

# Praktisches Labor: API aufrufen, um Wetterdaten mit fetch() abzurufen

**Geschätzte benötigte Zeit:** 30 Minuten

## Was Sie lernen werden

In diesem Labor integrieren Sie eine externe API **OpenWeatherMap** in eine Webanwendung mithilfe von JavaScript. Sie lernen, wie man Daten asynchron abrufen, JSON-Antworten analysiert und die Webseite dynamisch basierend auf den empfangenen Informationen aktualisiert. Sie werden die Grundkonzepte der API-Integration, asynchroner Operationen, DOM-Manipulation und Benutzerinteraktion verstehen, was Ihnen ein grundlegendes Verständnis der Webentwicklung für die Nutzung externer Datenquellen in einer einfachen Benutzeroberfläche vermittelt.

## Lernziele

Nach Abschluss dieses Labors werden Sie in der Lage sein:

- **Benutzerfreundlicher Wetterabruf:** Benutzereingaben für Städtenamen zu ermöglichen, um den Abruf von Echtzeit-Wetterinformationen über eine intuitive Weboberfläche zu erleichtern.
- **API-Integration für Wetterdaten:** Die OpenWeatherMap API zu nutzen, um präzise Wetterdaten basierend auf von Benutzern eingegebenen Städten abzurufen und Temperatur sowie Wetterbeschreibungen dynamisch auf der Webseite anzuzeigen.
- **Verarbeitung von HTML-Formularübermittlungen und Implementierung von JS-Ereignissen:** Formularübermittlungen innerhalb von HTML zu verwalten und Ereignis-Listener in JavaScript zu implementieren, um reibungslose Benutzerinteraktionen und Datenabrufprozesse zu gewährleisten.
- **Demonstration asynchroner Anfragen und dynamischer DOM-Updates:** Die praktische Anwendung asynchroner Anfragen mit fetch() zu zeigen, JSON-Antworten zu analysieren und den DOM dynamisch zu aktualisieren, um abgerufene Wetterdetails nahtlos anzuzeigen, ohne dass eine Seitenaktualisierung erforderlich ist.

## Voraussetzungen

- Grundkenntnisse in HTML.
- Webbrowser mit einer Konsole (Chrome DevTools, Firefox Console usw.).

## Schritt 1: Die Umgebung einrichten

1. Zunächst müssen Sie Ihr Hauptrepository in der **Skills Network Umgebung** klonen, die Sie im ersten Labor erstellt haben und in die Sie alle Ihre vorherigen Labor-Dateien und -Ordner hochgeladen haben. Befolgen Sie die angegebenen Schritte:
  - Klicken Sie dazu auf das Terminal im oberen rechten Fensterbereich und wählen Sie dann **Neues Terminal**.
  - Führen Sie den `git clone` Befehl aus, indem Sie den angegebenen Befehl im Terminal eingeben.

```
git clone <github-repository-url>
```

**Hinweis:** Setzen Sie Ihren eigenen GitHub-Repository-Link anstelle von `<github-repository-url>` ein.

- Der obige Schritt wird den Ordner für Ihr GitHub-Repository im Projektordner im Explorer klonen. Sie werden auch mehrere Ordner im geklonten Ordner sehen.
- Jetzt müssen Sie in den geklonten Ordner navigieren. Geben Sie dazu den angegebenen Befehl im Terminal ein:

```
cd <repository-folder-name>
```

**Hinweis:** Schreiben Sie Ihren geklonten Ordernamen anstelle von `<repository-folder-name>`. Führen Sie `git clone` aus, wenn Sie sich von der **Skills Network Umgebung** abgemeldet haben und nach der Anmeldung keine Dateien oder Ordner sehen können.

- Wählen Sie nun den Ordner **klonierter Ordnername**, klicken Sie mit der rechten Maustaste darauf und wählen Sie **Neuer Ordner**. Geben Sie den Ordernamen als **weatherReport** ein. Dies erstellt den Ordner für Sie. Wählen Sie dann den Ordner **weatherReport**, klicken Sie mit der rechten Maustaste und wählen Sie **Neue Datei**. Geben Sie den Dateinamen als **weather\_report.html** ein und klicken Sie auf OK. Dies erstellt Ihre HTML-Datei.
- Wählen Sie nun erneut den Ordner **weatherReport**, klicken Sie mit der rechten Maustaste und wählen Sie **Neue Datei**. Geben Sie den Dateinamen als **weather\_report.js** ein und klicken Sie auf OK. Dies erstellt Ihre JavaScript-Datei.
- Um Daten von der API abzurufen, benötigen Sie Ihren persönlichen **API-Schlüssel** von der jeweiligen Website, von der Sie die Anfrage zum Abrufen von Daten generieren werden.
- Dazu müssen Sie die angegebene Website <https://openweathermap.org/> öffnen und sich für diese Website anmelden.
- Nach der Anmeldung werden Sie auf die Dashboard-Seite weitergeleitet, auf der Sie das Menüelement **API-Schlüssel** anklicken müssen.
  - Hinweis: Wenn Sie nicht auf die oben genannte Seite weitergeleitet werden, können Sie auch darauf zugreifen, indem Sie auf Ihr Profil-Menüelement in der Navigationsleiste klicken und dann **Meine API-Schlüssel** auswählen, wie im angegebenen Screenshot gezeigt.
- Nachdem Sie die Seite für API-Schlüssel aufgerufen haben, gelangen Sie zu einer Seite, auf der die Website bereits einen persönlichen Schlüssel für Sie generiert hat. Dieser Schlüssel spielt eine entscheidende Rolle beim Abrufen von Daten von einer bestimmten URL.

Hinweis: Der persönliche Schlüssel fungiert als Passwort für Ihre URL, von der Sie die gewünschten Wetterdaten erhalten.

- Jetzt benötigen Sie auch die URL, mit deren Hilfe Daten für eine bestimmte Stadt abgerufen werden. Klicken Sie dazu auf **Dienste**, wie im angegebenen Screenshot gezeigt.
- Dann müssen Sie auf **anzeigen** klicken.
- Sobald Sie auf anzeigen klicken, öffnet sich ein neues Fenster und navigiert zu einer neuen Seite. Sie müssen ein wenig nach unten scrollen, wo Sie die Option sehen, die API kostenlos zu nutzen. Von dort aus müssen Sie auf **Aktuelles Wetter** klicken.
- Es öffnet sich erneut ein neues Fenster und navigiert Sie zu einer neuen Seite. Scrollen Sie erneut nach unten, wo Ihnen mehrere URL-Links bereitgestellt werden, die Sie zum Abrufen von Daten verwenden können.
- Wählen Sie von der aktuellen Seite die URL aus, die rot hervorgehoben ist. Sie werden diese URL auch in den kommenden Anweisungen verwenden, um wetterbezogene Daten abzurufen.
- Ein API-Schlüssel ist wie ein geheimer Zugangscode, und eine URL ist die Webadresse, von der wir die Informationen erhalten.

## HTML-Struktur erstellen

- Erstellen Sie die grundlegende Vorlagenstruktur einer HTML-Datei, indem Sie den bereitgestellten Inhalt hinzufügen.

```
<!DOCTYPE html>
<html>
<head>
  <title>Weather Report</title>
</head>
<body>
  <h1>Weather Report</h1>
  <form id="weatherForm">
    <label for="city">Enter City:</label>
    <input type="text" id="city" name="city">
    <button type="submit">Get Weather</button>
  </form>
  <div id="weatherInfo"></div>
<script src="./weather_report.js"></script>
</body>
</html>
```

**Hinweis:** Wenn Sie den Code eingefügt haben, speichern Sie Ihre Datei.

- Der obige Code hat Inhalte und Elemente gegeben:

- `<form>` Tag: Der `<form>` Tag erstellt einen Abschnitt, der Eingabeelemente und einen Absende-Button enthält. Er hat ein `id`-Attribut, das auf "weatherForm" gesetzt ist.
- `<label>` Tag: Dieses Tag erstellt ein Label für das Eingabefeld mit dem Text "Stadt eingeben:", der vom Label angezeigt wird.
- `<input>` Tag: Es wird verwendet, um den vom Benutzer eingegebenen Stadtnamen zu erfassen.
- `<button>` Tag: Dieser Button zeigt an, dass dieser Button das Formular beim Klicken absendet.
- `<div>` Tag: `id="weatherInfo"`: Ein leeres `<div>` Element mit einem Bezeichner. Hier werden die Wetterinformationen angezeigt.
- `<script>` Tag: Dies verweist auf eine externe JavaScript-Datei namens "weather\_report.js", die in die aktuelle "weather.html" Datei eingebunden werden soll.

## Schritt 2: Variablen und Funktionen definieren

1. Erstelle eine Funktion mit dem Namen **showweatherDetails** und füge sie in die Datei **weather\_report.js** ein.

```
function showweatherDetails(event) {
    event.preventDefault();
}
```

- Der obige Code verhindert das Standardverhalten eines Ereignisses, wie z.B. das Absenden eines Formulars, innerhalb einer Funktion namens **showweatherDetails**.

2. Initialisiere innerhalb der Funktion **showweatherDetails** drei Variablen für Stadt, apiKey und apiUrl wie folgt:

```
const city = document.getElementById('city').value;
const apiKey = 'c4f86ece00bc8aa272652ac9065af12d'; // Ersetze 'YOUR_API_KEY' durch deinen tatsächlichen API-Schlüssel
const apiUrl = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`;
```

- Der obige Code erhält den Wert, den der Benutzer im Eingabefeld mit der Bezeichnung 'city' eingegeben hat.
- Eine URL für die OpenWeatherMap API wurde erstellt, indem der vom Benutzer eingegebene Stadtname mit einem persönlichen API-Schlüssel kombiniert wurde, der für den Zugriff auf Wetterdaten erforderlich ist. Der Schlüssel wurde bereits für dich erstellt. Du kannst diesen Schlüssel direkt verwenden.
  - Hinweis: Du wirst deinen eigenen API-Schlüssel verwenden, der von dir im ersten Schritt von **Umgebung einrichten** generiert wurde.
- Dieser Code hat den Variablennamen **apiUrl**, der die gleiche URL enthalten wird, die du im ersten Schritt von **Umgebung einrichten** speichern sollst.

3. Verwende die Fetch-API-Methode, um Details zur Stadt abzurufen, die der Benutzer im bereitgestellten Eingabefeld in der HTML-Datei eingeben wird. Füge den folgenden Code innerhalb der Funktion **showweatherDetails** unter der Variableninitialisierung ein.

```
fetch(apiUrl)
    .then(response => response.json())
    .then(data => {
        const weatherInfo = document.getElementById('weatherInfo');
        weatherInfo.innerHTML = `<h2>Weather in ${data.name}</h2>
                                <p>Temperature: ${data.main.temp} &#8451;</p>
                                <p>Weather: ${data.weather[0].description}</p>`;
    })
```

4. Die oben genannte Funktion `fetch` initiiert eine asynchrone HTTP-Anfrage an die angegebene apiUrl (OpenWeatherMap API), um Wetterdaten abzurufen.

- Die Antwortverarbeitung erfolgt mithilfe von Promises mit **.then()**, um die Antwort zunächst in das JSON-Format **response.json()** zu konvertieren und dann auf die resultierenden Daten zuzugreifen.
- Die HTML-Datei wird ebenfalls aktualisiert, indem der Inhalt dynamisch geändert wird, indem das Element 'weatherInfo' ausgewählt und mit strukturierten Wetterinformationen wie dem Städtenamen **data.name**, der Temperatur **data.main.temp** und der Wetterbeschreibung **data.weather[0].description** gefüllt wird, um eine benutzerfreundliche Anzeige der abgerufenen Wetterdetails auf der Webseite zu gewährleisten.
- Fügen Sie dann den folgenden Code außerhalb der Funktion ein. Er fügt dem Element 'weatherForm' einen Ereignis-Listener hinzu, der auf ein 'submit'-Ereignis hört und die Funktion showweatherDetails beim Absenden des Formulars auslöst, um eine angepasste Handhabung oder Manipulation des Verhaltens des Formulars zu ermöglichen.

```
document.getElementById('weatherForm').addEventListener('submit', showweatherDetails );
```

## Schritt 3: Überprüfen Sie die Ausgabe

- Um Ihre HTML-Seite anzuzeigen, klicken Sie mit der rechten Maustaste auf die **weather\_report.html**-Datei, nachdem Sie diese Datei ausgewählt haben, und wählen Sie "Öffnen mit Live Server".
- Der Server sollte auf Port 5500 gestartet werden, was durch eine Benachrichtigung auf der unteren rechten Seite angezeigt wird.
- Klicken Sie auf die Schaltfläche Skills Network links an Position 1. Es öffnet sich das „Skills Network Toolbox“. Klicken Sie dann auf „Sonstiges“ und wählen Sie an Position 2 „Anwendung starten“. Von dort geben Sie die Portnummer 5500 an Position 3 ein und klicken Sie auf diese Schaltfläche .
- Ihr Standardbrowser öffnet sich, in dem Sie den Ordernamen **cloned-folder-name** sehen. Klicken Sie auf diesen Ordernamen **cloned-folder-name** und dann auf den Ordernamen **weatherReport**. Sie sehen Dateien, die zu diesem Ordner gehören, wo Sie erneut auf die **weather\_report.html**-Datei klicken, wie unten gezeigt.
- Es öffnet sich die Startseite und Sie sehen die Ausgabe wie unten gezeigt.
- Geben Sie einen beliebigen Stadtnamen in das Eingabefeld ein und klicken Sie dann auf die Schaltfläche **Get Weather**, und Sie werden das Ergebnis sehen.
- Sie können auch die Catch-Methode einfügen, um Fehler abzufangen, wenn der Benutzer einen falschen Stadtnamen eingibt. Fügen Sie den folgenden Code vor dem Ende der Funktion ein.

```
.catch(error => {
    console.error('Fehler beim Abrufen des Wetters:', error);
    const weatherInfo = document.getElementById('weatherInfo');
    weatherInfo.innerHTML = `

```

**Hinweis:** Speichern Sie Ihre Datei, nachdem Sie den Code eingefügt haben. Sie können jede Methode zum Speichern verwenden. Wenn Sie weitere Änderungen an Ihrem Code vornehmen, aktualisieren Sie einfach Ihren Browser, der auf Port 5500 läuft. Dadurch entfällt die Notwendigkeit, die Anwendung wiederholt neu zu starten.

## Schritt 4: Führen Sie Git-Befehle aus

### Wichtiger Hinweis: Bevor Sie auf GitHub pushen:

Laden Sie Ihren echten API-Schlüssel nicht hoch!  
Ersetzen Sie ihn durch 'YOUR\_API\_KEY' oder löschen Sie ihn vorerst.  
So bleibt Ihr Schlüssel sicher und Missbrauch wird vermieden.

- Führen Sie `git add` aus, um die neuesten Dateien und Ordner hinzuzufügen, indem Sie den angegebenen Befehl im Terminal in der Git-Umgebung eingeben.

```
git add --a
```

Stellen Sie sicher, dass das Terminal den folgenden Pfad hat:

2. Führen Sie dann `git commit` im Terminal aus. Während Sie `git commit` ausführen, kann das Terminal eine Nachricht anzeigen, um Ihre `git config --global` für Benutzernamen und Benutzer-E-Mail einzurichten. Wenn ja, müssen Sie auch den `git config`-Befehl für `user.name` und `user.email` wie angegeben ausführen.

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Ihr Name"
```

**Hinweis:** Ersetzen Sie die Daten in Anführungszeichen durch Ihre eigenen Angaben.

Führen Sie dann den Commit-Befehl wie angegeben aus:

```
git commit -m "message"
```

3. Führen Sie dann `git push` aus, indem Sie einfach den angegebenen Befehl im Terminal eingeben.

```
git push origin
```

- Nach dem Push-Befehl wird das System Sie auffordern, Ihren Benutzernamen und Ihr Passwort einzugeben. Geben Sie den Benutzernamen für Ihr GitHub-Konto und das Passwort ein, das Sie im ersten Labor erstellt haben. Nach Eingabe der Anmeldedaten werden alle Ihre neuesten Ordner und Dateien in Ihr GitHub-Repository gepusht.

## Übungsaufgabe

1. Sie werden eine andere URL verwenden, um detaillierte Wetterberichte abzurufen. Zum Beispiel müssen Sie die URL wie im gegebenen Screenshot gezeigt verwenden, indem Sie die gleiche Seite aufrufen, von der Sie die URL verwenden, die in den Laboranweisungen angegeben ist.

Hinweis: Anstelle des Stadtnamens müssen Sie nun die Details für Breiten- und Längengrad vom Benutzer erfassen.

2. Dazu benötigen Sie zwei `<input>`-Felder, eines, um den Breitengrad und eines, um den Längengradwert vom Benutzer zu erhalten.

3. Sie müssen einen Submit-Button erstellen, nach dem Klicken auf den die Ergebnisse angezeigt werden.
4. Wenden Sie dann die Fetch-API-Methode an, um Daten zu den Wetterbedingungen für die Benutzereingabe abzurufen und erstellen Sie den entsprechenden JavaScript-Code.

## Zusammenfassung

1. **Wetterabrufformular:** Eine HTML-Seite mit einem Formular erstellt, das es Benutzern ermöglicht, einen Städtenamen einzugeben und beim Absenden Wetterdaten von OpenWeatherMap abzurufen.
2. **JavaScript-Funktionalität:** Eine JavaScript-Funktion **showweatherDetails** definiert, um die Formularübermittlung zu verarbeiten, das Standardverhalten zu verhindern, die eingegebene Stadt zu extrahieren und eine URL zu generieren, um Wetterdetails über die OpenWeatherMap API abzurufen.
3. **Dynamische Wetteranzeige:** JavaScript's `fetch()` verwendet, um Wetterdaten abzurufen, die Webseite dynamisch mit den abgerufenen Wetterinformationen zu aktualisieren, einschließlich Städtenamen, Temperatur in Celsius und Wetterbeschreibung, um eine nahtlose Anzeige der Wetterdetails ohne Seitenneuladen zu gewährleisten.

© IBM Corporation. Alle Rechte vorbehalten.