

Stimulus JS

Stimulus est un framework JavaScript léger et centré sur les comportements, développé par Basecamp. Il est conçu pour faciliter l'ajout de comportements interactifs à des éléments HTML individuels, en suivant le modèle de conception du contrôleur de vue (MVC). Voici un cours de base sur Stimulus.js :

Introduction à Stimulus.js :

Stimulus simplifie l'ajout de comportements interactifs à votre interface utilisateur en utilisant une approche déclarative basée sur les attributs HTML. Il permet d'organiser le code JavaScript en "contrôleurs", qui sont liés à des éléments HTML spécifiques.

Installation de Stimulus.js :

Vous pouvez ajouter Stimulus à votre projet via npm ou en utilisant la version CDN. Pour npm, utilisez la commande :

```
bash
```

```
npm install stimulus
```

Structure de base d'un contrôleur Stimulus :

```
javascript
```

```
// app/javascript/controllers/hello_controller.js
import { Controller } from 'stimulus';

export default class extends Controller {
  connect() {
    // Code exécuté lorsque le contrôleur est connecté à l'élément DOM
    console.log("Hello from Stimulus!");
  }

  // Autres actions et méthodes du contrôleur ici
}
```

Utilisation des contrôleurs Stimulus dans le HTML :

```
html
```

```
<!-- app/views/pages/index.html.erb -->
<div data-controller="hello">
  <h1 data-target="hello.output">Hello, Stimulus!</h1>
  <input type="text" data-target="hello.name" placeholder="Your name">
  <button data-action="click->hello#greet">Greet</button>
</div>
```

Configuration globale de Stimulus :

Pour activer les contrôleurs Stimulus sur votre application, vous devez importer et initialiser Stimulus dans votre application JavaScript.

javascript

```
// app/javascript/packs/application.js
import { Application } from 'stimulus';
import { definitionsFromContext } from 'stimulus/webpack-helpers';

const application = Application.start();
const context = require.context('../controllers', true, /\.js$/);
application.load(definitionsFromContext(context));
```

Actions et cibles Stimulus :

Stimulus utilise le concept d'actions et de cibles. Une action est un événement déclenché sur un élément, tandis qu'une cible est un élément sur lequel l'action a un impact.

javascript

```
// app/javascript/controllers/hello_controller.js
import { Controller } from 'stimulus';

export default class extends Controller {
  static targets = ['output', 'name'];

  greet() {
    const name = this.nameTarget.value || "Stranger";
    this.outputTarget.textContent = `Hello, ${name}!`;
  }
}
```

Utilisation des attributs de valeur avec Stimulus :

html

```
<!-- app/views/pages/index.html.erb -->
<div data-controller="hello" data-hello-name-value="John">
  <h1 data-target="hello.output">Hello, Stimulus!</h1>
  <input type="text" data-target="hello.name" placeholder="Your name">
  <button data-action="click->hello#greet">Greet</button>
</div>
```

Les hooks Stimulus :

Stimulus propose des hooks pour gérer le cycle de vie des contrôleurs, tels que `initialize`, `connect`, `disconnect`, etc.

javascript

```
// app/javascript/controllers/hello_controller.js
import { Controller } from 'stimulus';

export default class extends Controller {
  initialize() {
    console.log("Controller initialized");
  }

  connect() {
    console.log("Controller connected");
  }
}
```

```
    disconnect() {  
        console.log("Controller disconnected");  
    }  
}
```

Conclusion :

Ceci est une introduction de base à Stimulus.js. Stimulus simplifie la gestion des interactions côté client en fournissant une structure organisée pour ajouter des comportements interactifs à des éléments HTML spécifiques. Pour des détails plus approfondis, consultez la documentation officielle de Stimulus.js et explorez les fonctionnalités avancées telles que les actions, les cibles, et les hooks.