

## Introduction aux formulaires

Nous allons maintenant introduire la notion de formulaire. Nous verrons plus tard la gestion des formulaires et des erreurs dans un contexte plus industrialisé en utilisant le module **Formik**.

En HTML les éléments de formulaire tels que input, textarea et select conservent leur propre état et sont mis à jour en fonction des données passées par l'utilisateur.

Dans React la gestion des “mutables” passera ici par un state avec setState pour sa mise à jour.

### Exemple de formulaire :

```
class UserForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = { value: '' };
  }

  handleChange = (event) => {
    this.setState({value: event.target.value});
  }

  handleSubmit = (event) => {
    console.log( `New User : ${this.state.value}` );
    event.preventDefault();
  }

  render() {
    return (
      <form onSubmit={this.handleSubmit}>
        <label>
          Username:
          <input type="text" value={this.state.value} onChange={this.handleChange} />
        </label>
        <input type="submit" value="Add" />
      </form>
    );
  }
}
```

## 1. Exercice formulaire User

Créez un formulaire pour ajouter à une liste user des utilisateurs, vous forcerez la saisie en majuscule dans le champ de saisie.

Vérifiez que l'utilisateur n'existe pas déjà dans la liste, si l'utilisateur se trompe. Alors affichez un message d'avertissement.

Gérez également le cas où le champ est vide dans l'ajout d'un utilisateur.

Affichez la liste sous le formulaire (Components conditionnels).

### Textarea

La particularité d'un textarea en HTML c'est qu'il définit son contenu texte par une structure de type composition. React utilise l'attribut value pour définir la valeur d'un élément, vous utiliserez donc la syntaxe suivante avec cet élément :

```
<textarea value={this.state.value} onChange={this.handleChange} />
```

### Select

En HTML classique on écrit :

```
<select>
  <option selected value="devReact">Alan</option>
  <option value="devSymfony">Alice</option>
  <option value="devDjango">Naoudi</option>
  <option value="devAngular">Fenley</option>
</select>
```

En React vous écrirez la syntaxe suivante :

```
// constructor
this.state = {value: 'devReact'};

// render
<select value={this.state.value} onChange={this.handleChange}>
  <option value="devReact">Alan</option>
  <option value="devSymfony">Alice</option>
  <option value="devDjango">Naoudi</option>
  <option value="devAngular">Fenley</option>
</select>
```

## Select multiple

Pour la gestion d'une sélection multiple, vous pouvez passer un array comme suit :

```
<select multiple={true} value={['a', 'b', 'c']}>
```

La gestion de l'upload des fichiers sera vue plus tard dans le cours.

## Gestion des inputs multiples

On peut évidemment gérer les différentes valeurs des champs du formulaire en s'inspirant de ce que l'on vient de faire, mais dans ce cas la gestion des variables peut devenir compliquée selon la longueur du formulaire.

Vous pouvez cependant utiliser la syntaxe suivante pour contrôler la saisie des valeurs de plusieurs champs :

```
const fields = { username : '', email : '' };

this.setState(
  { [name] : {...fields} }
)
```

## 2. Exercice formulaire de connexion

Récupérez le code ci-dessous et gérez une connexion **login/password** en React. Implémentez la gestion des erreurs.

```
<!-- source bootstrap Twitter CSS v4.0 -->
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" />
    <small id="emailHelp" class="form-text text-muted">We'll never share your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password" />
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Voici différentes vues du projet pour vous aider :

Email address

Enter email

Password

Password

Submit

Figure 1: Login password 1

Email address

alan@alan.fr

Email or password invalid

Password

Password

Submit

Figure 2: Login password 2

---

Success !

Email address

alan@alan.fr

Password

.....

Submit

Figure 3: Login password 3