|  |  |
| --- | --- |
|  | **Année universitaire**  **2024-2025** |
| **Workshop:**  **Atelier - intégration REST API** | |
|  | |

**Objectifs :**

* Comprendre la différence entre une API et un bundle.
* Intégrer une REST API de traduction dans un projet symfony.
* Intégrer un Bundle de traduction dans un projet symfony.

**Prérequis :**

* Avoir un projet Symfony6 configuré.
* Avoir python dans la machine afin de tester le script de traduction.

**Partie 1 : Intégrer une REST API de traduction dans un projet Symfony**

* **Etape 1 :**

Mettre à disposition de l’utilisateur une liste déroulante contenant les langues désirées.

<select id="language-selector">

<option value="en">English</option>

<option value="fr">French</option>

<option value="ar">Arabic</option>

</select>

* **Etape 2 :**

Affecter des id aux éléments qu’on désire traduire.

<h1 **id='title'**>Se former autrement</h1>

<p **id="description"**>"ESPRIT" lors de sa création s'est donnée comme mission principale de former des ingénieurs « autrement ».

Pour cela l'école a adopté dès le départ une pédagogie nouvelle, dynamique mettant l'élève au centre de l'apprentissage. L'adoption de l'APP (apprentissage par projet/problème) est venue renforcer cette volonté. Un centre d'innovation pédagogique permet la formation des enseignants à ces nouvelles pratiques pédagogiques qui s'étendent petit à petit à l'ensemble du cursus. </p>

* **Etape 3 :**

Créer la fonction permettant d’envoyer la requête.

async function translateText(text, targetLang) {

const requestBody = {

texts: [text],

target\_language: targetLang

};

const response = await fetch('http://127.0.0.1:5000/translate', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify(requestBody),

});

if (!response.ok) {

const errorData = await response.json();

console.error('API Error:', errorData);

throw new Error(`API Error: ${response.status} - ${response.statusText}`);

}

const data = await response.json();

return data.translated\_texts[0];

}

* **Etape 4 :**

Faire **l’écoute** sur le document afin de détecter le changement de la langue sélectionnée et affecter les textes traduits dans les champs adéquats.

document.getElementById('language-selector').**addEventListener**('change', async function () {

const lang = this.value;

const title = document.getElementById('title').innerText;

const description = document.getElementById('description').innerText;

try {

const translatedTitle = await translateText(title, lang);

document.getElementById('title').innerText = translatedTitle;

const translatedDescription = await translateText(description, lang);

document.getElementById('description').innerText = translatedDescription;

} catch (error) {

console.error('Translation failed:', error);

alert('Translation failed. Please check the console for details.');

}

});

* **Etape 5 :**

Lancer le script python « translate\_WS.py » avec la commande : **python translate\_WS.py**

* **Etape 6 :** Tester.

**A screenshot of a computer

AI-generated content may be incorrect.**