

Softwareentwicklung mit FlowProtocol 2

Wolfgang Maier

15. November 2024



Inhaltsverzeichnis

1 Bezug und Konfiguration

5

Vorwort

Die vorliegende Anleitung beschreibt Möglichkeiten, die Anwendung *FlowProtocol 2* in der professionellen Softwareentwicklung einzusetzen und damit die besonderen Herausforderungen zu meistern, die sich in diesem Bereich stellen. An vielen kleinen und auch auch größeren Beispielen wird gezeigt, wie man Skripte in *FlowProtocol 2* erstellt und welche Bandbreite an Hilfsmitteln sich damit bereitstellen lassen. Man lernt, Skripte als Ablageform für Wissen und Konventionen innerhalb eines Entwicklungsteams zu sehen und die eigenen Fähigkeiten in dieser Form weiterzugeben.

FlowProtocol 2 ist die komplett überarbeitete Nachfolgeversion von *FlowProtocol*, das im Winter 2021/22 entwickelt wurde. Es handelt sich um eine kleine Anwendung, die über einen Browser bedient wird, und auf der Skripte ausgeführt werden können. Die Skripte bestehen aus einfachen Textdateien, die in einem beliebigen Editor erstellt werden können. Die Verwaltung der Skripte erfolgt in einer lokalen Verzeichnisstruktur, mit der eine organisatorische oder aufgabenbezogene Gliederung abgebildet werden kann.

Bei der Ausführung eines Skriptes werden Informationen über Eingabefelder abgefragt und über die Anweisungen im Skript verarbeitet. Die Ausgabe erfolgt ausschließlich als Ergebnisdokument im Browser, aus dem dann z.B. Textpassagen über die Zwischenablage weiterverwendet werden können. Alle eingegebenen Daten werden ausschließlich als Parameter in der URL verwaltet, es gibt keine angebundene Datenbank, keine Benutzerverwaltung und es werden durch die Anwendung keine Dateien erstellt und geändert.

Der Anwendungsfall, für den *FlowProtocol* ursprünglich entwickelt wurde, ist die Erstellung von Checklisten, die durch Interaktion mit dem Benutzer auf einen individuellen Fall zugeschnitten werden, und so beliebig ins Detail gehen können, ohne unnötige Einträge aufzulisten. Daran werden sich auch die ersten Beispiele in dieser Anleitung orientieren. Auf dieser Basis entstanden zahlreiche Skripte für den Product Owner, die für die verschiedenen Standardentwicklungen alle benötigten Einstellungen und Informationen abfragten, die durch das Designsystem und die damit assoziierten Framework-Komponenten verfügbar waren. Der Vorteil für den Product Owner bestand darin, dass ihm aufgrund der in den Skripten hinterlegten Abhängigkeiten immer nur die Optionen angeboten wurden, die für den jeweiligen Fall sinnvoll waren, und er so auch an alle Entscheidungen herangeführt wurde, die an der jeweiligen Stelle getroffen werden mussten. Das Ergebnisdokument bestand in diesem Fall aus einem sehr umfangreichen Userstory-Entwurf, in dem schon alle abgefragten Informationen und Entscheidungen eingearbeitet waren, und der mehr oder weniger nur noch um einige Benennungen und Aufzählungen angereichert werden musste. Schon bald wurde *FlowProtocol* um die Möglichkeit erweitert, auch Texteingaben abzufragen und diese in das Ergebnisdokument einzuarbeiten.

Die Systematik, die auf Seiten der Story-Formulierungen möglich ist, ist hauptsächlich begründet durch die Komponenten des über viele Jahre aufgebauten eigenen Frameworks und deren Möglichkeiten und die Erfahrung des Teams, also das über zahlreiche Wiederholungen aufgebaute Wissen, wie man wiederkehrende Muster umsetzt, und was dabei zu beachten ist. Entsprechend lag es nahe, ähnliche Unterstützungswerkzeuge auch auf Entwicklungsseite zu schaffen, die den Programmierer dazu anleiten, die richtigen Klassen zu verwenden und am Ende idealerweise sogar fertigen Programmcode erstellen. Zu diesem Zweck wurde die Formatierungsform als Code geschaffen, bei der man einen Block per Schaltfläche in die Zwischenablage zu kopieren kann, und der CamelCase-Befehl, mit dem Namen für Felder, Funktionen oder Variablen erzeugen kann. Die Verwendung

interaktiver Anleitungen mit Code-Generierung steigert nicht nur die Effizienz bei der Umsetzung von Standardaufgaben, sie hilft auch bei der Sicherstellung von Einheitlichkeit und ist damit eine wichtige Säule in der Qualitätssicherung.

Auch wenn *FlowProtocol 2* selbst nicht direkt mit anderen Anwendungen interagiert, so kann in einem Skript praktisch jede URL mit beliebigen Parametern aufgebaut werden, so dass sowohl bei der Ausführung, als auch aus dem Ergebnisdokument der parametrisierte Aufruf anderer Web-, oder Intranet-Anwendungen möglich ist. Allen voran können Skripte auf diese Weise andere Skripte aufrufen, aber auch viele der im Entwicklungsumfeld eingesetzten Anwendungen für die Verwaltung von Vorgängen oder als Wiki bieten gute Steuerungsmöglichkeiten. Schon allein mit der Übergabe eines Suchbegriffs kommt man schon recht weit, und über einen MailTo-Link lassen sich sogar vollständig ausformulierte E-Mails vorbereiten. Die Codierung der Parameter für die URL lässt sich dabei sehr einfach mit dem `UrlEncode`-Befehl umsetzen.

Inzwischen umfasst der Befehlssatz von *FlowProtocol 2* alle notwendigen Befehle für den Aufbau von Programmen, wie If-Abfragen, For-Schleifen, sowie die Definition von Funktionen, die auch rekursiv aufgerufen werden können. Zusammen mit den verschiedenen Befehlen für das Rechnen mit Zahlen und Datum-Uhrzeit-Werten und zur Manipulation von Zeichenketten lassen sich mit sehr geringem Aufwand kleinere und größere nützliche Hilfsanwendungen schreiben, die unmittelbar auf jedem Arbeitsplatz verfügbar sind.

Der Nutzen von *FlowProtocol 2* innerhalb eines Teams oder einer Einrichtung erhöht sich in besonderem Maße dadurch, dass die Skripte von einer größeren Zahl an Kollegen gepflegt und erweitert werden, und dass der bestehende Erfahrungsschatz auf diese Weise permanent durch neues Wissen erweitert wird. In einfacher Form lässt sich dies sogar hinbekommen, ohne dass dafür Programmierkenntnisse vorausgesetzt werden, indem ein Skript die für seine eigene Erweiterung notwendigen Informationen abfragt, und daraus den resultierenden Programmcode samt Einbauanleitung selbst erzeugt. Auch diese Methode wird an einem Beispiel beschrieben.

Bis dahin werden aber erst einmal die grundlegenden Befehle und Funktionsweisen beschrieben, um einen einfachen Einstieg in *FlowProtocol 2* zu ermöglichen. Ich freue mich über jeden, der mit Hilfe dieser kleinen Anleitung auf Entdeckungstour geht, und wünsche viel Spaß und einen hoffentlich nutzbringenden Einsatz.

Wolfgang Maier

1 Bezug und Konfiguration

FlowProtocol 2 steht unter der MIT-Lizenz und ist unter folgender Adresse auf GitHub verfügbar :

`https://github.com/maier-san/FlowProtocol2`

Die Programmcode kann direkt mit Git oder der frei verfügbaren Entwicklungsumgebung Visual Studio Code in ein lokales Verzeichnis, z.B. `D:\Anwendungen\FlowProtocol2`, übertragen, und dort mit dem `dotnet`-Befehl kompiliert werden:

```
dotnet.exe build
```

```
D:\Anwendungen\FlowProtocol2\FlowProtocol2\FlowProtocol2.csproj
```

Die Konfiguration erfolgt über die Datei `appsettings.json`, in der hauptsächlich der Pfad auf das Skripte-Verzeichnis mit dem Parameter `ScriptPath` eingestellt werden muss. Dieses Verzeichnis enthält die Skripte und Unterverzeichnisse, die von *FlowProtocol 2* auf der Startseite angezeigt werden, und ist damit der Dreh- und Angelpunkt der Skriptverwaltung. Für die ersten Versuche kann man den Parameter auf den `Scripts`-Ordner innerhalb des Projektes setzen, etwa so:

```
"ScriptPath": "D:\\Anwendungen\\FlowProtocol2\\Scripts",
```

In einer Firma oder Einrichtung wird man das Verzeichnis so wählen, dass die Mitarbeiter, die aktiv an den Skripten arbeiten, dort direkt Dateien editieren und erstellen können, z.B. indem man dieses auf einem Netzlaufwerk verfügbar macht. Zusätzlich wird man die Skripte regelmäßig sichern, idealerweise mit Hilfe einer Versionsverwaltung. Wenn man Manipulation befürchtet, kann man das Editieren auch vollständig auf den Weg über die Versionsverwaltung beschränken, verbaut sich damit aber die Möglichkeit, die Wirkung von Änderungen an einem Skript unmittelbar nach dem Speichern durch die Aktualisierung des Browser-Tabs zu überprüfen. In diesem Fall wäre dann eine getrennte Skript-Entwicklungsumgebung sinnvoll, analog zu den sonstigen Entwicklungsumgebungen. *FlowProtocol 2* selbst benötigt auf dieses Verzeichnis nur Leserechte.

Wie schon im Vorwort beschrieben ist die Möglichkeit, Links zu generieren, ein mächtiges Mittel um die Interaktion mit anderen Anwendungen zu ermöglichen. Wie man jedoch aus jeder IT-Sicherheitsbelehrung weiß, kann das Anklicken von Links auch Gefahren mit sich bringen, und speziell in Phishing-E-Mails nutzen Angreifer vertrauenswürdig aussehende Links, um den Empfänger auf eine nachgebaute oder mit Schadcode gespickte Seite zu leiten. Auch mit den Links in einem Skript sind solche Angriffe möglich, wenn auch aufgrund der Beschränkung auf die eigenen Mitarbeiter eher unwahrscheinlich. Aus diesem Grund werden alle durch ein Skript ausgegebenen Links, deren Domäne nicht in der Auflistung des `LinkWhitelist`-Parameters steht, zusätzlich zu dem Anzeigetext mit der vollständigen URL ausgegeben. Damit kann bei der Konfiguration entschieden werden, welche Seitenaufrufe so vertrauenswürdig sind, dass sie auch nur mit dem Anzeigetext dargestellt werden können, was zum einen die Lesbarkeit erhöht und zum anderen die Aufmerksamkeit des Anwenders auf die Links konzentriert, die nicht diese Einstufung haben.

Für die regelmäßige lokale Bereitstellung von *FlowProtocol 2* ist es am einfachsten, vom Basisrepository auf GitHub mittels Fork zu verzweigen und die lokalen Anpassungen im eigenen Zweig zu verwalten.