

NuGet.exe
in .csproj Verzeichnis
kopieren

Schnellstart: Erstellen und Veröffentlichen eines Pakets mithilfe von Visual Studio (.NET Framework, Windows)

13.05.2018 • 8 Minuten Lesedauer •  

In diesem Artikel

[Voraussetzungen](#)

[Erstellen eines Klassenbibliotheksprojekts](#)

[Konfigurieren der Projekteigenschaften für das Paket](#)

[Erstellen des ersten Manifests](#)

[Bearbeiten des Manifests](#)

[Ausführen des Befehls pack](#)

[Veröffentlichen des Pakets](#)

[Nächste Schritte](#)

Das Erstellen eines NuGet-Pakets aus einer .NET Framework-Klassenbibliothek umfasst das Erstellen der DLL in Visual Studio unter Windows, gefolgt von der Verwendung des Befehlszeilentools „nuget.exe“ zum Erstellen und Veröffentlichen des Pakets.

Hinweis

Dieser Schnellstart gilt ausschließlich für Visual Studio 2017 und höher unter Windows. Visual Studio für Mac enthält nicht die hier beschriebenen Funktionen. Verwenden Sie stattdessen die [dotnet-CLI-Tools](#).

Voraussetzungen

1. Installieren Sie über visualstudio.com eine beliebige Edition von Visual Studio 2017 oder höher mit einer beliebigen .NET-bezogenen Workload. Visual Studio 2017 enthält automatisch NuGet-Funktionen, wenn eine .NET-Workload installiert ist.
2. Installieren Sie die `nuget.exe`-CLI, indem Sie sie von nuget.org herunterladen, die `.exe`-Datei in einen geeigneten Ordner speichern und den Ordner der Umgebungsvariable `PATH` hinzufügen.

3. [Registrieren Sie sich für ein kostenloses Konto auf nuget.org](https://www.nuget.org), falls Sie noch kein Konto haben. Wenn Sie ein neues Konto erstellen, wird Ihnen eine Bestätigungs-E-Mail gesendet. Sie müssen das Konto bestätigen, bevor Sie ein Paket hochladen können.


Erstellen eines Klassenbibliotheksprojekts

Sie können ein vorhandenes Projekt in der .NET Framework-Klassenbibliothek für den Code verwenden, den Sie packen möchten, oder wie folgt ein einfaches Projekt erstellen:

1. Klicken Sie in Visual Studio auf **Datei > Neu > Projekt**, wählen Sie den Knoten **Visual C#** und dann die Vorlage „Klassenbibliothek (.NET Framework)“ aus, nennen Sie das Projekt „AppLogger“, und klicken Sie dann auf **OK**.
2. Klicken Sie mit der rechten Maustaste auf die resultierende Projektdatei, und wählen Sie **Erstellen** aus, um sicherzustellen, dass das Projekt ordnungsgemäß erstellt wurde. Die DLL befindet sich im Ordner „Debuggen“ (oder im Ordner „Release“, wenn Sie stattdessen diese Konfiguration erstellen).

In einem echten NuGet-Paket implementieren Sie viele hilfreiche Features, mit denen andere Apps erstellen können. Sie können die Zielframeworks beliebig festlegen. Beispiele finden Sie in den Leitfäden zu [UWP](#) und [Xamarin](#).

In dieser exemplarischen Vorgehensweise schreiben Sie jedoch keinen zusätzlichen Code, da eine Klassenbibliothek aus der Vorlage für die Erstellung eines Pakets ausreicht. Wenn Sie noch immer Funktionscode für das Paket benötigen, verwenden Sie folgenden:

C#	 Kopieren
<pre>using System; namespace AppLogger { public class Logger { public void Log(string text) { Console.WriteLine(text); } } }</pre>	

**Tipp**

Sofern es keinen Grund dafür gibt, der dagegen spricht, ist .NET Standard das bevorzugte Ziel für NuGet-Pakete, da es die größte Kompatibilitätsreichweite für verarbeitende Projekte bietet. Informationen dazu finden Sie unter [Erstellen und Veröffentlichen eines Pakets mithilfe von Visual Studio \(.NET Standard\)](#).

Konfigurieren der Projekteigenschaften für das Paket

Ein NuGet-Paket enthält ein Manifest (eine .nuspec-Datei), das relevante Metadaten enthält, z.B. den Paketbezeichner, die Versionsnummer, die Beschreibung, usw. Einige von diesen Metadaten können direkt aus den Projekteigenschaften entnommen werden, wodurch vermieden werden kann, dass diese im Projekt und im Manifest separat aktualisiert werden müssen. Dieser Abschnitt beschreibt, wo die entsprechenden Eigenschaften festgelegt werden müssen.

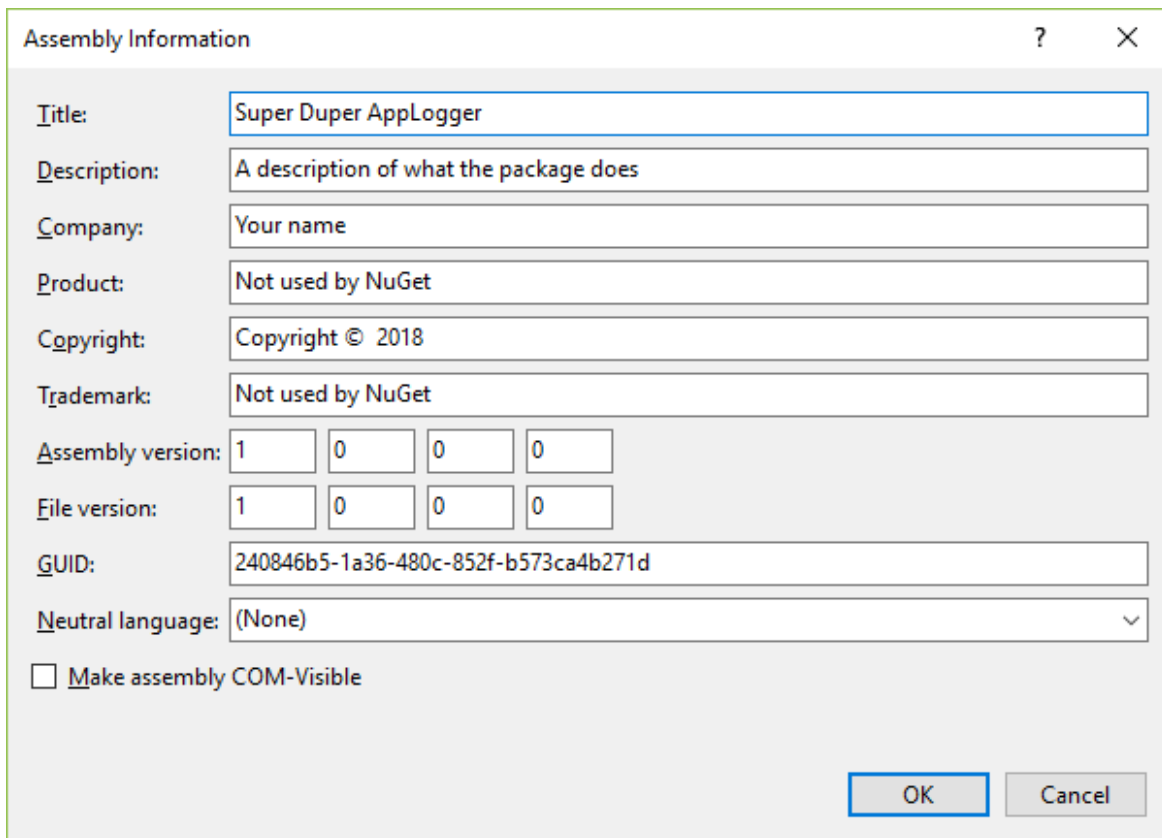
1. Wählen Sie den Menübefehl **Projekt > Eigenschaften** und dann die Registerkarte **Anwendung** aus.
2. Legen Sie im Feld **Assemblyname** einen eindeutigen Bezeichner für Ihr Paket fest.

Wichtig

Sie müssen dem Paket einen Bezeichner zuweisen, der auf nuget.org bzw. auf dem Host, den Sie verwenden, einzigartig ist. Für diese exemplarische Vorgehensweise wird empfohlen, im Namen „Sample“ oder „Test“ zu verwenden, da der Name in einem späteren Veröffentlichungsschritt öffentlich sichtbar gemacht wird (auch wenn es unwahrscheinlich ist, dass jemand versucht, es zu verwenden).

Bei dem Versuch, ein Paket mit einem Namen zu veröffentlichen, der bereits vorhanden ist, wird Ihnen ein Fehler angezeigt.

3. Klicken Sie auf die Schaltfläche **Assemblyinformationen**, wodurch ein Dialogfeld geöffnet wird, in dem Sie weitere Eigenschaften eingeben können, die dann im Manifest übernommen werden (siehe [.nuspec file reference - replacement tokens \(NUSPEC-Referenz: Ersetzungstoken\)](#)). Die am häufigsten verwendeten Felder sind **Titel**, **Beschreibung**, **Unternehmen**, **Copyright** und **Assemblyversion**. Diese Eigenschaften werden letztlich mit Ihrem Paket auf einem Host wie nuget.org angezeigt, weshalb Sie sich vergewissern sollten, dass diese ausführlich beschrieben sind.



The screenshot shows the 'Assembly Information' dialog box with the following values:

- Title: Super Duper AppLogger
- Description: A description of what the package does
- Company: Your name
- Product: Not used by NuGet
- Copyright: Copyright © 2018
- Trademark: Not used by NuGet
- Assembly version: 1.0.0.0
- File version: 1.0.0.0
- GUID: 240846b5-1a36-480c-852f-b573ca4b271d
- Neutral language: (None)
- ☐ Make assembly COM-Visible

4. Optional: Öffnen Sie zum direkten Anzeigen und Bearbeiten der Eigenschaften die Datei `Properties/AssemblyInfo.cs` im Projekt.
5. Wenn die Eigenschaften festgelegt sind, legen Sie die Projektkonfiguration auf **Release** fest, und erstellen Sie das Projekt neu, um die aktualisierte DLL zu generieren.

Erstellen des ersten Manifests

Mit einer DLL und den festgelegten Projekteigenschaften können Sie nun den Befehl `nuget spec` verwenden, um eine `.nuspec`-Startdatei aus dem Projekt zu erstellen. Dieser Schritt umfasst die relevanten Ersetzungstoken zum Auslesen von Informationen aus der Projektdatei.

Führen Sie `nuget spec` nur einmal aus, um das Startmanifest zu generieren. Beim Aktualisieren des Pakets ändern Sie die Werte in Ihrem Projekt, oder Sie bearbeiten das Manifest direkt.

1. Öffnen Sie eine Eingabeaufforderung, und navigieren Sie zum Projektordner, der die Datei `AppLogger.csproj` enthält.
2. Führen Sie den folgenden Befehl aus: `nuget spec AppLogger.csproj` Durch das Angeben eines Projekts erstellt NuGet ein Manifest, das dem Namen des Projekts

entspricht, in diesem Fall `AppLogger.nuspec`. Ebenfalls werden die Ersetzungstoken im Manifest eingeschlossen.

3. Öffnen Sie `AppLogger.nuspec` in einem Text-Editor, um die Inhalte zu überprüfen, die wie folgt angezeigt werden sollten:

XML Kopieren

```
<?xml version="1.0"?>
<package >
  <metadata>
    <id>Package</id>
    <version>1.0.0</version>
    <authors>YourUsername</authors>
    <owners>YourUsername</owners>
    <license type="expression">MIT</license>

    <projectUrl>http://PROJECT_URL_HERE_OR_DELETE_THIS_LINE</projectUrl>
    <iconUrl>http://ICON_URL_HERE_OR_DELETE_THIS_LINE</iconUrl>
    <requireLicenseAcceptance>false</requireLicenseAcceptance>
    <description>Package description</description>
    <releaseNotes>Summary of changes made in this release of the
package.</releaseNotes>
    <copyright>Copyright 2019</copyright>
    <tags>Tag1 Tag2</tags>
  </metadata>
</package>
```

Bearbeiten des Manifests

1. NuGet löst einen Fehler aus, wenn Sie versuchen, ein Paket mit Standardwerten in Ihrer `.nuspec`-Datei zu erstellen. Bevor Sie fortfahren können, müssen Sie deshalb die folgenden Felder bearbeiten. Eine Beschreibung dazu, wie diese verwendet werden, finden Sie unter [.nuspec file reference - optional metadata elements](#) ([NUSPEC-Referenz: Optionale Metadatenelemente](#)).

- `licenseUrl`
- `projectUrl`
- `iconUrl`
- `releaseNotes`
- `Tags`

2. Bei Paketen für die öffentliche Nutzung sollten Sie besonders auf die **Tags**-Eigenschaft achten, da Tags anderen dabei helfen, Ihr Paket auf Quellen wie `nuget.org` zu finden und dessen Funktion zu verstehen.

3. Zu diesem Zeitpunkt können Sie auch weitere Elemente zu Ihrem Manifest hinzufügen, wie unter [.nuspec file reference \(NUSPEC-Referenz\)](#) beschrieben wird.
4. Speichern Sie die Datei, bevor Sie fortfahren.

Ausführen des Befehls pack

1. Führen Sie den Befehl `nuget pack` über eine Eingabeaufforderung im Ordner mit Ihrer `.nuspec`-Datei aus.
2. NuGet generiert im aktuellen Ordner eine `.nupkg`-Datei in Form von *identifizierungs.version.nupkg*.

Veröffentlichen des Pakets

Sobald Sie eine `.nupkg`-Datei haben, können Sie diese, gemeinsam mit einem API-Schlüssel von `nuget.org`, über `nuget.exe` auf `nuget.org` veröffentlichen. Für `nuget.org` benötigen Sie `nuget.exe` 4.1.0 oder höher.

ⓘ Hinweis

Virenprüfung: Alle auf `nuget.org` hochgeladenen Pakete werden auf Viren überprüft und abgelehnt, falls Viren erkannt werden. Alle auf `nuget.org` aufgelisteten Pakete werden zudem regelmäßig überprüft.

Pakete, die auf `nuget.org` veröffentlicht werden, sind für andere Entwickler sichtbar, sofern Sie deren Auflistung nicht aufheben. Informationen zum privaten Hosten von Paketen finden Sie unter [Hosting packages](#) (Hosten von Paketen).

Erwerben des API-Schlüssels

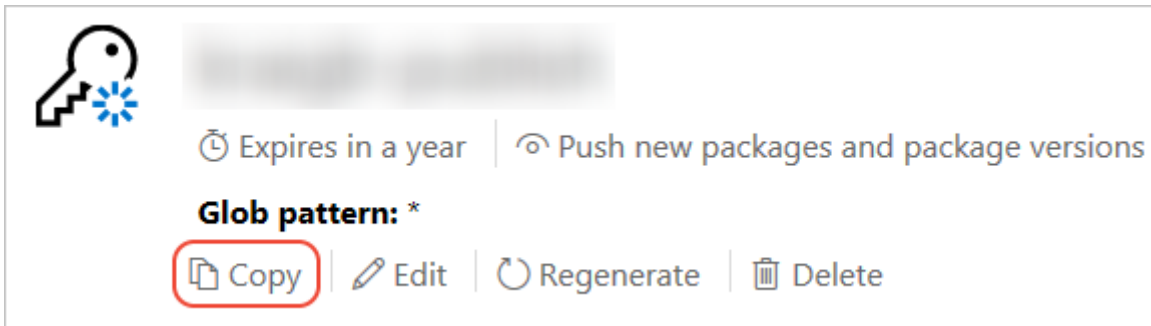
1. [Melden Sie sich bei Ihrem nuget.org-Konto an](#), oder erstellen Sie ein Konto, wenn Sie noch keines besitzen.

Weitere Informationen zum Erstellen Ihres Kontos finden Sie unter [Individuelle Konten](#).

2. Klicken Sie auf Ihren Benutzernamen (oben rechts) und anschließend auf **API Keys** (API-Schlüssel).
3. Wählen Sie **Erstellen**, geben Sie einen Namen für den Schlüssel ein, und wählen Sie **Bereiche auswählen > Push**. Geben Sie für **Globmuster** ein Sternchen (*) ein,

und wählen Sie dann **Erstellen** aus. (Weitere Informationen zu Bereichen finden Sie unten.)

- Nachdem der Schlüssel erstellt wurde, klicken Sie auf **Copy** (Kopieren), um den Zugriffsschlüssel abzurufen, den Sie in der CLI benötigen:

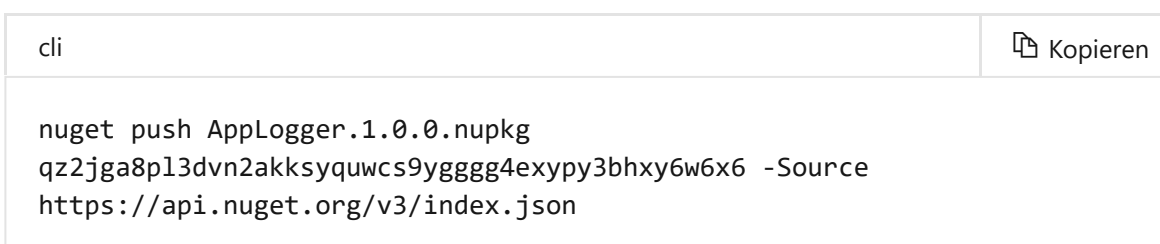


- Wichtig:** Speichern Sie Ihren Schlüssel an einem sicheren Ort, Sie können den Schlüssel später nicht erneut kopieren. Wenn Sie auf die Seite „API-Schlüssel“ zurückkehren, müssen Sie den Schlüssel erneut generieren, um ihn zu kopieren. Sie können den API-Schlüssel auch entfernen, wenn sie keine Pakete mehr mithilfe von Push über die CLI übertragen möchten.


Mit der Bereichsauswahl können Sie separate API-Schlüssel für verschiedene Zwecke erstellen. Jeder Schlüssel hat seinen Ablaufzeitraum und kann auf bestimmte Pakete (oder Globmuster) festgelegt werden. Jeder Schlüssel ist zudem auf bestimmte Vorgänge begrenzt: Push von neuen Paketen und Updates, Push von ausschließlich Updates oder Entfernen aus Listing. Durch das Festlegen des Gültigkeitsbereichs können Sie API-Schlüssel für verschiedene Personen erstellen, die Pakete für Ihre Organisation so verwalten, dass sie nur über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie unter [bereichsbezogene API-Schlüssel](#).

Veröffentlichen mit dem Befehl „nuget push“

- Öffnen Sie eine Befehlszeile, und navigieren Sie zu dem Ordner, der die `.nupkg`-Datei enthält.
- Führen Sie den folgenden Befehl aus, geben Sie dabei Ihren Paketnamen an, und ersetzen Sie den Schlüsselwert durch Ihren API-Schlüssel:



- In „NuGet.exe“ werden die Ergebnisse des Veröffentlichungsvorgangs angezeigt:


Ausgabe	 Kopieren
<pre>Pushing AppLogger.1.0.0.nupkg to 'https://www.nuget.org/api/v2/package'... PUT https://www.nuget.org/api/v2/package/ Created https://www.nuget.org/api/v2/package/ 6829ms Your package was pushed.</pre>	

Siehe [nuget push-Befehl](#).

Veröffentlichungsfehler

Fehler des Befehls `push` geben das Problem in der Regel an. Beispielsweise könnte das Problem darin liegen, dass Sie vergessen haben, die Versionsnummer in Ihrem Projekt anzupassen, und Sie versuchen ein Paket zu veröffentlichen, das bereits vorhanden ist.


Es werden Ihnen auch Fehler angezeigt, wenn Sie versuchen, ein Paket zu veröffentlichen, das einen Bezeichner verwendet, der bereits auf dem Host vorhanden ist. Zum Beispiel ist der Name „AppLogger“ bereits vorhanden. In diesem Fall gibt der `push`-Befehl folgende Fehlermeldung aus:

Ausgabe	 Kopieren
<pre>Response status code does not indicate success: 403 (The specified API key is invalid, has expired, or does not have permission to access the specified package.).</pre>	

Wenn Sie einen gültigen API-Schlüssel verwenden, den Sie eben erst erstellt haben, wird in dieser Nachricht ein Namenskonflikt angegeben, der durch den Abschnitt „permission“ dieser Fehlermeldung nicht ganz klar wird. Ändern Sie den Paketbezeichner, erstellen Sie das Projekt neu, erstellen Sie die Datei `.nupkg` neu, und versuchen Sie den `push`-Befehl erneut.

Verwalten des veröffentlichten Pakets

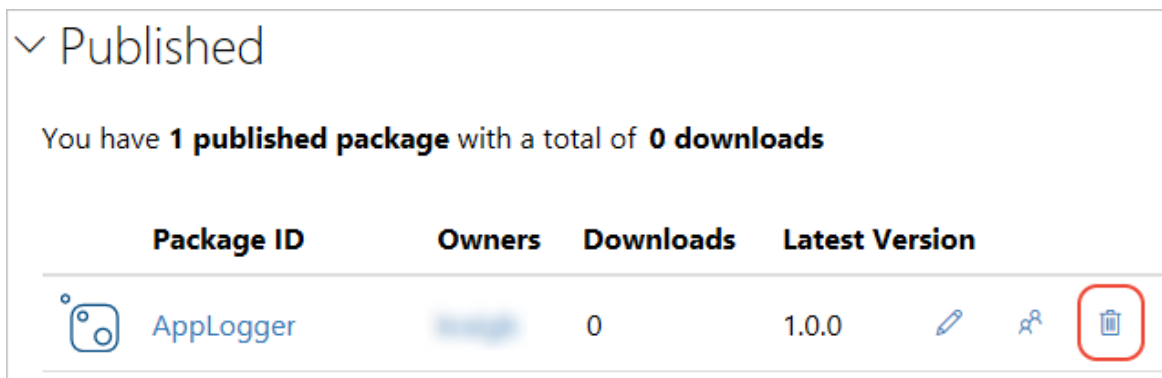
Wählen Sie in Ihrem Profil auf [nuget.org](#) **Pakete verwalten** aus, um das gerade veröffentlichte Paket anzuzeigen. Sie erhalten auch eine Bestätigungs-E-Mail. Beachten Sie, dass es möglicherweise etwas dauert, bis Ihr Paket indiziert wurde und anderen Benutzern in den Suchergebnissen angezeigt wird. In diesem Zeitraum wird auf Ihrer Paketseite folgende Nachricht angezeigt:

 **This package has not been published yet.** It will appear in search results and will be available for install/restore after both validation and indexing are complete. Package validation and indexing may take up to an hour. [Read more.](#)

Das ist schon alles! Sie haben gerade Ihr erstes NuGet-Paket auf nuget.org veröffentlicht, das andere Entwickler nun in ihren eigenen Projekten verwenden können.





Wenn Sie in dieser exemplarischen Vorgehensweise ein Paket erstellt haben, das keinen tatsächlichen Nutzen erfüllt (z.B. ein Paket mit einer leeren Klassenbibliothek), sollten Sie das Paket wie folgt *aus der Liste entfernen*:

1. Klicken Sie auf nuget.org auf Ihren Benutzernamen (oben rechts auf der Seite), und klicken Sie dann auf **Manage Packages** (Pakete verwalten).
2. Suchen Sie das Paket, das Sie aus der Liste entfernen wollen, unter **Published** (Veröffentlicht), und klicken Sie auf das Papierkorbsymbol auf der rechten Seite:

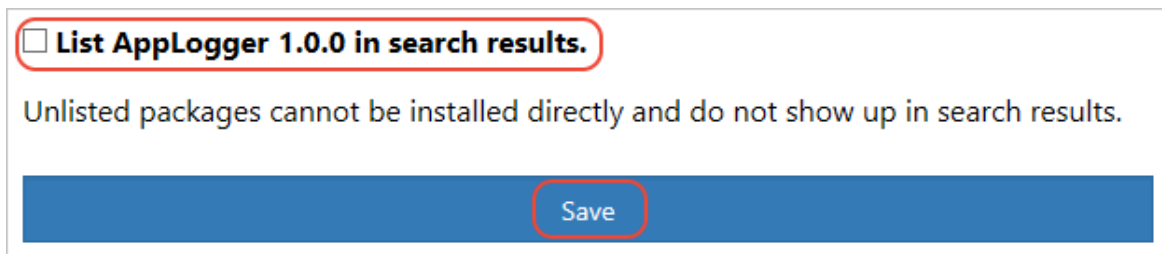


Published

You have **1 published package** with a total of **0 downloads**

Package ID	Owners	Downloads	Latest Version	
 AppLogger		0	1.0.0	  

3. Deaktivieren Sie auf der nachfolgenden Seite das Kontrollkästchen **List (package-name) in search results** ((Paketname) in Suchergebnissen auflisten), und klicken Sie auf **Save** (Speichern):



☐ **List AppLogger 1.0.0 in search results.**

Unlisted packages cannot be installed directly and do not show up in search results.

Save

Nächste Schritte

Herzlichen Glückwunsch zur Erstellung Ihres ersten NuGet-Pakets!

Erstellen eines Pakets

Klicken Sie für weitere Informationen zu den Features von NuGet auf folgende Links.

- [Veröffentlichen eines Pakets](#)
- [Vorabversionen von Paketen](#)
- [Unterstützung mehrerer Zielframeworks](#)
- [Paketversionsverwaltung](#)

- [Erstellen von lokalisierten Paketen](#)

Ist diese Seite hilfreich?

 Yes  No
