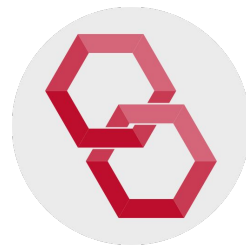


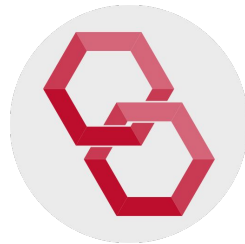
**Formation
React.js**

Sommaire

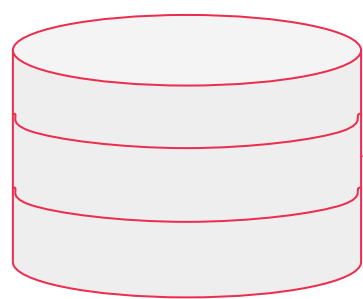
1. Le contexte
2. React : c'est quoi ?
3. Un peu de pratique !



Le contexte



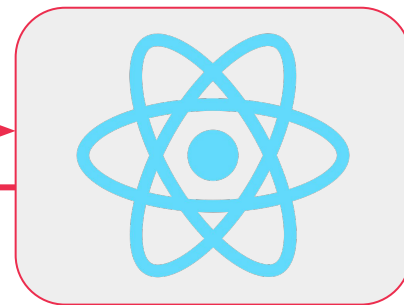
Le contexte



Database



Backend
(API)



Frontend

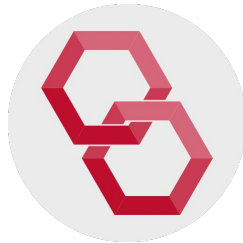


FRONT-END



BACK-END

React : c'est quoi ?



React : c'est quoi ?

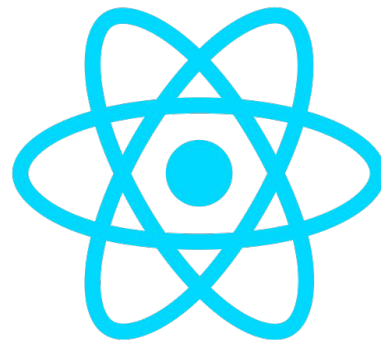
- Un framework web de front-end
- Il en existe d'autres : Angular, Vue...

Pourquoi faire ?

- Créer des applications web “single-page”, rapidité, UX
- Performances : système de *components*, DOM virtuel

Pourquoi React ?

- Le JSX, une syntaxe très pratique qui lie HTML, CSS et JS
- Très à la mode, grande communauté, beaucoup de modules



React : c'est quoi ?

```
import React from 'react'; 8.6K (gzipped: 3.4K)

const MyComponent = ({ message }) => {
  return (
    <div style={{ backgroundColor: '#ab0028' }}>
      <button onClick={() => alert(message)}>Click me</button>
    </div>
  );
}

const OtherComponent = () => {
  return (
    <MyComponent message='Hello World!' />
  );
}
```

Déclaration de “Components”

React : c'est quoi ?

```
const MessageList = (props) => {
  const [messages, setMessages] = useState([]);
  const [users, setUsers] = useState([]);

  useEffect(
    () => {
      getMessagesList(props.board).then(data => setMessages(data));
      getUserList().then(data => setUsers(data));
    },
    [props.board]
  );

  useEffect(
    () => {
      const timer = setTimeout(() => {
        getMessagesList(props.board).then(data => setMessages(data));
      }, 1000);
      return () => clearTimeout(timer);
    },
    [messages, props.board]
  );

  return (
    <div>
      {messages.map(message => (
        <MessageDisplay message={message} user={users[message.userId-1]}/>
      ))}
    </div>
  );
};
```

React : c'est quoi ?

```
const MessageList = (props) => {  
  const [messages, setMessages] = useState([]);  
  const [users, setUsers] = useState([]);  
  
  return (  
    <div>  
      {messages.map(message => (  
        <MessageDisplay message={message} user={users[message.userId-1]}/>  
      ))}  
    </div>  
  );  
};
```

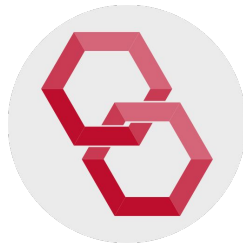
Définition de variables dans le Component avec un "State"

React : c'est quoi ?

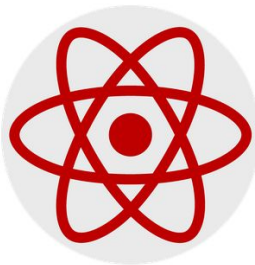
```
useEffect(  
  () => {  
    getMessagesList(props.board).then(data => setMessages(data));  
    getUserList().then(data => setUsers(data));  
  },  
  [props.board]  
);  
  
useEffect(  
  () => {  
    const timer = setTimeout(() => {  
      getMessagesList(props.board).then(data => setMessages(data));  
    }, 1000);  
    return () => clearTimeout(timer);  
  },  
  [messages, props.board]  
);
```

useEffect : appeler une fonction à chaque fois qu'une variable change

Un peu de pratique !



Résultat



Username

Password

Login

Register

Spam

Send

jean-mich@09:35:41
tranquiiiille

safarte@09:35:16
ça va ?

safarte@22:06:19
Hello



Installation

- Git
- NodeJS
- Yarn

Installation

```
git clone https://gitlab.viarezo.fr/LinkCS/react-tutorial-front.git  
git@github.com:Safarte/formation-react-front.git
```

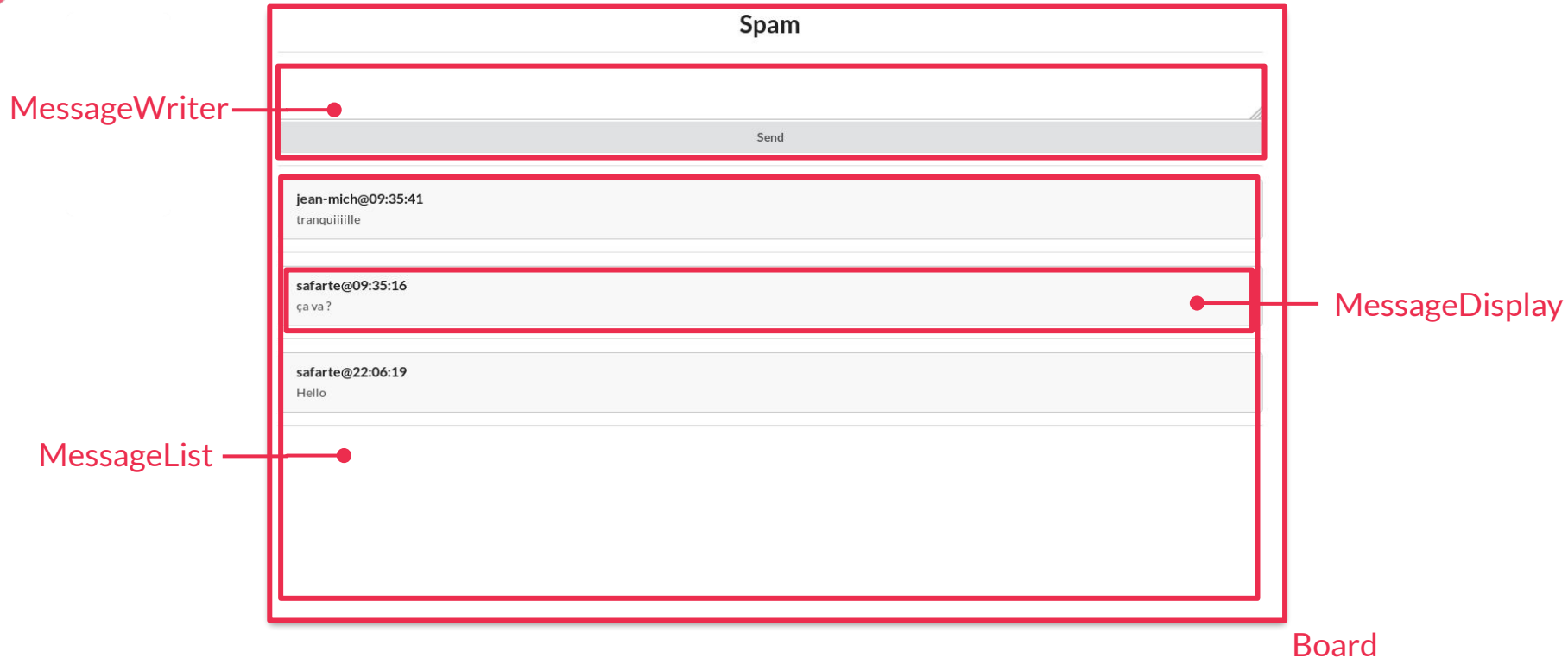
```
cd react-tutorial-front
```

```
yarn install
```

Pour lancer votre Front :

```
yarn start
```

Structure d'un "Board"



Envoyer une requête HTTP

```
const API_URL = 'http://138.195.142.10';

const postMessage = async (message, board) => {
  const request_options = {
    url : `${API_URL}/messages`,
    method : 'POST',
    data: {
      userId: 1,
      content: message,
      boardId: parseInt(board)
    }
  };
  return axios.request(request_options).then(resp => resp.status);
};
```

Documentation de l'API dans le README.md !

Pense-bête

```
const myFunction = (arg) => {  
  console.log(arg);  
}
```

Définition de la fonction *myFunction*.

`axios.request(req_options);` Fait une requête HTTP avec les paramètres *req_options*, renvoie une **Promise**.

`prom.then(returnValue => f(returnValue));` Applique la fonction *f* à la valeur renvoyée par *prom*, qui est une **Promise**.

Component :

```
const myComponent = () => {  
  return (  
    <p>Hello World!</p>  
  );  
}
```

Définition d'un Component appelé *myComponent*.

`[var, setVar] = useState(default);` Définit un “State” *var* modifiable avec *setVar(value)*.

```
useEffect(() => {  
  console.log(var);  
},  
[var]  
);
```

Exécute la fonction en premier argument à chaque fois que *var* est modifiée.

Promise

```
const getMessagesList = async (board) => {  
  const request_options = {  
    method: 'GET',  
    url: `${API_URL}/messages`,  
    params: {  
      board: board  
    }  
  };  
  return axios.request(request_options).then(resp => resp.data);  
};  
  
getMessagesList(props.board).then(data => setMessages(data));
```