DOWNLOADS

Microservices und Skripting

ClearScript stattet das eigene Projekt mit Skripting-Fähigkeit aus.

Als Schwerpunkt haben wir diesmal das Thema Microservices ausgesucht. Das sind die kleinen Codeeinheiten, aus denen eine verteilte Applikation entsteht. Mit der monolithischen Architektur waren alle unzufrieden. Also musste etwas Modulares her.

Apropos unzufrieden: Kaum verwenden Anwender eine Software, fallen ihnen Hunderte Erweiterungen dafür ein. Haben die Entwickler dann alle gewünschten Features eingebaut, kommen neue Wünsche auf: Fehlerfrei soll die Software dann doch bitte auch noch sein. Und selbst wenn das erfüllt ist, wollen die Anwender die Software automatisieren.

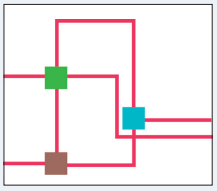
Um das zu realisieren, gibt es die eine oder andere Komponente. ClearScript ist so eine, die sich einfach per NuGet in die Anwendung einbauen lässt. Als Skriptsprache versteht sich ClearScript auf JavaScript und VBScript.

Wie einfach das funktioniert, zeigt der folgende Quellcode:

```
using System;
using Microsoft.ClearScript.V8;

namespace ClearScript
{
 class Program
 {
  static void Main(string[] args)
  {
```

Schwerpunkt Microservices



Der Monolith ist schon lange verpönt. Er gilt als unwartbar und auch die Entwicklung lässt sich nicht von mehreren Teams gleichzeitig vollführen. Die Aufteilung in Komponenten hat sich bewährt und wird mit Microservices auf die nächste Stufe gehoben: Die Komponenten werden gehostet und sind noch mehr voneinander entkoppelt. Zur Umsetzung von Microservices gibt es für .NET viele Bibliotheken.

Konfiguration per Dapplo.Config

INI-Dateien sind schon seit den ersten Windows-Versionen im Einsatz. Sie bestehen aus Abteilungen und Schlüssel-Wert-Paaren. Wer diese Dateien lesen will, kann dafür auf Dapplo.Config setzen. Die Bibliothek liest und schreibt INI-Dateien.

```
  using (var engine = new V8ScriptEngine()) {
   engine.AddHostType("Console", typeof(Console));

   engine.Execute("Console.WriteLine(
    '{0} ist nicht die Antwort auf alle Fragen!',
    Math.PI)");

   engine.Execute("person =
    { name: 'dotnetpro', type: 'magazine' }");
   Console.WriteLine(engine.Script.person.name);

   engine.Execute("var z = Array(3.14, 42, 7, 11);");
   engine.Execute("Console.WriteLine(z.join('#'))");
   engine.Execute("console.log(z.join('#'))");

   engine.Execute("function printout(name){
    return name + ' ist toll'; }");
   engine.Execute("Console.WriteLine(
    printout('dotnetpro'))");
  }
 }
}
```

Die Ausgabe lautet:

```
3,14159265358979 ist nicht die Antwort auf alle Fragen!
dotnetpro
3.14#42#7#11
dotnetpro ist toll
```

■

Schwerpunkt

UserDataMicroservice
Ein User-Data-Microservice mit .NET Core implementiert für eine soziale, plattformübergreifende Applikation. User-Daten werden in einer Postgresql-Amazon-Web-Services-RDS-Datenbank gespeichert. Der Microservice wird in einem Docker Container ausgeführt.

Microservice-Demo
Test Projekt und Demo zum Erstellen einer Microservice-Infrastruktur mit .NET-Core-Tools.

ConferenceCloud – .NET Core Microservices auf Azure
Beispielprojekt und Referenz zum Erstellen von Microservice-basierten Applikationen mit .NET Core unter Verwendung von Docker, Kubernetes und Azure.

Microservice
Xigadee ist ein Open-Source-Microservice. Die Bibliotheken bieten einen einfachen und durchgängigen Ansatz, um ein modernes Enterprise-API und Microservice-basierte Lösungen zu bauen.

Augmented-ASP.NET-Backend
Ein ASP.NET-API-Microservice unter Verwendung von Code First Entity Framework.

Microservices Using ASP.NET Core
Bauen Sie Microservices mithilfe von ASP.NET Core auf.

mockroservices-dotnet
Toolkit zum Erstellen von Microservices in einer Multi-Projekt-Testumgebung unter Verwendung von Mock-Distributed Mechanismen.

Cosella
Das Cosella Microservice Framework erleichtert das Entwickeln von .NET-Service-basierten Lösungen. Ein neues Projekt.

Simple-Transaction
Simple-Transaction ist eine .NET-Core-Musteranwendung und ein Beispiel dafür, wie man ein Microservice-basiertes Backend-System für ein einfaches automatisiertes Banking-Feature implementiert.

XComponent Resource Center
XComponent ist eine Plattform zum Erstellen, Überwachen und Teilen von Microservices. In XComponent ist ein Microservice eine Sammlung aus Komponenten. Jede Komponente basiert auf Zustandsautomaten, die grafisch entwickelt sind.

LoremIpsum-Microservice-WebApp
Diese dynamisch skalierbare Webapplikation basiert auf Microservices, Docker, ASP.Net, Angular u.v.m.

Aktuelle Downloads

Piranha.Core
Piranha.Core ist die aktuelle Version von Piranha CMS, komplett neu geschrieben für .NET Standard und ASP.NET Core.

HandlerInvoker
HandlerInvoker ist ein experimentelles Projekt zum Implementieren einer Instanz wie ASP.NET Core, um Aktionen von einem Handler aufzurufen.

ASP.NET Core & VueJS Starter Kit
Ein entkoppeltes, einfaches Starter-Kit-Template für ASP.NET Core 2.2 und VueJS 2.6. Es enthält ein angepasstes Eigenschaftsvalidierungselement auf der Serverseite sowie entkoppelte Statusverwaltung für Vuex Webpack.

Aphid
Aphid ist eine einzigartige Multiparadigma-Sprache, die interpretiert oder kompiliert werden kann. Es ist sowohl eine eigenständige .NET-Sprache, kann aber auch in andere Sprachen wie Python, PHP und Verilog kompiliert werden.

Network-Tester
Erstellen eines kleinen Netzwerktools unter Verwendung von C#. Diesem Tool werden noch Features hinzugefügt.

ASP.NET Core Tooling
Tools für ASP.NET Core Apps, wie z. B. MSBuild Targets, Visual Studio Erweiterungen und Befehlszeilentools.

.NET Extensions
NET Extensions ist eine plattformübergreifende Sammlung von APIs für häufig genutzte Programmiermuster und Utilities, wie z. B. Dependency Injection, Logging sowie App-Konfiguration.

TypeKitchen
TypeKitchen ist eine kleine Library für schnelle Meta-Programmierung in .NET.

Getit
Getit ist ein einfaches Paket zum Erstellen von GraphQL-Queries. Es erlaubt auch RAW-Queries, falls Sie den Query-Builder nicht verwenden wollen. Aktuell erstellt Getit nur Queries und hilft nicht bei Mutationen.

Solid Instruments
Solid Instruments ist eine fortgeschrittene Mehrzweck-.NET-API-Suite. Solid Instruments hilft Ihnen beim raschen Lösen von Designanforderungen und liefert stabile, sichere, hoch leistungsfähige Software.

WebDAVServerSamples
WebDAV, CalDAV & CardDAV Serverbeispiele in C# und VB, basierend auf IT Hit WebDAV Server Engine für .NET.