

JavaScript-Debugging mit VS Code und Edge

Effizientes Debugging von Code ist entscheidend, um Fehler zu finden und zu korrigieren. Visual Studio Code bietet eine nahtlose Debugging-Erfahrung für JavaScript, die sich mit Browsern wie Microsoft Edge erweitern lässt. Hier erfährst du, wie du JavaScript-Code mit VS Code und Edge debuggst und was es bei der Konfiguration der `launch.json`-Datei zu beachten gilt. Auch Troubleshooting-Tipps sind enthalten.

Setup fürs Debugging

Folgende Schritte sind nötig, um das Debugging vorzubereiten:

1. Installiere Visual Studio Code.
2. Installiere Microsoft Edge.
3. Füge die Erweiterung „Debugger for Microsoft Edge“ in VS Code hinzu.

Aufbau der `launch.json`

Die `launch.json` definiert Debug-Einstellungen und wird im JSON-Format verfasst. Neue `launch.json`-Dateien legst du über das Zahnrad-Symbol im Debug-Bereich von VS Code an. Wähle „Edge“ aus, um eine entsprechende Vorlage zu erhalten.

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "pwa-msedge",
      "request": "launch",
      "name": "Launch Edge against localhost",
      "url": "http://localhost:5500",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```

- **type**: Legt den Debugger-Typ fest, hier `pwa-msedge` für Edge.
- **request**: `launch` startet eine neue Browser-Instanz, `attach` dockt an eine laufende Instanz an.
- **name**: Individuell wählbarer Name für die Konfiguration.
- **url**: Die URL, die im Browser geöffnet wird. (der Port muss dem des `Live Servers` Ports entsprechen)
- **webRoot**: Pfad zum Projektverzeichnis, `${workspaceFolder}` verweist auf das geöffnete Projekt in VS Code.

Debugging-Workflow

1. Breakpoints setzen, indem du neben die Zeilennummern im Editor klickst.
2. Debug-Sitzung starten mit der gewählten Konfiguration und dem grünen Pfeil oder F5.

3. Im Edge-Browser mit der Anwendung interagieren. Stoppt der Code an einem Breakpoint, kannst du in VS Code Variablen prüfen und den Ablauf steuern.

Probleme und Lösungen

Debugger startet nicht

- URL in `launch.json` prüfen, muss mit dem laufenden Server übereinstimmen.
- Erweiterung „Debugger for Microsoft Edge“ sollte aktiv sein.
- Port-Konflikte vermeiden, falls der in `url` angegebene Port schon belegt ist.

Breakpoints werden nicht erkannt

- Pfad in `webRoot` muss zum Verzeichnis der Dateien passen.
- Sourcemaps sollten korrekt generiert sein, falls ein Compiler oder Bundler im Spiel ist.
- Vermeide minimierte JavaScript-Dateien, die das Zuordnen von Breakpoints erschweren können.