

## INTERESSANTE OPEN-SOURCE-Projekte: DISCOVER .NET

# Nette Bekanntschaft

Mit Discover .NET findet man schnell interessante .NET-Projekte.

Im .NET-Umfeld hat sich in den letzten Jahren eine gesunde Open-Source-Community entwickelt, die immer wieder interessante und sinnvolle Projekte liefert. Leider finden diese aufgrund der deutlich größeren und besser vermarkteten Open-Source-Lösungen von Microsoft häufig zu wenig Beachtung.

Aus diesem Grund wirft dotnetpro immer wieder einen genaueren Blick auf solche Projekte, diesmal auf das Projekt Discover .NET [1]. Dieses ist allerdings keine Open-Source-Software, sondern eine Webseite voller interessanter Inhalte rund um das Thema .NET. Dabei geht es oft auch um Inhalte, die nicht auf den großen Kanälen wie etwa Microsofts eigenem Blog aufzufinden sind. Die Site wird zudem von einem anderen großartigen und populären .NET-Projekt generiert: Statiq.

## Worum geht's?

Wie findet man am schnellsten neue, sinnvolle .NET-Projekte? Wo werden Probleme und Fragen von .NET-Open-Source-Projekten für eine schnelle Ansicht aggregiert? Gibt es eine Website, um gute Blogs rund um .NET zu finden und öffentlich zu machen? Welche Podcasts und Broadcasts zu aktuellen Entwicklungen im .NET-Ökosystem gibt es? Vor Discover .NET gab es auf solche Fragen keine gemeinsame und konsistente Antwort.

Mit Discover .NET gibt es nun einen zentralen Anlaufpunkt, der diese Fragen beantwortet [2].

Discover .NET ist in fünf Kategorien eingeteilt:

- Projekte
- Beiträge
- Blogs
- Broadcasts
- Ressourcen

Ein zentraler Punkt ist die Suche. Die Suchfunktion wird hierbei von Algolia hergestellt. Dieser Anbieter stellt Suche als Software as a Service bereit und wird oft für seine sehr guten Ergebnisse mit ausgezeichneter Reaktionszeit geschätzt. Die Einbindung von JavaScript erlaubt auch die Suche auf der Site, ohne den aktuellen Inhalt zu verlassen.

Die Einstiegsseite stellt die neuesten Blogposts, Podcasts und andere interessante Inhalte zusammen. Daneben gibt es noch die wichtigsten Links, um schnell auf andere Inhalte wie beispielsweise lohnenswerte .NET-Projekte zu gelangen.

Die wichtigste Funktion der Website ist jedoch die Möglichkeit, neue Inhalte vorzuschlagen.

## Neue Inhalte einstellen

Hinter Discover .NET liegt keine Datenbank. Tatsächlich handelt es sich bei diesem Projekt um statische Webseiten, die aus dem Quellcode auf GitHub beim Build mithilfe des .NET-Projekts Statiq erstellt und hochgeladen werden. Neue Inhalte werden daher ausschließlich über Pull Requests veröffentlicht.

Neben einer Beschreibung des Inhalts sollte der Pull Request mindestens eine neue Definitionsdatei liefern. Diese Definitionsdateien werden im YAML-Format eingestellt und liegen – abhängig von der Art des Inhalts – in einem Unterverzeichnis von *data*. Als Beispiel werden Projekte in *data/projects* abgelegt.

Listing 1 zeigt die Definition eines .NET-Projekts. Neben projektspezifischen Metadaten (zum Beispiel *Language*, *SourceCode*, *Website*) gibt es hier noch weitere organisatorische Felder wie etwa *DiscoveryDate*, *Comment*.

Wer neue Inhalte lieber grafisch vorschlägt, kann auch das Formular auf der Seite benutzen (Bild 1). Es zeigt alle Felder der zugehörigen YAML-Vorlage an und füllt sie beim Abschieken für den anschließenden Pull Request aus. Die dabei beschriebenen Hilfestellungen – zum Beispiel welche Felder über das GitHub-API automatisch ausgefüllt werden können – sind sinnvoll und gehen sicherlich über die sofort ersichtliche Struktur der YAML-Dateien hinaus.

Alle Einträge bei Discover .NET lassen sich auch ohne die Website verwenden. Hierfür stellt das Website-Projekt statische JSON-Dateien zur Verfügung, die als „API“ auf der Webseite aufgeführt sind [3]. Listing 2 stellt einen Teil der ►

### • Listing 1: Definition eines .NET-Projekts

```
Image: https://raw.githubusercontent.com/AngleSharp/AngleSharp/master/header.png
Language: C#
NuGet:
  - AngleSharp
  - AngleSharp.Scripting.JavaScript
SourceCode: https://github.com/AngleSharp/AngleSharp
Tags:
  - HTML
Website: https://anglesharp.github.io/
DiscoveryDate: 6/3/2018
Comment: If you're using HTML Agility Pack (which is great), might be worth looking at AngleSharp too.
```

## ● Interview mit Dave Glick

### „Arbeite mit dem, was du hast“

Dave Glick ist der Autor des Frameworks Discover .NET. Bei GitHub gehören ihm unter anderem noch Statiq, Buildalyzer, Netlify Sharp und AzurePipelines.TestLogger.

*Dave, was ist dein Hintergrund?*

**Dave Glick:** Ich entwickle mittlerweile seit fast zwei Jahrzehnten professionell Software. Ich habe während meiner Schulzeit damit begonnen, Webseiten für Kunden eines lokalen Internet-providers zu bauen. Obwohl ich mich aktuell auf C# und das .NET Framework spezialisiert habe, bin ich durch so einige Stacks inklusive Java, C++ und PHP gegangen, bis ich schließlich hier gelandet bin.

Jede dieser Sprachen und Plattformen bietet Teile, die mir gut gefallen, weshalb diese Erfahrungen maßgeblich die Art und Weise geprägt haben, wie ich heute über Code nachdenke. Wie die meisten Entwickler versuche ich, auch in JavaScript und beim Web auf dem aktuellen Stand zu bleiben.

Im Tagesgeschäft arbeite ich an einer Applikation im Bereich der Finanzregulierung. Meine wahre Leidenschaft ist jedoch die Arbeit an Open-Source-Projekten und Software, die in der Öffentlichkeit weiterentwickelt wird. Ich helfe bei einigen und warte einige solcher Projekte.

*Warum hast du Discover .NET erstellt?*

**Glick:** Ich höre häufig den Satz „.NET hat ein Open-Source-Problem“, und obwohl ich denke, dass das in gewisser Weise sicherlich stimmt – wie für jede Open-Source-Community –, sehe ich, dass die .NET-Community dazu neigt, „top-heavy“ zu sein. Damit meine ich, dass meistens nur eine kleine Anzahl von Entwicklern und Projekten die Aufmerksamkeit erhält.

Im Gegensatz dazu gibt es eine großartige Anzahl und Vielfalt an tollen .NET-Projekten, die leider fast niemand kennt. Ich vermute, dass dies den Eindruck eines kleineren und weniger gesunden Ökosystems vermittelt und gleichzeitig einige unterrepräsentierte Community-Mitglieder davon abhält, die verdiente Aufmerksamkeit zu erhalten. Discover .NET ist ein Versuch, solchen Projekten, Blogs und anderen Bestrebungen der Community diese Aufmerksamkeit zu geben.

*Für welche anderen Projekte leistest du noch Beiträge?*

**Glick:** Mein Hauptprojekt ist ein statischer Webseiten-Generator namens Statiq. Das meiste meiner haupt- und nebenberuflichen Tätigkeit hat mit Webentwicklung – zumindest im einen oder anderen Sinn – zu tun. Ich war schon immer fasziniert davon, warum wir Webseiten so bauen und rendern, wie wir das tun. Vor ungefähr fünf Jahren, als vorgerenderte Websites, die auf Client-Code und -Diensten basieren (zum Beispiel der Jamstack), wirklich Aufmerksamkeit erlangten, sah ich viele Parallelen zu Ideen, die ich in der Vergangenheit hatte. Das Erstellen komplexer datenorientierter Webanwendungen könnte so aussehen. Statiq ist mein Versuch, diese Vision durch den Aufbau eines statischen .NET-

Webseiten-Generators zu verwirklichen, der so leistungsstark wie ASP.NET als serverbasierte Site-Plattform ist.

Neben Discover .NET und Statiq verwalte ich auch mehrere Dienstprogramm-bibliotheken wie Buildalyzer, MsBuildPipeLogger und NetlifySharp und bin Mitglied des Kernteams des Cake-Build-Orchestration-Tools. Im Allgemeinen stelle ich mir meine Open-Source-Arbeit gerne als Portfolio vor. Damit meine ich, dass ich gerne versuche, alles in Beziehung zu halten, damit die Arbeit an einem dieser Projekte in die anderen einfließt.

Meine großen Schwerpunkte sind derzeit die Unterstützung von Entwicklern beim Erstellen, Dokumentieren und Bewerben ihrer eigenen Projekte. Alle meine Open-Source-Beiträge fallen in diese Kategorie.

*Was ist deine Vision für die Zukunft von Discover .NET?*

**Glick:** Discover .NET wird mit Statiq erstellt. Wenn der Statiq-Site-Generator ausgereift und verbessert wird, eröffnet sich für Discover .NET ein neues Potenzial. Was ich in naher Zukunft gerne tun würde, ist, einzelne Entwickler zusätzlich zu dem, was sie erstellen, zu katalogisieren und zu fördern (wie Projekte und Blogs, die bereits katalogisiert sind). Ich denke, es gibt eine große Chance, die Aufmerksamkeit auf die Menschen hinter der Community zu lenken und dabei zu helfen, die Konditionen für Open-Source-Entwicklung und deren Entwickler zu verbessern.

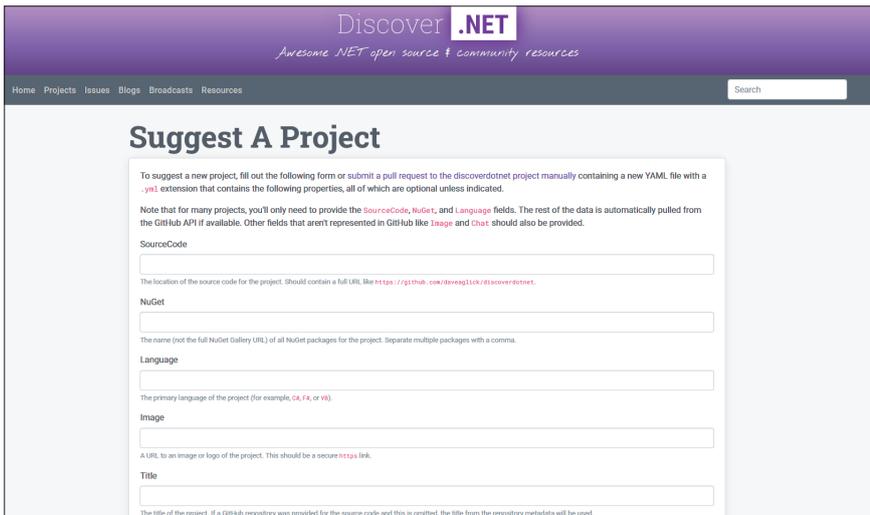
Ich möchte auch bessere Wege finden, um Neuentwicklungen zu fördern, insbesondere bei weniger bekannten Projekten und Entwicklern. Das Endziel war immer, ein zusammenhängendes und in hohem Maße auffindbares Web von .NET-Open-Source-Daten zu erstellen.

*Welche Features fehlen aus deiner Sicht noch in C# und .NET?*

**Glick:** Um ehrlich zu sein, wünsche ich mir schon länger nicht mehr viel von der Sprache oder der Plattform. Versteh mich nicht falsch – ich schätze und genieße es wirklich, einige der neueren Sprachfunktionen zu verwenden, aber meine Denkweise ist fast immer: „Arbeite mit dem, was du hast“. Infolgedessen bin ich nicht frustriert über fehlende Funktionen und habe auch keine Wunschliste.

Ich denke, wenn absolut notwendig, würde ich gerne sehen, dass das Konzept „Extension Everything“ umgesetzt wird. Dies liegt schon eine Weile auf dem Tisch und es wäre cool, in Code zum Beispiel Erweiterungseigenschaften einzusetzen.

Ich würde es auch sehr begrüßen, wenn weiterhin daran gearbeitet würde, die Plattform stabil zu machen und die Benutzerfreundlichkeit zu verbessern, insbesondere für Entwickler, die neu in .NET sind. Es ist in letzter Zeit viel besser geworden, aber es gab eine Zeit kurz nach der ersten .NET-Core-Version, in der die Werkzeuge wirklich grottig waren. Ich stoße gelegentlich immer noch auf schlecht dokumentiertes und merkwürdiges Verhalten, daher ist jede Anstrengung, diese Fälle weiter zu beseitigen, willkommen.



Vorschlagen eines Projekts über das Formular auf der Discover-.NET-Webseite (Bild 1)

Datei *project-counts.json* dar. Es gibt eine Vielzahl solcher Dateien, sodass für verschiedene Anfragen bereits die Antworten vorliegen.

Die Daten auf der Seite werden alle zwei Stunden erneuert, sodass neue Inhalte permanent akzeptiert oder aktualisiert werden. Nach dem Bauvorgang kommt es zu einem teilweisen Update der statischen Seiten. Bereitgestellt wird die Website über Netlify.

### Statiq

Statische Webseitengeneratoren wie Statiq gibt es viele: Hugo, Jekyll, Gatsby, um nur ein paar zu nennen. Was Statiq auszeichnet, sind zwei Eigenschaften:

- Es ist in .NET/C# geschrieben.
- Es verwendet C# als domänenspezifische Sprache, um den Build zu beschreiben.

Mit Statiq lassen sich beliebige Daten zu Dokumenten verarbeiten. Ob diese ursprünglich in Markdown, HTML oder beispielsweise JSON verfasst sind, spielt keine Rolle. So stellt Discover .NET die einzelnen Einträge als YAML-Dateien bereit. Die aus den Daten erzeugten Dokumente lassen sich ebenfalls in beliebigen Formaten ausgeben. Dadurch ist es

möglich, Discover .NET ein JSON-basiertes API zu geben – sozusagen ohne Mehraufwand.

Die Funktionsweise des Statiq-Generators ist in Bild 2 dargestellt. Man kann Statiq beibringen, wie die Abhängigkeiten zwischen den Quelldaten richtig erkannt werden können. Das Programm kann daraufhin automatisch eine Vielzahl von Quelldateien in das Seitenmodell, das erstellt werden soll, mit aufnehmen.

Wie sich an Discover .NET zeigt, können durchaus auch Hunderte oder Tausende von Eingabedateien schnell verarbeitet werden. Der leistungsfähige Unterbau mit .NET Core ist modular aufgebaut. Wie man es vom .NET Core SDK gewöhnt ist, lassen sich alle wichtigen Befehle auch direkt in der Kommandozeile ausführen.

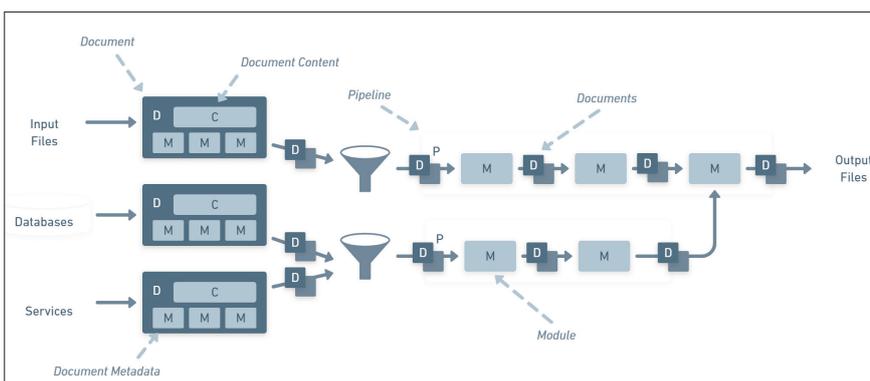
Dank desselben Autors besteht zwischen Discover .NET und Statiq eine enge Beziehung. Discover .NET kann man als einen primären Einsatz von Statiq betrachten, weshalb sich Discover .NET auch sehr gut als Einstiegsprojekt für Statiq nutzen lässt.

### Einen Beitrag leisten

Einer der wirklich wichtigen Punkte für ein solches Projekt ist ein ansprechender Guide, der erläutert, wie man an der Codebasis mitwirken kann. Discover .NET hat einen solchen auf der Webseite veröffentlicht [4]. Er geht allerdings nicht wirklich auf die Codebasis ein, sondern hauptsächlich auf allgemeine Prinzipien für Beiträge in Open-Source-Projekten. Am Discover-.NET-Projekt auf GitHub wirken laut GitHub fast 200 Teilnehmer mit.

Um wirklich komplett zu sein, sollte das Repository von Discover .NET auch über eine ausführliche Readme-Datei und einen Code of Conduct verfügen. Während die Readme-Datei durchaus – wenn im Inhalt auch äußerst sparsam – vorhanden ist, fehlt ein Code of Conduct. Letztlich lässt sich das aber auch als Vorteil ansehen. Der bürokratische Overhead sollte hier wohl möglichst gering gehalten werden. Die einfache Trunk-basierte Entwicklung funktioniert auch ohne besondere Zeremonie.

Als Ziel-Branch bei einem Pull Request empfiehlt sich, einfach mit *master* zu arbeiten. Idealerweise erstellt man vorher für nicht besprochene Änderungen ein GitHub-Issue, um den Beitrag anzukündigen und in aller Breite zu diskutieren. Auch Bug Reports, Diskussionen sowie Feature Requests sind willkommen. Für neue Inhaltsbeiträge muss vorab selbstverständlich kein Issue erstellt werden. Hier wird der Pull Request als Ankündigung und als Diskussionsgrundlage gesehen.



Die Funktionsweise des Webseitengenerators Statiq (Bild 2)

### Listing 2: Anzahl offener Issues für alle Projekte

```
[
  {
    "projectKey": "project-key",
    "issuesCount": 159,
    "recentIssuesCount": 5,
    "helpWantedIssuesCount": 78
  },
  {
    ...
  }
]
```

Die vorhandenen Issues sind an sich gut gepflegt und teilweise mit Tags versehen. Hier findet sich schnell ein Bereich, in dem man tätig werden kann.

#### Einschätzung

Discover .NET kann durchaus zur Community-Startseite avancieren. Durch die offene und transparente Betreuung ist die notwendige Objektivität vollständig gegeben. Das Schöne an einem Projekt wie diesem ist, dass es bei stärkerer Beteiligung der Community immer bedeutsamer wird.

In der Codebasis sind durchaus interessante Konzepte wie beispielsweise die Verwendung von Statiq als Dokumentengenerator zu finden. Die Skalierbarkeit dieser Kombination ist unbestreitbar und die verwendete Technologie nahezu zeitlos.

Alles in allem bietet Discover .NET damit nicht nur eine interessante Anlaufstelle für .NET-Entwickler, sondern auch noch eine verwendbare Blaupause für zukünftige Webprojekte. ■

[1] Discover .NET bei GitHub,

[www.dotnetpro.de/SL2105HiddenGems1](http://www.dotnetpro.de/SL2105HiddenGems1)

[2] Discover .NET, <https://discoverdot.net>

[3] Discover .NET, API, <https://discoverdot.net/api>

[4] Discover .NET, Contribute to Discover .NET, [www.dotnetpro.de/SL2105HiddenGems2](http://www.dotnetpro.de/SL2105HiddenGems2)



#### Dr. Florian Rappl

ist Solution Architect bei smapiot mit Einsatzgebiet digitale Transformation. Als Microsoft MVP für Entwicklungstools befasst er sich mit Themen wie C#/.NET, TypeScript sowie skalierbaren Backends und Frontends in der Cloud.

@FlorianRappl

dnpCode

A2105HiddenGems



## Patterns & Clean Code mit C# und .NET

Inhaltlich wird den Teilnehmern bewusst gemacht, warum es von Bedeutung ist, den Code im Team verständlich zu gestalten, eine gemeinsame Meinung von „sauberen Code“ aufzubauen und daran zu feilen, eine bessere Codebasis anzustreben. Dafür werden gängige Patterns vorgestellt und in diversen Szenarien angewendet, um zu zeigen, dass zwar jede Software anders ist, aber der Weg dafür, oft bereits beschritten wurde – mit Patterns.

#### Was wird behandelt

- Modulare Codearchitektur
- GOF – Patterns
- Grundsätze von Clean Code
- Qualitätsmerkmale von gutem Code
- Code Refactoring - Testbarkeit herstellen
- Kopplung im Code erkennen und verbessern



Trainer: Christian Giesswein

3 Tage  
remote



Weitere Informationen unter [www.developer-media.de/trainings](http://www.developer-media.de/trainings) ••• Termine nach Absprache

Ihre Ansprechpartnerin: [Susanne Herl](mailto:susanne.herl@developer-media.de) • +49 (0)731 880058 - 835 • [susanne.herl@developer-media.de](mailto:susanne.herl@developer-media.de)