
1. Constants and DOM Element Selection

```
```javascript
const Base_URL = "https://cdn.jsdelivr.net/npm/@fawazahmed0/currency-api@latest/v1/
currencies";
const fromOption = document.querySelectorAll(".converter select");
const btn = document.querySelector(".btn-converter button");
const fromCurr = document.querySelector(".from select");
const toCurr = document.querySelector(".to select");
const msg = document.querySelector(".converted p");
```
```

- **Base_URL**: This is the base URL for the currency API. It provides the latest exchange rates for various currencies.
- **fromOption**: Selects all `<select>` elements inside the `.converter` container. These are the dropdowns for selecting the "from" and "to" currencies.
- **btn**: Selects the button that triggers the currency conversion.
- **fromCurr**: Selects the `<select>` element for the "from" currency.
- **toCurr**: Selects the `<select>` element for the "to" currency.
- **msg**: Selects the `<p>` element where the converted amount will be displayed.

2. Populating Currency Options

```
```javascript
window.addEventListener("load", () => {
 for (let select of fromOption) {
 for (let curr in countryList) {
 let newOption = document.createElement("option");
 newOption.innerText = curr;
 newOption.value = curr;
 if (select.name === "from" && curr === "USD") {
 newOption.selected = "selected";
 } else if (select.name === "to" && curr === "INR") {
 newOption.selected = "selected";
 }
 select.append(newOption);
 }
 select.addEventListener("change", (evt) => {
 updateFlag(evt.target);
 });
 }
 updateExchangeRate();
});
```
```

- When the page loads (`window.addEventListener("load", ...)`), the script populates the dropdowns (`<select>` elements) with currency options.
- It iterates over a `countryList` object (assumed to be defined elsewhere) that maps currency codes (e.g., `USD`, `INR`) to country codes (e.g., `US`, `IN`).
- For each currency, a new `<option>` element is created and added to the dropdowns.
- The default selected options are:
 - "From" currency: `USD` (US Dollar).
 - "To" currency: `INR` (Indian Rupee).
- An event listener is added to each dropdown to update the flag image when the currency is changed.

3. Updating the Flag Image

```
```javascript
const updateFlag = (element) => {
 const currCode = element.value;
 const countryCode = countryList[currCode];
 const newSrc = `https://flagsapi.com/${countryCode}/flat/64.png`;
 const img = element.parentElement.querySelector(".converter img");
 img.src = newSrc;
};
```
```

- This function updates the flag image next to the dropdown when the user selects a new currency.
- It uses the `countryList` object to map the selected currency code (`currCode`) to a country code (`countryCode`).
- The flag image is fetched from `https://flagsapi.com/` using the country code.

4. Handling the Conversion Button Click

```
```javascript
btn.addEventListener("click", (evt) => {
 evt.preventDefault();
 updateExchangeRate();
});
```
```

- When the user clicks the conversion button, the `updateExchangeRate` function is called to fetch the latest exchange rate and perform the conversion.
- `evt.preventDefault()` prevents the default form submission behavior.

5. Fetching and Displaying the Exchange Rate

```
```javascript
const updateExchangeRate = async () => {
 const amount = document.querySelector(".inputv input");
 let amtValue = amount.value;
 if (amtValue === "" || amtValue < 1) {
 amtValue = 1;
 amount.value = "1";
 }

 const URL = `${Base_URL}/${fromCurr.value.toLowerCase().json}`;

 try {
 const response = await fetch(URL);
 if (!response.ok) {
 throw new Error(`Error: ${response.status}`);
 }
 const data = await response.json();
 console.log(data);
 const rate = data[fromCurr.value.toLowerCase()][toCurr.value.toLowerCase()];
 const finalAmount = amtValue * rate;
 console.log(finalAmount);
 msg.innerText = `${amtValue} ${fromCurr.value} = ${finalAmount.toFixed(2)} ${toCurr.value}`;
 } catch (error) {
 console.error("Error fetching exchange rate:", error);
 msg.innerText = "Failed to fetch exchange rate. Please try again later.";
 }
};
```
```

- **Step 1**: Get the amount entered by the user. If the amount is empty or less than 1, it defaults to `1`.
- **Step 2**: Construct the API URL using the selected "from" currency.
- **Step 3**: Fetch the exchange rate data from the API.
 - If the response is not successful, an error is thrown.
- **Step 4**: Extract the exchange rate for the selected "to" currency from the API response.
- **Step 5**: Calculate the converted amount by multiplying the user's input amount by the exchange rate.
- **Step 6**: Display the result in the `

` element (msg).
- **Error Handling**: If there's an error (e.g., network issue, invalid response), an error message is displayed.

How It Works

1. When the page loads, the dropdowns are populated with currency options, and default values are set.
2. The user selects the "from" and "to" currencies and enters an amount.
3. When the user clicks the conversion button, the script fetches the latest exchange rate from the API.
4. The script calculates the converted amount and displays it on the page.
5. If the user changes the currency, the flag image is updated accordingly.

Assumptions

- The `countryList` object is defined elsewhere and maps currency codes to country codes.
- The API returns exchange rates in the format:

```
```json
{
 "usd": {
 "inr": 75.0,
 "eur": 0.85,
 ...
 }
}
```

---

### ### **Improvements**

1. **Input Validation**: Ensure the user enters a valid number.
2. **Caching**: Cache exchange rates to reduce API calls.
3. **Error Handling**: Provide more detailed error messages.
4. **UI Enhancements**: Add a loading spinner while fetching data.