



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет Информационных технологий
Кафедра Информатики и информационных технологий

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № _18_

Дисциплина: _Backend разработка

Тема:

Выполнил(а): студент(ка) группы __231-336__

_____ Канищев И.М _____
(Фамилия И.О.)

Дата, подпись _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание) (Оценка)

Дата, подпись _____
(Дата) (Подпись)

Замечания:

Москва
2025

1. Настройка проекта

1.1 Установка пакета

```
dotnet add package prometheus-net.AspNetCore
```

1.2 Program.cs

```
using Microsoft.AspNetCore.Builder;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
using Prometheus;

var builder = WebApplication.CreateBuilder(args);

// Добавляем MVC
builder.Services.AddControllersWithViews();

var app = builder.Build();

// Middleware Prometheus для сбора метрик HTTP
app.UseHttpMetrics();

// Статические файлы и маршрутизация
app.UseStaticFiles();
app.UseRouting();

// Endpoint для Prometheus метрик
app.UseEndpoints(endpoints =>
{
    endpoints.MapMetrics("/metrics"); // Prometheus собирает метрики здесь
    endpoints.MapControllerRoute(
```

```
        name: "default",

        pattern: "{controller=Home}/{action=Index}/{id?}";

    });

app.Run ();
```

3. Представления

Views/Home/Index.cshtml

```
@{

    ViewData["Title"] = "Главная";

}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8" />

    <title>@ViewData["Title"]</title>

</head>

<body>

    <h1>Мониторинг через Prometheus</h1>

    <p>Эта страница демонстрирует сбор метрик HTTP-запросов и
пользовательских счетчиков.</p>

    <a href="/Home/Error">Сгенерировать ошибку</a>

</body>

</html>
```

Views/Home/Error.cshtml

```
@{

    ViewData["Title"] = "Ошибка";
```

```
}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8" />

    <title>@ViewData["Title"]</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            background-color: #f8d7da;

            color: #721c24;

            text-align: center;

            padding: 50px;

        }

        h1 { font-size: 48px; }

        p { font-size: 20px; }

        a { color: #0056b3; text-decoration: none; font-weight: bold; }

        a:hover { text-decoration: underline; }

    </style>

</head>

<body>

    <h1>Что-то пошло не так</h1>

    <p>Произошла ошибка при обработке вашего запроса.</p>

    <p>Вернитесь на <a href="/">главную страницу</a>.</p>

</body>

</html>
```

4. Установка Grafana и подключение к Prometheus

1. Установка Prometheus

1. Скачиваем Prometheus с официального сайта: <https://prometheus.io/download/>
2. Распакуем архив в удобную директорию.
3. В корне создаем файл конфигурации `prometheus.yml`:

```
global:

  scrape_interval: 5s

scrape_configs:

  - job_name: 'aspnetcore-app'

    static_configs:

      - targets: ['localhost:7247'] # Порт вашего ASP.NET Core приложения
```

4. Запускаем Prometheus:

Prometheus будет доступен по адресу: `http://localhost:9090`

Endpoint `/metrics` приложения должен быть доступен (например, `http://localhost:7247/metrics`).

2. Установка Grafana

1. Скачиваем Grafana с официального сайта: <https://grafana.com/grafana/download>
2. Установим и запустим Grafana.
3. По умолчанию Grafana доступна по адресу: `http://localhost:3000`

- Логин: `admin`
- Пароль: `admin` (потом можно изменить)

3. Подключение Prometheus к Grafana

1. В Grafana перейдём в **Configuration** → **Data Sources** → **Add data source**
2. Выбираем **Prometheus**
3. В поле URL указываем адрес Prometheus, например:

```
http://localhost:9090
```

4. Нажмите **Save & Test** — Grafana должна подтвердить подключение.

4. Создание дашборда для метрик ASP.NET Core

1. В Grafana выберите **Create** → **Dashboard** → **Add new panel**

2. В поле **Query** выберите источник Prometheus.
3. Введите нужную метрику, например:

`home_index_requests_total`

4. Настройте визуализацию: график, счетчик, таблицу.
5. Сохраните дашборд.

Теперь мы видим в реальном времени:

- количество запросов к Home/Index
- стандартные HTTP метрики (`http_request_duration_seconds_sum`, `http_requests_received_total`)
- ошибки и статус-коды

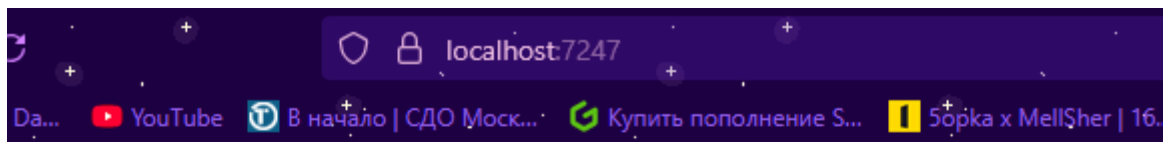
5. Полезные Prometheus метрики для ASP.NET Core

- **`home_index_requests_total`** — пользовательский счетчик запросов
- **`http_requests_total`** — количество HTTP запросов
- **`http_request_duration_seconds`** — время обработки запросов
- **`http_response_size_bytes`** — размер ответа

Вы можете добавлять пользовательские метрики через `Prometheus.Metrics.CreateCounter` или `Gauge`.

5. Проверка работы мониторинга

1. Запускаем приложение.
2. Открываем страницу /Home/Index → счетчик `home_index_requests_total` увеличивается.

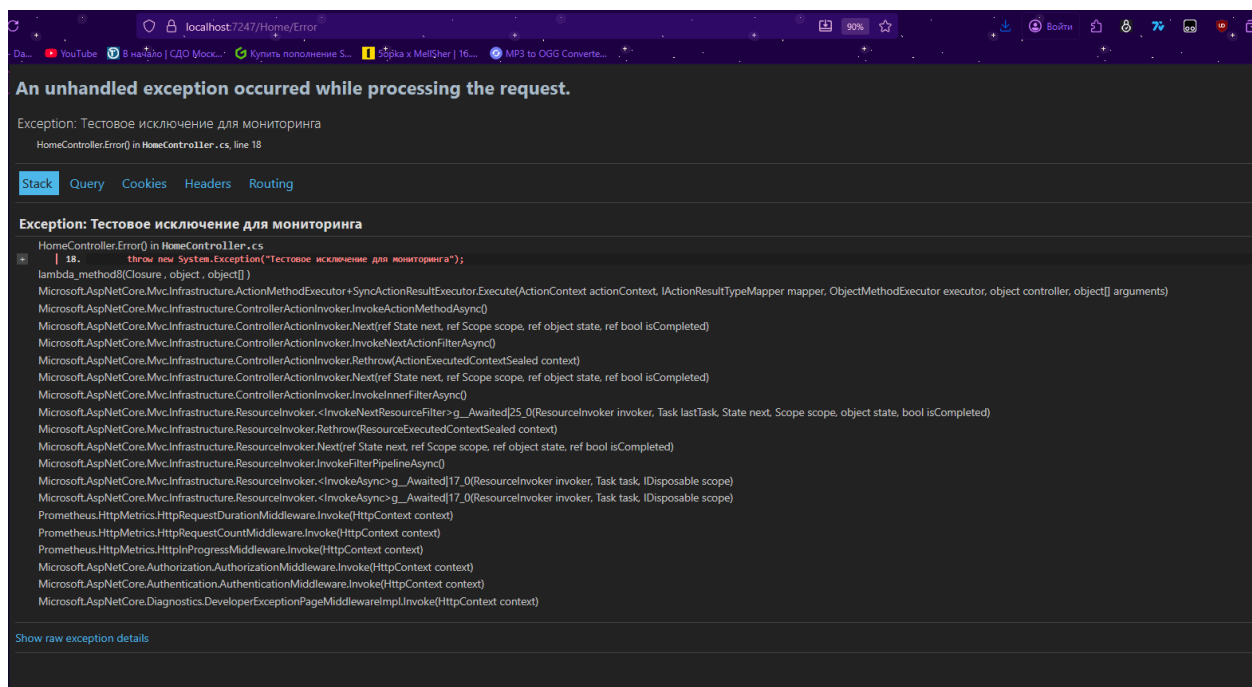


Мониторинг через Prometheus

Эта страница демонстрирует работу Prometheus метрик.

[Сгенерировать ошибку](#)

3. Переходим на `/Home/Error` → срабатывает исключение.



4. Переходим на `/metrics` → Prometheus собирает метрики:
 - o `home_index_requests_total`
 - o стандартные HTTP метрики (время отклика, количество запросов, статус-коды).

```

localhost7247/metrics

# HELP http_request_duration_seconds The duration of HTTP requests processed by an ASP.NET Core application.
# TYPE http_request_duration_seconds histogram
http_request_duration_seconds_sum{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}" } 0.0629011
http_request_duration_seconds_count{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}" } 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.001"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.002"} 1
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.004"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.008"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.016"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.032"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.064"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.128"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.256"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.512"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="1.024"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="2.048"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="4.096"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="8.192"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="16.384"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="32.768"} 3
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}",le="+Inf"} 3
http_request_duration_seconds_sum{code="200",method="GET",controller="",action="",endpoint="" } 0.1504381
http_request_duration_seconds_count{code="200",method="GET",controller="",action="",endpoint="" } 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.001"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.002"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.004"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.008"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.016"} 1
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.032"} 1
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.064"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.128"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.256"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="0.512"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="1.024"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="2.048"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="4.096"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="8.192"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="16.384"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="32.768"} 4
http_request_duration_seconds_bucket{code="200",method="GET",controller="",action="",endpoint="",le="+Inf"} 4
http_request_duration_seconds_sum{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}" } 0.22852070000000002
http_request_duration_seconds_count{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}" } 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.001"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.002"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.004"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.008"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.016"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.032"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.064"} 0
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.128"} 1
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.256"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="0.512"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="1.024"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="2.048"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="4.096"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="8.192"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="16.384"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="32.768"} 2
http_request_duration_seconds_bucket{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}",le="+Inf"} 2
# HELP http_requests_received_total Provides the count of HTTP requests that have been processed by the ASP.NET Core pipeline.
# TYPE http_requests_received_total counter
http_requests_received_total{code="200",method="GET",controller="Home",action="Index",endpoint="{controller=Home}/{action=Index}/{id?}" } 3
http_requests_received_total{code="200",method="GET",controller="",action="",endpoint="" } 4
http_requests_received_total{code="200",method="GET",controller="Home",action="Error",endpoint="{controller=Home}/{action=Index}/{id?}" } 2
# HELP http_requests_in_progress The number of requests currently in progress in the ASP.NET Core pipeline. One series without controller/action label values counts all
# TYPE http_requests_in_progress gauge
http_requests_in_progress{method="GET",controller="",action="",endpoint="" } 0
# HELP home_index_requests_total Количество запросов к Home/Index
# TYPE home_index_requests_total counter
home_index_requests_total 3
# HELP dotnet_collection_count_total GC collection count

```

5. Подключаем Grafana к Prometheus для визуализации метрик:

