Mini HTTP Server

Responses папка

"**Responses**" папката ще съдържа класове и интерфейси, които съдържат и манипулират информация за "**HTTP Responses**".

IHttpResponse

Създайте интерфейс, който се казва "**IHttpResponse**" и ще се съдържа следните свойства и методи:

```
public interface IHttpResponse
{
    HttpResponseStatusCode StatusCode { get; set; }

    IHttpHeaderCollection Headers { get; }

    byte[] Content { get; set; }

    void AddHeader(HttpHeader header);

    byte[] GetBytes();
}
```

HttpResponse

Създайте клас, който се казва "**HttpResponse**" и имплементира "**IHttpResponse**" интерфейса.

```
public class HttpResponse : IHttpResponse
{
    public HttpResponse()
    {
        this.Headers = new HttpHeaderCollection();
        this.Content = new byte[0];
    }

    public HttpResponse(HttpResponseStatusCode statusCode)
        : this()
    {
        CoreValidator.ThrowIfNull(statusCode, name: nameof(statusCode));
        this.StatusCode = statusCode;
    }

    public HttpResponseStatusCode StatusCode { get; set;  }

    public IHttpHeaderCollection Headers { get; }

    public byte[] Content { get; set; }

    public void AddHeader(HttpHeader header)...

    public byte[] GetBytes()...

    public override string ToString()...
}
```

Както виждате " **HttpResponse** " съдържа "**StatusCode**", "**Headers**", "**Content**" и т.н. Това са единствените неща, от които ние се нуждаем за сега. "**HttpResponse**" се инициализира с обект с Null ли по подразбиране стойности.

Сървърът получва "**Requests**" в текстов формат и трябва върне "**Responses**" в същият формат.

Репрезентацията на низа от "**HTTP Responses**" са в следният формат:

```
{protocol} {statusCode} {status}
{header1key}: {header1value}
{header2key}: {header2value}
...
<CRLF>
{content}
```

ЗАБЕЛЕЖКА: Както вече знаете, съдържанието (**Response body**) не е задължително.

Сега, докато изграждаме нашият "**HttpResponse**" обект, може да присвоим стойност за нашият "**StatusCode**" или може да го оставим за напред. Най-често ще присвояваме стойностите чрез конскструктора.

AddHeader() метод

We can add **Headers** to it, gradually with the processing of the **Request**, using the **AddHeader()** method.

Можем добавяме "**Headers**", като използваме "**AddHeader()**" метода.

```
public void AddHeader(HttpHeader header)
{
    CoreValidator.ThrowIfNull(header, name: nameof(header));
    this.Headers.Add(header);
}
```

Другите пропъртита, "**StatusCode**" и "**Content**" могат да бъдат присвоени стойности от "външният свят", като използват публичните им сетъри.

Сега нека да видим "**ToString()**" и "**GetBytes()**" какво правят.

ToString() метод

"**ToString()**" метода формира "**Response**" реда – този ред съдържа протокола, статус кода, статус и "**Response Headers**", като завършва с празен ред. Тези пропъртита са съединени в един низ и върнати в края.

```
public override string ToString()
{
    StringBuilder result = new StringBuilder();

    result
        .Append($"{GlobalConstants.HttpOneProtocolFragment} {(int)this.StatusCode} {this.StatusCode.ToString()}")
        .Append(GlobalConstants.HttpNewLine)
        .Append(this.Headers)
        .Append(GlobalConstants.HttpNewLine);

    result.Append(GlobalConstants.HttpNewLine);

    return result.ToString();
}
```

И точно сега се нуждаем от "**GetBytes()**" метода.

GetBytes() метод

And with that we are finished with the **HTTP work** for now. We can proceed to the main functionality of the Server.

"**GetBytes()**" метода конвертира резултата от "**ToString()**" метода до "**byte[]**" масив, и долепя към него "**Content bytes**", затова формираме целият "**Response**" до байт формат. Точна това, което трябва да изпратим до сървъра.

И вече приключихме с работата по нашият HTTP сървър за сега.

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).