

# Inhaltsverzeichnis

- Allgemein** ..... 1
- Was tut das Plugin?** ..... 1
- Funktionen** ..... 2
- Unterstützte Spezifikationen** ..... 2
- Zweck & Anwendung** ..... 2
  - Script Interface Plugin Development Guide ..... 3
- Installation** ..... 3
- Konfiguration** ..... 3
- Fehlerdiagnose** ..... 3
- Entities** ..... 3
- Ordner & Dateien** ..... 4
- Versionsinformation** ..... 4
  - Dieses Document ..... 4
  - Plugin ..... 4
  - Assembly ..... 4



# Script Interface Plugin

## Allgemein

Das Script Interface Plugin ermöglicht es Ihnen, **Scripts** in der Sprache *JavaScript* zu schreiben, um CoDaBix zu programmieren. Scripts sind **leichtgewichtige Erweiterungen** von CoDaBix.

- Ein schneller und einfacher (aber dennoch mächtiger) Weg, die Funktionalität von CoDaBix zu erweitern
- implementiert in **JavaScript**, einer leistungsfähigen Scriptsprache, und **TypeScript**, um Fehler durch Bereitstellung von statischer Typisierung zu vermeiden und eine reichhaltige Entwicklererfahrung zu bieten
- kann verändert und neu gestartet werden, während CoDaBix läuft
- gespeichert in der CoDaBix Backend-Datenbank - somit ist es an die Daten gebunden, nicht an die Installation
- läuft in einer isolierten Umgebung, hat nur Zugriff auf bestimmte definierte CoDaBix-APIs, nicht Betriebssystem-APIs
- Sie müssen keine Entwicklungsumgebung installieren, Projekte anlegen, kompilieren, DLLs kopieren, usw. ...  
Stattdessen schreiben Sie einfach ein Script im eingebauten, webbasierten Scripteditor mit **IntelliSense**-Unterstützung

## Was tut das Plugin?

- Suchen, Erstellen und Manipulieren von Nodes und Lesen und Schreiben von Nodewerten
- Abonnieren von Events, z.B. wenn ein Nodewert geschrieben wurde
- Berechnungen ausführen, mathematische Funktionen aufrufen, Zufallszahlen generieren
- Einen Intervalltimer erstellen, welcher regelmäßig eine Scriptfunktion aufruft
- Dateien lesen und schreiben
- Eingehende HTTP(S)-Requests des CoDaBix Webservers verarbeiten (inkl. WebSocket-Verbindungen)
- HTTP(S)-Anfragen zu externen Servern ausführen

Wenn Sie bereits grundlegende Programmiererfahrung haben, werden Sie sich beim Schreiben von Scripts schnell zurecht finden. JavaScript ist eine der beliebtesten Scriptsprachen und in Kombination mit TypeScript (das statische Typsicherheit bietet) kann ein Script **von einem Einzeiler** (z.B. Anpassung eines Nodewerts, bevor er geschrieben wird) **bis hin zu komplexem Code** mit Namespaces, Klassen, Abhängigkeitsbeziehungen zu anderen Scripts und vielem mehr **skalieren**.

**Hinweis:** Sie müssen keine TypeScript-Kenntnisse besitzen, um Scripts schreiben zu können.

## Funktionen

- Hohe Performance durch die Kompilierung von Scriptcode in CIL-Bytecode
- Typsicherheit von Variablen, Eigenschaften und mehr durch TypeScript
- Geschützt gegen versehentliche Endlosschleifen durch das Anwenden eines Timeouts auf die Scriptausführung

## Unterstützte Spezifikationen

- Unterstützung von **TypeScript 4.1** im Scripteditor
- Vollständige Unterstützung für **ECMAScript 5.1** (Syntax und Bibliothek)
- Nahezu vollständige Unterstützung der **ECMAScript 2015+**-Syntax (let / const, class, for-of, async / await etc.) über Downlevel-Kompilierung zu ECMAScript 5.1 durch TypeScript
- Teilweise Unterstützung der **ECMAScript 2015**-Bibliothek (Collections, Typed Arrays, Promise)
- Unterstützt **Async Functions** für langanhaltende Operationen über die async- / await-Schlüsselwörter (erweitert das ereignisgesteuerte, blockierungsfreie JavaScript-Modell)

## Zweck & Anwendung

Sie können Scripts benutzen für

- einfache Schaltungen („Wenn ein Knopf gedrückt wird, soll das Licht angeschalten werden und nach 3 Minuten wieder ausgehen“)
- Nachbearbeitung von Werten, die von einem Gerät gelesen oder auf ein Gerät geschrieben werden, z.B. durch das Anstellen von Berechnungen
- die Generierung von komplexen Nodehierarchien / -strukturen
- die Generierung von Demodaten (z.B. alle 2 Sekunden einen zufälligen Wert auf einen Node schreiben)
- Big Data: Sammeln von externen Daten um einen bestimmten Punkt herum, z.B. um das aktuelle Wetter von Berlin in einem Datenpunktnode zu speichern
- das Bereitstellen einer komplexen, benutzerdefinierten Bedingung für Trigger

## Script Interface Plugin Development Guide

Hier finden Sie den Development Guide für das Script Interface Plugin:

- [Script Interface Plugin Development Guide](#)

## Installation

Das Script Interface Plugin ist ein fester Bestandteil von CoDaBix und muss nicht installiert werden.

## Konfiguration

Das Script Interface Plugin ermöglicht die Konfiguration von Scripts in der CoDaBix Webkonfiguration über den Menüpunkt „Script Interface“ - siehe [Scripts verwalten](#) im Script Interface Plugin Development Guide.

## Fehlerdiagnose

Das Script Interface Plugin ermöglicht die Diagnose von Scripts in der CoDaBix Webkonfiguration über den Menüpunkt „Script Interface“ - siehe [Scripts verwalten](#) im Script Interface Plugin Development Guide.

## Entities

Das Script Interface Plugin verwendet das CoDaBix Entity Model nicht, da es das Verwalten von Scripts über die CoDaBix Webkonfiguration ermöglicht, und stellt deshalb keine Entities zur Verfügung.

## Ordner & Dateien

Das Script Interface Plugin ist ein fester Bestandteil von CoDaBix und hat deshalb keine zusätzlichen Dateien oder Ordner.

## Versionsinformation

### Dieses Document

|                |                |
|----------------|----------------|
| <b>Datum</b>   | 2017 - 11 - 14 |
| <b>Version</b> | 1.8            |

### Plugin

|                |                                |
|----------------|--------------------------------|
| <b>Name</b>    | Script Interface Plugin        |
| <b>Version</b> | Entspricht der CoDaBix-Version |

### Assembly

Das Script Interface Plugin ist ein fester Bestandteil von CoDaBix und hat deshalb keine separate Assembly.

From:

<https://www.codabix.de/> - **CoDaBix®**

Permanent link:

<https://www.codabix.de/de/plugins/interface/scriptinterfaceplugin>

Last update: **2022/01/31 20:19**