

# **Доклад на тему Системы управление пакетами NuGet**

**Операционные системы**

Румянцев Артём Олегович

# Содержание

|   |   |    |
|---|---|----|
| 1 | Введение в NuGet  | 5  |
| 2 | Поток пакетов между создателями, узлами и потребителями | 7  |
| 3 | Совместимость пакета с разными целевыми платформами     | 9  |
| 4 | Управление зависимостями                                | 10 |
| 5 | Что ещё делает NuGet                                    | 12 |
| 6 | Источники   | 14 |

## **Список иллюстраций**

## Список таблиц

# 1 Введение в NuGet

Ключевой инструмент для любой современной платформы разработки — это механизм, с помощью которого разработчики могут создавать, передавать друг другу и использовать полезный код. Часто такой код распределен по “пакетам”, включающим скомпилированный код (в виде библиотек DLL) и другое содержимое, необходимое использующим эти пакеты проектам.

Для .NET (в том числе .NET Core) механизмом совместного использования кода, поддерживаемым Майкрософт, является NuGet, который определяет, как создаются, размещаются и используются пакеты для .NET, а также предоставляет средства для каждой из этих ролей.

Проще говоря, пакет NuGet представляет собой отдельный ZIP-файл с расширением `.nupkg`, который содержит скомпилированный код (DLL), другие файлы, связанные с этим кодом, и описательный манифест, включающий такие сведения, как номер версии пакета. Разработчики, у которых есть код, к которому нужно предоставить общий доступ, создают пакеты и публикуют их на закрытых или открытых узлах. Потребители получают эти пакеты из соответствующих узлов, добавляют их в свои проекты, а затем вызывают функции пакета в коде своего проекта. При этом NuGet сам обрабатывает все промежуточные данные.

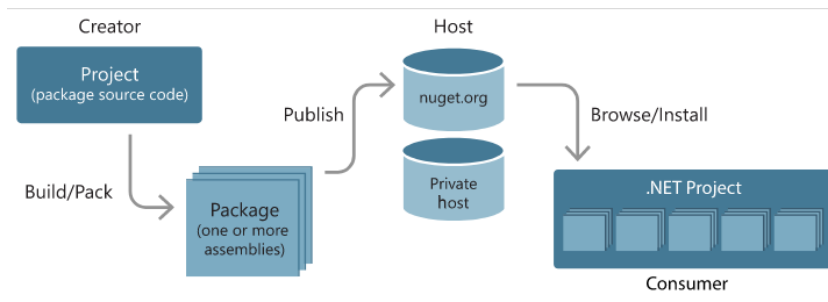
Так как NuGet поддерживает закрытые узлы наряду с открытым узлом `nuget.org`, с помощью пакетов NuGet вы можете делиться кодом, используемым в рамках организации или рабочей группы. Пакеты NuGet также являются удобным способом факторизовать свой код для использования только в собственных проектах. Иными словами, пакет NuGet является совместно

используемой единицей кода, однако не требует и не подразумевает какого-либо определенного способа предоставления общего доступа.

## 2 Поток пакетов между создателями, узлами и потребителями

Выполняя роль общедоступного узла, NuGet обслуживает центральный репозиторий с более чем 100 000 уникальных пакетов на сайте [nuget.org](https://nuget.org). Ежедневно эти пакеты используют миллионы разработчиков .NET и .NET Core. NuGet также позволяет размещать пакеты в частном порядке в облаке (например, в Azure DevOps), в частной сети или даже в вашей локальной файловой системе. Таким образом эти пакеты доступны только тем разработчикам, у которых есть доступ к узлу, что позволяет сделать их доступными для отдельных групп потребителей. С помощью параметров конфигурации можно также контролировать, какие узлы могут быть доступны любому компьютеру, тем самым гарантируя, что пакеты получают из определенных источников, а не из общего репозитория, такого как [nuget.org](https://nuget.org).

Независимо от своей природы, узел выступает в качестве точки подключения между создателями и потребителями пакета. Создатели разрабатывают полезные пакеты NuGet и публикуют их на узле. Потребители ищут полезные и совместимые пакеты на доступных узлах, скачивая эти пакеты и включая их в свои проекты. После установки в проекте API пакеты становятся доступны остальной части кода проекта.





## 3 Совместимость пакета с разными целевыми платформами

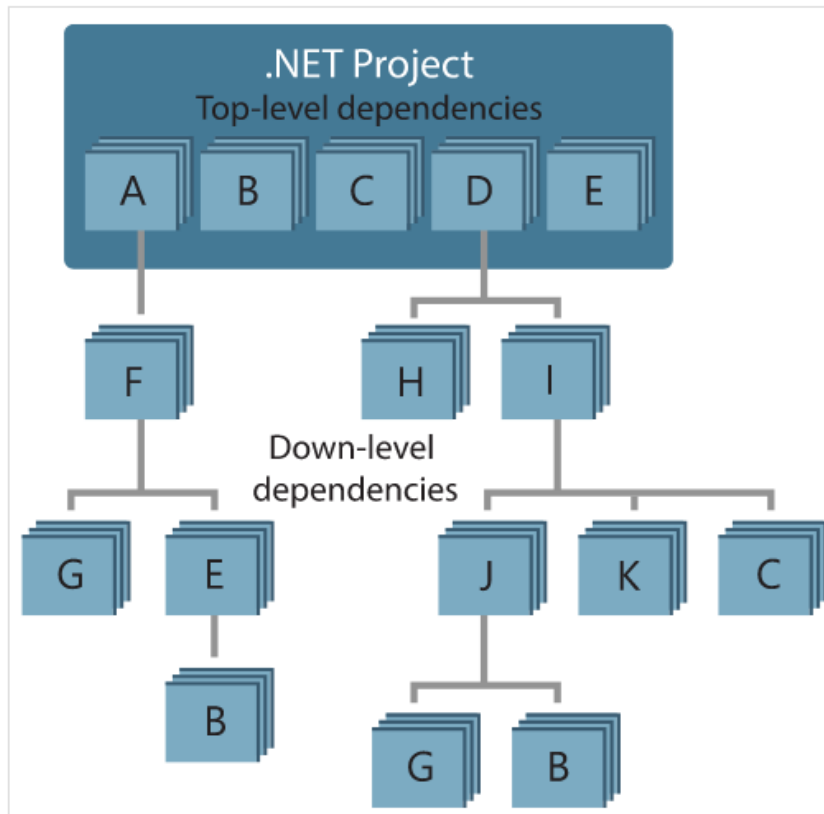
“Совместимый” пакет означает, что он содержит сборки, созданные по меньшей мере для одной целевой платформы .NET, совместимой с целевой платформой используемого проекта. Разработчики могут создавать пакеты, которые относятся к одной платформе, как и элементы управления универсальной платформы Windows (UWP), или они могут поддерживать более широкий набор целевых объектов. Для максимальной совместимости пакета разработчики используют версию .NET Standard, которую поддерживают все проекты .NET и .NET Core. Это наиболее эффективное средство как для создателей, так и для потребителей, так как единый пакет (обычно содержащий единую сборку) подходит для всех потребляющих проектов.

Разработчики пакетов, которым требуется API вне .NET Standard, с другой стороны, создают отдельные сборки для различных требуемых версий .NET Framework, которые они будут поддерживать, и включают все эти сборки в один пакет (который называется пакетом многоплатформенного нацеливания). Когда потребитель устанавливает такой пакет, NuGet извлекает только сборки, нужные в проекте. Это сводит к минимуму занимаемое пакетом место в конечном приложении или сборках, создаваемых данным проектом. Пакет настройки для различных версий, конечно, сложнее в обслуживании.

## 4 Управление зависимостями

Возможность легко брать за основу работу других — это одна из наиболее мощных функций системы управления пакетами. Соответственно, значительная часть работы NuGet заключается в управлении этим деревом или “схемой” зависимостей от имени проекта. Проще говоря, вам нужно заботиться только о тех пакетах, которые вы используете непосредственно в проекте. Если эти пакеты используют другие пакеты (которые, в свою очередь, также используют пакеты), все эти зависимости нижнего уровня обрабатывает NuGet.

На рисунке показан проект, зависящий от пяти пакетов, которые, в свою очередь, зависят от нескольких других.



Обратим внимание, что некоторые пакеты встречаются на графе зависимостей несколько раз. Например, существует три разных потребителя пакета B, и каждый из них может также указывать другую версию этого пакета (не показано). Это обычное дело, особенно для широко используемых пакетов. NuGet выполняет всю работу, чтобы определить, какая именно версия пакета B отвечает потребностям всех потребителей. Затем NuGet делает то же самое для всех других пакетов, независимо от того, насколько глубока схема зависимостей.

## 5 Что ещё делает NuGet

|

Чтобы обеспечить эффективную работу процессов, NuGet осуществляет некоторые оптимизации в фоновом режиме. В частности, NuGet управляет кэшем пакета и папкой глобальных пакетов, что позволяет упростить установку и повторную установку. Кэш позволяет избежать загрузки пакета, который уже установлен на компьютере. Папка глобальных пакетов позволяет в нескольких проектах совместно использовать один установленный пакет, тем самым уменьшая общий размер пакетов NuGet на компьютере. Это очень удобно, когда вы часто восстанавливаете большее количество пакетов, например, как на сервере сборки.

В рамках отдельного проекта NuGet управляет общей схемой зависимостей, что включает в себя разрешение нескольких ссылок на различные версии одного пакета. Довольно часто проект зависит от одного или нескольких пакетов, имеющих такие же зависимости. Некоторые из наиболее полезных пакетов служебных программ на сайте [nuget.org](http://nuget.org) используются многими другими пакетами. В общей схеме зависимостей вы легко можете иметь десять различных ссылок на разные версии одного пакета. Чтобы избежать переноса нескольких версий этого пакета в само приложение, NuGet определяет, какую отдельную версию могут использовать все потребители.

Кроме того, NuGet обслуживает все спецификации, связанные со структурированием пакетов (включая локализацию и отладочные символы) и ссылками на них (включая диапазоны версий и предварительные версии). NuGet также имеет различные API для работы со своими службами программно и предоставляет

поддержку разработчикам, которые пишут расширения Visual Studio и шаблоны проектов.

## 6 Источники

- <https://ru.wikipedia.org/wiki/NuGet>
- <https://www.nuget.org/>