

# Über npm / cdn, um Axios zu erhalten

## Was ist Node.js?

Node.js nutzt die V8-JavaScript-Engine von Chrome, die es Entwicklern ermöglicht, JavaScript auf der Serverseite auszuführen und skalierbare sowie leistungsstarke Anwendungen zu erstellen. Eine seiner Hauptmerkmale ist das nicht-blockierende, ereignisgesteuerte Design, das es effizient für den Bau von Echtzeitanwendungen macht.

## So installieren Sie Node.js:

- **Node.js herunterladen:** Besuchen Sie die offizielle Node.js-Website und laden Sie den Installer für Ihr Betriebssystem von <https://nodejs.org/en> herunter.
- **Node.js installieren:** Befolgen Sie die Installationsanweisungen, um den Installer auszuführen.
- **Installation überprüfen:** Öffnen Sie ein Terminal oder die Eingabeaufforderung und geben Sie `node -v` ein, um die Installation von Node.js zu überprüfen. Sie sollten die installierte Versionsnummer sehen.

## Was ist npm?

npm steht für Node Package Manager. Es dient als Paketmanager für Node.js-Module und -Pakete. Npm installiert, verwaltet und teilt Codepakete aus dem Repository (das eine Sammlung von auf Servern gehosteten Paketen ist).

## So verwenden Sie npm:

- **Pakete installieren:** Um ein Paket zu installieren, verwenden Sie `npm install <package-name>` im Terminal eines beliebigen Code-Editors Ihrer Wahl. Zum Beispiel installiert `npm install express` das Express-Framework.
- **Pakete verwenden:** Nach der Installation können Sie diese Pakete in Ihrer Node.js-Anwendung mit `require()` einfügen. Zum Beispiel `const express = require('express')`.
- **Pakete verwalten:** Verwenden Sie npm-Befehle wie `npm uninstall <package-name>` zum Entfernen von Paketen, `npm update <package-name>` zum Aktualisieren von Paketen und `npm search <keyword>` zum Suchen von Paketen im npm-Registry.
- **Package.json:** Diese Datei enthält Metadaten über das Projekt und die Liste der Abhängigkeiten. Sie können sie manuell erstellen oder npm verwenden, um ein neues Projekt zu erstellen.

Node.js und npm sind integrale Bestandteile der modernen Webentwicklung geworden und ermöglichen es Entwicklern, robuste und skalierbare Anwendungen mit JavaScript zu erstellen.

## Axios

Axios ist eine JavaScript-Bibliothek, die hauptsächlich für das Durchführen von HTTP-Anfragen sowohl in Node.js-Umgebungen als auch in Webbrowsern verwendet wird. Sie bietet eine einfache und intuitive API zur Handhabung asynchroner HTTP-Anfragen. Axios unterstützt verschiedene Funktionen wie Interceptors, die Möglichkeit, Anfragen abubrechen, automatische JSON-Datenumwandlung und vieles mehr.

## Axios installieren

Zuerst stellen Sie sicher, dass Node.js und npm installiert sind. Dann können Sie Axios mit npm installieren:

```
npm install axios
```

## Verwendung von Axios in Node.js

Hier ist ein Beispiel, wie Sie eine einfache GET-Anfrage mit Axios in einem Node.js-Skript durchführen können:

```
// Import Axios
const axios = require('axios');
// Make a GET request
axios.get('https://jsonplaceholder.typicode.com/posts')
  .then(response => {
    // Handle successful response
    console.log('Response:', response.data);
  })
  .catch(error => {
    // Handle error
    console.error('Error:', error);
  });
```

## In diesem Beispiel:

- Wir importieren Axios mit `require('axios')`.
- Verwenden Sie `axios.get`, um eine GET-Anfrage an einen Beispiel-API-Endpunkt (<https://jsonplaceholder.typicode.com/posts>) zu senden.

- Der `.then`-Block verarbeitet die erfolgreiche Antwort, und die Daten werden in der Konsole protokolliert.
- Der `.catch`-Block fängt alle Fehler ab, die während der Anfrage auftreten können, und protokolliert sie in der Konsole.
- Sie können verschiedene HTTP-Methoden (GET, POST, PUT, DELETE usw.) mit Axios ausführen, indem Sie `axios.<method>` aufrufen (z. B. `axios.post`, `axios.put`, `axios.delete`) und deren jeweilige Antworten und Fehler mit `.then`- und `.catch`-Blöcken behandeln.

Denken Sie daran, dass Axios Promises zurückgibt, sodass Sie `async/await` verwenden können, um asynchrone Anfragen in einem synchroneren Stil zu behandeln. Zum Beispiel:

```
async function fetchData() {
  try {
    const response = await axios.get('https://jsonplaceholder.typicode.com/posts');
    console.log('Response:', response.data);
  } catch (error) {
    console.error('Error:', error);
  }
}
```

`fetchData();`

Dieses `async/await`-Beispiel erzielt das gleiche Ergebnis wie das vorherige Beispiel, verwendet jedoch eine synchroner aussehende Code-Struktur.



# Skills Network

## Änderungsprotokoll

Datum	Version	Geändert von	Änderungsbeschreibung
2023-12-11	0.1	Pooja Bhardwaj	Erste Version erstellt

© IBM Corporation 2023. Alle Rechte vorbehalten.