**VUE**

**Dure totale du module : 21H00**

Table des matières

[Objectifs pédagogiques : 3](#_Toc111650891)

[Création d’une première application vue (mode cdn) 4](#_Toc111650892)

[Réalisation d’une application web TaskList en Javascript 4](#_Toc111650893)

[Exercice : Créer une TaskList en JS 4](#_Toc111650894)

[Application web TaskList en VUE 6](#_Toc111650895)

[Notions / Concept de base de VUE 8](#_Toc111650896)

# Objectifs pédagogiques :

Être capable de comprendre les enjeux des Frameworks

Être capable d'initialiser et paramétrer un projet grâce à un Frameworks front-end

Être capable de réaliser des interface utilisateur grâce à un Frameworks front-end

~~Être capable de mettre en place un système de Routing (url)~~

Être capable de gérer des requêtes HTTP (BDD)

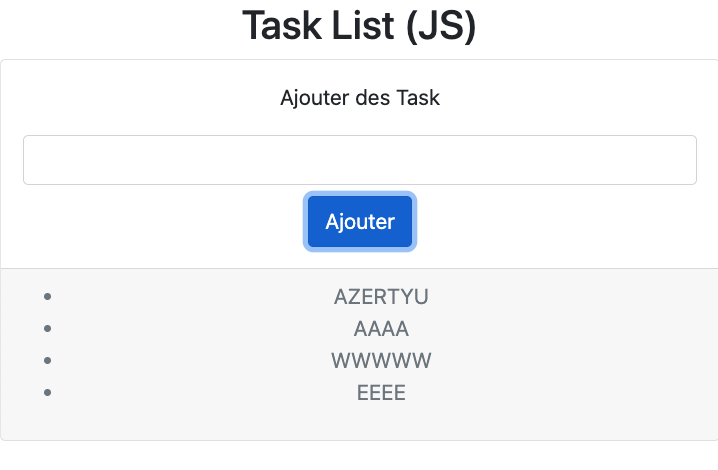
Être capable de connecter une application à une base de données

# Création d’une première application vue (mode cdn)

Pour illustrer plus explicitement l’avantage d’utiliser un Framework nous allons créer une application en Javascript (classique), puis nous allons voir les différences avec cette même application en VUE.

## Réalisation d’une application web TaskList en Javascript

### Exercice : Créer une TaskList en JS



Créer une page HTML (liée à un fichier JS) qui comporte :

* 1 input
* 1 button
* 1 <ul>

En JS :

* 1 variable leInput récupère l’élément input
* 1 variable leButton récupère l’élément button
* 1 variable laList récupère l’élément ul
* 1 fonction ajouterTask dans laquelle :
  + on créer un élément <li>
  + on assigne à la propriété textContent (ou innerText) du <li>, la value de leInput
  + on place le <li> dans laListe avec la fonction append()

[SOLUTION taskListJS (StackBlitz)](https://stackblitz.com/edit/web-platform-cxqwxr?file=index.html)  
[SOLUTION taskListJS (GitHub)](https://github.com/jefff404/taskList-JS-)

Code JS de l’application :

//On stock les éléments HTML dans des variables

const leButton = document.querySelector('button');

const leinput = document.querySelector('input');

const laListe = document.querySelector('ul');

function ajouterTask(){

//Dans inputContent on stock ce que l'utilisateur tape dans l'input

const inputContent = leinput.value;

//Création d'un élément <li></li>

const listElement = document.createElement('li');

//Complétion du texte de l'élément <li></li>

listElement.textContent = inputContent;

//Ajout du <li></li> dans la <ul></ul> (laListe)

laListe.append(listElement);

//Bonus : apès avoir ajouté une Task, on vide l'input

leinput.value = '';

}

leButton.addEventListener('click', ajouterTask);

Avec Javascript, nous pouvons le constater, il est nécessaire de prendre le contrôle de tous les éléments du DOM.

L’utilisation de Vue va simplifier la syntaxe JS en permettant d’avoir des interactions directement dans la partie Template (côté HTML).

## Application web TaskList en VUE

Fichier JavaScript (mode VUE)

//Création d'une instance Vue c'est notre application

Vue.createApp({

//Dans une fonction data on va retourner un objet qui contiendra les variables de l'application

//Toutes les data seront contenues dans this

data(){

return {

tasks: [],

valeurDeInput:''

}

},

// Dans cet Objet methods, on va écrire nos fonctions

methods:{

ajouterTask(){

this.tasks.push(this.valeurDeInput);

this.valeurDeInput = '';

}

}

// L'application est placée sur la balise HTML qui possède l'id app

}).mount('#app');

Page HTML (body)



Via son système de directives (v-on, v-model, v-for) Vue permet une interaction plus directe entre les variables et fonctions du fichier JS et le Template HTML.

# Notions / Concept de base de VUE

L’exemple précèdent (application taskList) nous donne un aperçu des principaux concepts et règles de VUE.

Côté Javascript, une application VUE comportera toujours des fonctions et objets de base :

* La fonction createApp() : va permettre de créer l’application VUE.
* La fonction data() : qui retournera toujours un objet qui dans lequel on pourra définir les variables de l’application
* L’objet methods : {} : dans lequel on va écrire les fonctions de l’application
* La fonction mount() : qui va permettre de placer noter application sur une balise du Template html.

Côté HTML, VUE nous permet d’utiliser le système de directive (v-on, v-model, v-for …), utilisables comme des attribut HTML ainsi qu’une syntaxe pour afficher directement des données   
{{leNomDuneVariable}}.

VUE facilite donc le développement d’application Web en proposant une interaction enrichie côté Template ce qui simplifie le code Javascript.