JS WebClient Entwicklungsumgebung

- Vorraussetzungen
- Installation der benötigten Libraries
  - Konfiguration des Nuclos-Servers
  - Starten des HTTP-Servers für den Webclient
  - Debuggen mit Chrome:

Vorraussetzungen

- Node.js
- Nuclos MUSS im Entwicklungsmodus gestartet werden

Installation der benötigten Libraries

Folgender Befehl muss nicht bei jedem Start, sondern nur initial und nach Änderungen der verwendeten Bibliotheken (Datei: `package.json`) durchgeführt werden.

```
cd nuclos/nuclos-webclient
npm install
```

df
npm install --save-dev eslint --verbose

Konfiguration des Nuclos-Servers

Die Server-Konfiguration wird mittels:

```
npm run config
```

erstellt.

Der Hostname, Port und Webcontextname des verwendeten Nuclos-Server muss anschließend über die Datei `nuclos/nuclos-webclient/src/assets/config.json` angepasst werden.

Starten des HTTP-Servers für den Webclient

```
node ./node_modules/@angular/cli/bin/ng serve --aot
```

Der Webclient ist dann über http://localhost:4200 erreichbar.

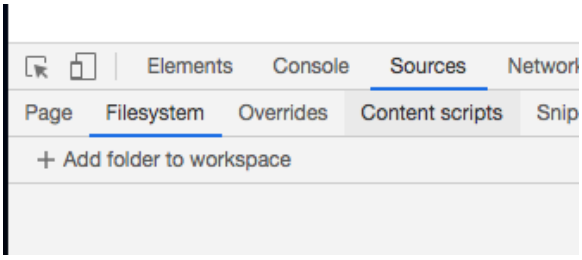
Mögliche Probleme beim Starten mit einem Addon

Es kann sein, dass die oben genannten Schritte in Verbindung mit einem Webaddon nicht funktionieren. Sollte dies der Fall sein, gibt es folgende Möglichkeiten um das Problem zu lösen:

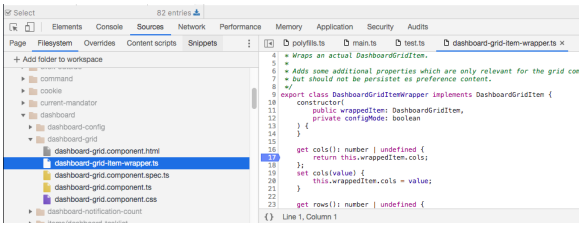
1. Löschen des Ordners **node_modules** und sicherstellen, dass keine anderen(unbekannten) Dateien des Webclients geändert wurden (ggf. mit einem VCS prüfen)
2. **npm install @nuclos/nuclos-addon-api**
3. **npm run install-addon-deps**
4. **npm install --save-exact @nuclos/nuclos-addon-api@latest rxjs@5.4.3 @nuclos/nuclos-addon-api@0.0.7**
5. **npm install**

Debuggen mit Chrome:

Um direkt in Chrom zu debuggen, muss man mit "Insprect" im Tab “Sources”, Untertab “Filesystem”, die TS-Sourcen (nuclos-webclient/src) mit “Add folder to workspace” angeben.



Der Chrome Debugger ist ausgezeichnet, sehr schnell und zuverlässig, weitaus besser als der IntelliJ Debugger. Einziger Nachteil: Zum persistenten Ändern der Sourcen sollte man die Stelle in IntelliJ suchen und dort Änderung. Zum schnellen Debuggen ist der Chrome Debugger zu empfehlen.
Mit Ctrl+P (Command+P beim Mac) kann man auch einzelne TS-Files öffnen, Breakpoints setzen.



Beispiel für Breakpoint in Zeile 17 von dashboard-grit-ite-wrapper.ts: