

## Construisez le composant de Tweet



Vous devez utiliser git pour faire des snapshots de votre travail. Chaque état du composant Tweet doit être capturé avec git.

### Plan de construction :

Nous allons construire un tweet statique simple dans cette section, en commençant par le composant de niveau supérieur, Tweet.

Créez un nouveau projet avec Create React App en exécutant cette commande :

```
$ create-react-app static-tweet && cd static-tweet
```

Le projet nouvellement généré est livré avec un fichier index.html dans le répertoire public. Ajoutez Font Awesome en ajoutant cette ligne à l'intérieur de la balise <head> (mettez tout sur une seule ligne) :

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
```

Nous avons également un fichier index.css généré dans le répertoire src. Ouvrez-le et remplacez son contenu par ceci :

```
.tweet {  
  border: 1px solid #ccc;  
  width: 564px;  
  min-height: 68px;  
  padding: 10px;  
  display: flex;  
  font-family: "Helvetica", arial, sans-serif;  
  font-size: 14px;  
  line-height: 18px;  
}
```

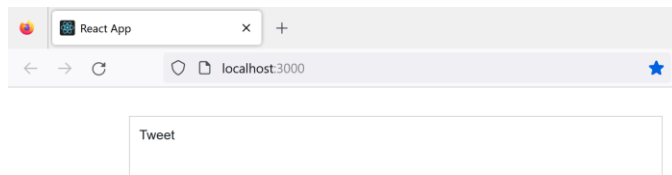
Créez le composant Tweet et montez le dans le fichier index.js en lieu et place du composant habituel App.

```

import React from 'react';
import './index.css';
function Tweet() {
  return (
    <div className="tweet">
      Tweet
    </div>
  );
}

```

Nous obtenons



Créez le composant Avatar

```

function Avatar() {
  return (
    
  );
}

```

Ensuite, nous devons modifier le composant de Tweet pour afficher Avatar :

```

function Tweet() {
  return (
    <div className="tweet">
      <Avatar/>
      Tweet
    </div>
  );
}

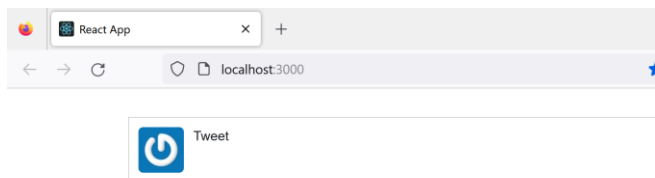
```

```
);  
}
```

Donnez simplement un peu de style à Avatar, dans index.css :

```
.avatar {  
width: 48px;  
height: 48px;  
border-radius: 5px;  
margin-right: 10px;  
}
```

Nous obtenons :



Ensuite, nous allons créer deux autres composants, Message et NameWithHandle :

```
function Message() {  
  return (  
    <div className="message">  
      This is less than 140 characters.  
    </div>  
  );  
}  
  
function NameWithHandle() {  
  return (  
    <span className="name-with-handle">  
      <span className="name">Your Name</span>  
      <span className="handle">@yourhandle</span>  
    </span>  
  );  
}
```

Montez ces nouveaux composants dans le composant principal Tweet :

```

function Tweet() {
  return (
    <div className="tweet">
      <Avatar/>
      <div className="content">
        <NameWithHandle/>
        <Message/>
      </div>
    </div>
  );
}

```

Ajoutez ces styles :

```

.name {
  font-weight: bold;
  margin-bottom: 0.5em;
  margin-right: 0.3em;
}

.handle {
  color: #8899a6;
  font-size: 13px;
}

```

On obtient plus ou moins :



Ajoutez ces composants :

```

const Time = () => (
  <span className="time">Il y a 3 heures</span>
);

const ReplyButton = () => (

```

```

<i className="fa fa-reply reply-button"/>
);
const RetweetButton = () => (
<i className="fa fa-retweet retweet-button"/>
);
const LikeButton = () => (
<i className="fa fa-heart like-button"/>
);
const MoreOptionsButton = () => (
<i className="fa fa-ellipsis-h more-options-button"/>
);

```

Ajoutez ces composants dans le composant Tweet comme suit :

```

function Tweet() {
  return (
    <div className="tweet">
      <Avatar/>
      <div className="content">
        <NameWithHandle/><Time/>
        <Message/>
        <div className="buttons">
          <ReplyButton/>
          <RetweetButton/>
          <LikeButton/>
          <MoreOptionsButton/>
        </div>
      </div>
    </div>
  );
}

```

Enfin ajoutez ces styles :

```

.time {

```

```
padding-left: 0.3em;
color: #8899a6;
}
.time::before {
content: "\00b7";
padding-right: 0.3em;
}
.buttons {
margin-top: 10px;
margin-left: 2px;
font-size: 1.4em;
color: #aab8c2;
}
.buttons i {
width: 80px;
}
```

Nous devons obtenir :

