

Лекция 1

GIT, ASP.NET, Postman

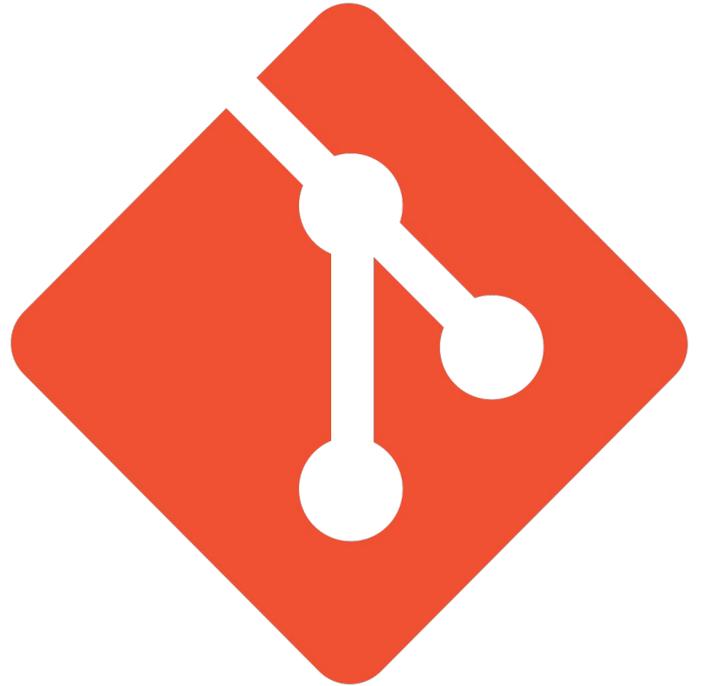
Харьковский Александр
.NET Developer

Темы

- Система контроля версий GIT. Методы ведения проекта.
- Платформа для создания веб-приложений ASP.NET Core
- Настройка служб и конвейера обработки запросов в веб-приложении
- Привязка моделей (model binding) и основы валидации с использованием аннотации данных
- Тестирование API веб-приложения с использованием Postman

Система контроля версий GIT

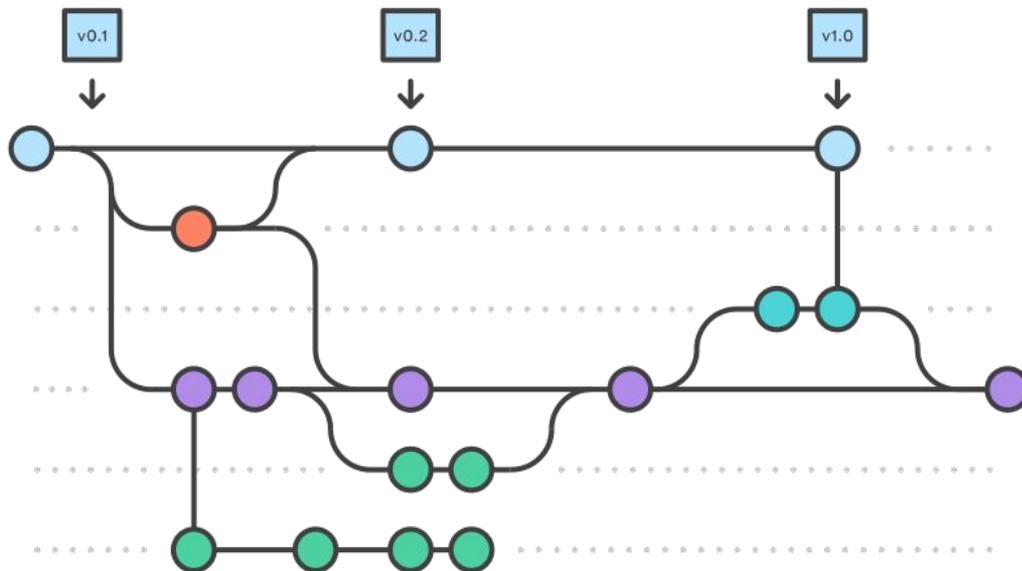
Git - это распределенная система контроля версий. Разработана Линусом Торвальдсом для использования в управлении разработки ядра Linux.



Методы ведения проекта в GIT



Git Flow - популярная методология ветвления и слияния.



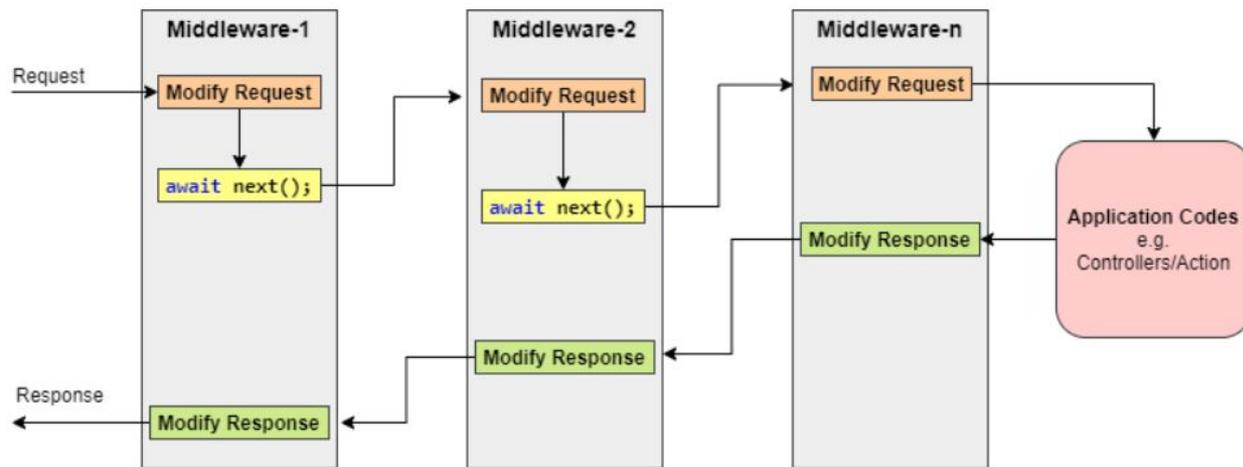
ДЕМО

Платформа для создания веб-приложений

Платформа ASP.NET Core представляет технологию от компании Microsoft, предназначенную для создания различного рода веб-приложений.

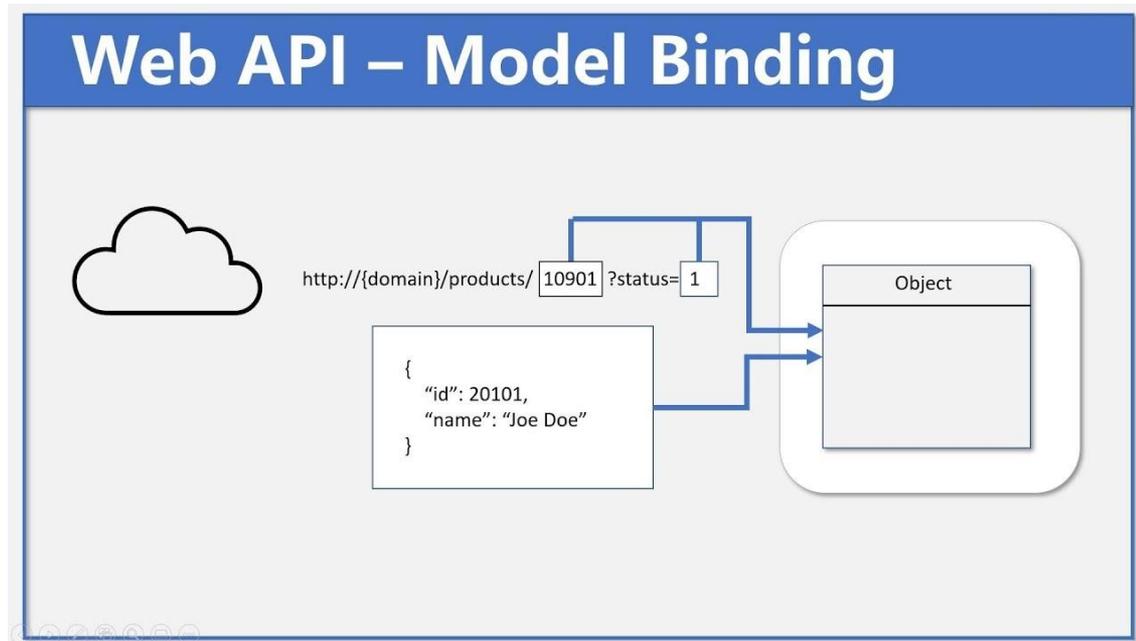
ASP.NET Core

Настройка служб и конвейера обработки запросов в веб-приложении



Привязка моделей (model binding)

Привязка модели или Model binding представляет механизм сопоставления значений из HTTP-запроса с параметрами метода контроллера. При этом параметры могут представлять как простые типы (int, float и т.д.), так и более сложные типы данных, например, объекты классов.



Основы валидации с использованием аннотации данных

Важную роль в ASP.NET Core играет валидация входных данных. Валидация позволяет проверить входные данные на наличие неправильных, корректных значений и должным образом обработать эти значения.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.ComponentModel.DataAnnotations;
6 using Microsoft.AspNetCore.Mvc;
7
8
9 namespace WebApplication1
10 {
11     0 references
12     public class Class
13     {
14         [Required]
15         [Display(Name = "name")]
16         0 references | 0 exceptions
17         public String ClientName { get; set; }
18
19         [UIHint("Date")]
20         [Required(ErrorMessage = "Please enter a date")]
21         [Remote("ValidateDate", "Home")]
22         0 references | 0 exceptions
23         public DateTime Date { get; set; }
24
25         [MustBeTrue(ErrorMessage = "You must accept the terms")]
26         0 references | 0 exceptions
27         public bool TermsAccepted { get; set; }
28     }
29 }
```

ДЕМО

Тестирование API веб-приложения с использованием Postman

Postman - приложение для тестирования API, где можно отправить входные данные в запросе и проверить точность выходных данных.



POSTMAN

Тестирование API веб-приложения с использованием Postman

The screenshot displays the Postman interface. On the left, the 'Collections' tab is active, showing a tree view of API collections: 'Invitro' (11 requests), 'Delivery', 'Loyalty', 'FrontendApi', 'V2', 'SoftCheck', 'HardCheck', 'MedicalCardService', and 'ResultClientVisitsCore'. The 'SoftCheck' collection is selected, and a 'POST' request is visible.

The main workspace shows the configuration for the 'POST SoftCheck' request:

- Environment: No Environment
- Request Method: POST
- URL: `{{loyalty_frontend_api}}/api/v2/Loyalty/SoftCheck`
- Buttons: Send, Save
- Params: Authorization, Headers (13), Body (selected), Pre-request Script, Tests, Settings
- Additional options: Cookies, Code

The 'Query Params' section is expanded, showing a table with the following structure:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

ДЕМО