt="arrow" />  
      </div>  
      <div className={`housing-accordion__content ${active ? "show" : ""}`}>  
        {content}  
      </div>  
    </div>  
  )  
}
```

You, il y a 30 secondes • Uncommitted changes



Étape 9 : Gérez l'erreur sur l'id d'une propriété

```
return logements.find((logement) => logement.id === id);  
}, [id]);  
  
// useEffect se lance uniquement lorsque la valeur de sa dépendance change. Ici, sa dépendance est logement.  
useEffect(() => {  
  if (!logement) {  
    return navigate("/404");  
  }  
})
```

