

# NuGet PACKAGES

NuGet е вградена функционалност във VisualStudio, която ви позволява да изтеглите кодове на други автори намиращи се в „галерия” за свободен достъп. Принципно самите кодове имат някакви лицензи(условия с които трябва да се съобразите), но те се появяват по време на инсталация.

Вероятното тълкувание на NuGet е комбинация от игра на думи – „Взemi нещо ново” към проектът си.

какви типове код може да инсталираме през nuget? само библиотеки и тулове ли? или може и фреймуорци някакви



През **NuGet** можеш да инсталираш основно библиотеки, инструменти и помощни пакети (tools) за разработка, но не и фреймуъркове като .NET или други платформи за изпълнение. По-конкретно, ето какви типове код можеш да инсталираш през NuGet:

1. **Библиотеки** – това са пакети, съдържащи готови класове, методи и функционалности, които можеш да използваш директно в проекта си. Пример: библиотеки за работа с JSON, за Unit Testing, за база данни и т.н.
2. **Тулове (Tools)** – това са помощни инструменти, които може да използваш по време на разработката, като генератори на код, статичен анализ на кода и други помощни утилити.
3. **SDK и API** – някои SDK (software development kits) за работа с външни услуги или API-та също могат да се инсталират като пакети през NuGet.
4. **Разширения** – добавят специфични функционалности или интеграции към проекта ти, които могат да улеснят разработката или да автоматизират определени процеси.

Важно е да се отбележи, че **фреймуъркове като .NET не се инсталират през NuGet**, защото те са основната платформа, върху която се стартира приложението. NuGet се използва за допълнителни пакети, които разширяват функционалността на твоя код, но не и за основния фреймуърк, на който той ще работи.

**Entity Framework** – Фреймуорк, който ще използвате 2 срок (много улеснява писането в бази данни)

Ентити фреймуорк обаче може нали? Той към кое от по горните 4 категории спада според теб?



Да, **Entity Framework (EF)** може да се инсталира през **NuGet** и това е често срещан начин да го добавиш в проекта си. Entity Framework попада в категорията на **библиотеките**, тъй като предоставя набор от класове и методи, които улесняват работата с бази данни чрез ORM (object-relational mapping) модел. Чрез него можеш да работиш с базата данни, използвайки C# обекти, вместо да пишеш директно SQL заявки.

Така че, EF е библиотека, която предоставя допълнителна функционалност в твоя проект, но се инсталира като пакет през NuGet, а не като основен фреймуърк.

Някъде бе ми попаднало сравнение между фреймуорк и библиотека и там бе описано, че за фреймуорка е типично да има инвържън контрол, докато при библиотеките ние се обръщаме към тях само при нужда. По този критерий EF към фреймуорк ли би трябвало да го причиним или към библиотека?



Ти си попаднал на много важен критерий за разграничаване на **фреймуърк** от **библиотека** – инверсията на контрола (**Inversion of Control**). Това означава, че при фреймуърковете управлението на потока на изпълнение е в ръцете на самия фреймуърк, а не на разработчика. Разработчикът предоставя определени части от приложението (като класове или конфигурации), но фреймуъркът решава кога и как да ги използва.

## Къде попада Entity Framework по този критерий?

Entity Framework се счита за **фреймуърк** поради следните причини:

1. **Инверсията на контрола (Inversion of Control):** EF управлява взаимодействието с базата данни и решава кога да създаде, актуализира или изтрие записи. Той предоставя механизми като контекста на базата данни (**DbContext**), който се грижи за състоянието на обектите и тяхното съответствие с базата данни. Разработчикът конфигурира EF и определя модела на данните, но самият EF управлява операциите в базата.
2. **Фиксиран работен поток:** EF определя стъпките, които трябва да се следват за работа с данни, като задава конвенции и структура на взаимодействието с базата данни. Той предоставя и възможности за надграждане чрез конфигурации, но основният контрол остава в ръцете на EF.
3. **Поема контрола върху ресурсите:** EF управлява отварянето и затварянето на връзките с базата данни и синхронизирането на обектите с базата данни. Това е част от концепцията за "инверсия на контрол", защото ти предоставяш на фреймуърка моделите и базовите конфигурации, а той решава кога и как да ги използва.

### Заклучение:

По този критерий Entity Framework е **фреймуърк**, въпреки че го инсталираме през **NuGet** като библиотека. Основната разлика е, че EF управлява значителна част от взаимодействието с базата данни и процесите, докато библиотеки се използват само когато ти изрично ги извикаш.