# Mini HTTP Server

### Responses папка

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

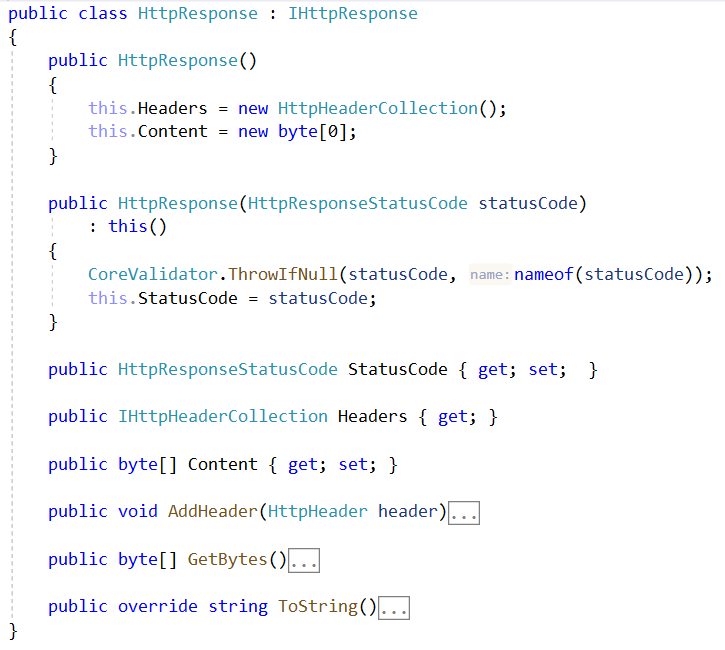
#### IHttpResponse

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



#### HttpResponse

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



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 или стойности по подразбиране.

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

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

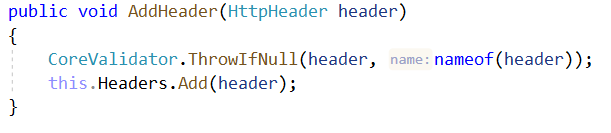
{protocol} {statusCode} {status}  
{header1key}: {header1value}  
{header2key}: {header2value}  
...  
<CRLF>  
{content}

**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 конструктора.

##### AddHeader() метод

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

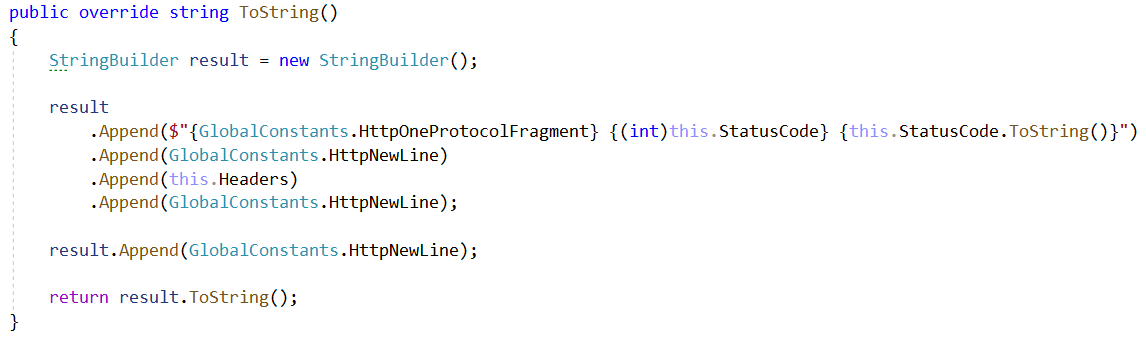


Другите пропъртита, "StatusCode" и "Content" могат да бъдат присвоени стойности от "външният свят", като използват публичните им сетъри.

Сега нека да разгледаме методите "ToString()" и "GetBytes()"

##### 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.



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

##### GetBytes() метод

"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.