Внедрение зависимостей

№ урока: 5 Курс: Внедрение зависимостей

Средства обучения: Visual Studio .NET SDK

Обзор, цель и назначение урока

В этом уроке будет рассмотрена проблема написания сильно связанного кода. Варианты уменьшения связанности между компонентами через выделение зависимостей в отдельные классы и внедрение дополнительного уровня абстракции. Будет рассмотрен принцип инверсии управления и один из шаблонов который его реализует – Dependency Injection.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое инверсия управления
- Настраивать внедрение зависимостей в ASP.NET Core приложениях
- Разбираться с разными способами настройки сервисов с разным временем жизни

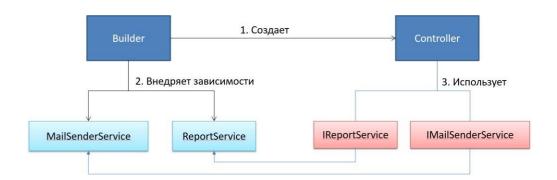
Содержание урока

- 1. Inversion of Control. Dependency Injection
- 2. Что такое зависимость.
- 3. Dependency Injection в ASP.NET Core
- 4. Captured Dependencies
- 5. Использование нескольких зависимостей с одним интерфейсом
- 6. Внедрение зависимостей в методы действий
- 7. Использование зависимостей для middleware

Резюме

Inversion Of Control (IoC) – Это абстрактный принцип, набор рекомендаций по написанию слабо связанного кода. Суть которого в том, что каждый компонент системы должен быть как можно более изолированным от других, не полагаясь в своей работе на детали реализации других компонентов.

Dependency Injection - шаблон, описывающий технику внедрения зависимости программному компоненту.





Page | 1

Title: ASP.NET Core

Lesson: 5

itvdn.com

Способы внедрения зависимостей можно классифицировать по

Способы внедрения зависимостей

- 1. Через конструктор
- 2. Через свойство
- 3. Через интерфейс

Способы создания зависимостей

- 1. С помощью фабрики
- 2. С помощью контейнера

В ASP.NET Core при регистрации зависимости можно указать три варианта времени жизни зависимости:

Transient – на каждое требование контейнер создает новый экземпляр сервиса **Scoped** – сервис создается один раз на запрос Singleton – сервис один на все приложение

Закрепление материала

- Что такое Inversion of Control?
- Как происходит регистрация сервисов в ASP.NET Core?
- Какие есть варианты жизненного цикла сервиса?
- Что будет если на один интерфейс в приложении будет зарегистрировано несколько реализаций сервсиа?

Дополнительное задание

Задание

Создайте класс с методами для арифметических операций с именем CalcService. Создайте контроллер, который будет содержать методы действия для арифметических операций. Hастройте DI таким образом, чтобы контроллер воспользовался сервисом CalcService.

Самостоятельная деятельность учащегося

Задание 1

Создайте контроллер, который выводит пользователю список строковых значений. Сделайте два сервиса, один который предоставляет массив названий дней недели, второй, который предоставляет массив с названых месяцев. Реализуйте через DI передачу контроллеру одного из сервисов, используемый сервис зависит от настроек DI.

Рекомендуемые ресурсы

Внедрение зависимостей в ASP.NET Core

https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-3.1



Title: ASP.NET Core

Lesson: 5

CyberBionic Systematics ® 19 Mariny Raskovoy Str., 5 floor Kyiv, Ukraine

E-mail: edu@cbsystematics.com Site: www.edu.cbsystematics.com |

itvdn.com

t. +380 (44) 361-8937