# Контекст контроллера

Последнее обновление: 20.03.2022

При обращении к контроллеру среда ASP.NET создает для этого контроллера контекст, который содержит различные связанные с контроллером данные. Для получения контекста в классе контроллера нам доступно свойство **ControllerContext**, которое представляет одноименный класс ControllerContext. Этот класс определяет ряд важный свойств:

- **HttpContext**: содержит информацию о контексте запроса
- **ActionDescriptor**: возвращает дескриптор действия - объект ActionDescriptor, который описывает вызываемое действие контроллера
- **ModelState**: возвращает словарь ModelStateDictionary, который используется для валидации данных, отправленных пользователем
- **RouteData**: возвращает данные маршрута

Для получения информации о запросе нас прежде всего будет интересовать свойство HttpContext, которое представляет объект `Microsoft.AspNetCore.Http.HttpContext`. В принципе это тот же самый объект, который нам доступен в любом компоненте middleware в ASP.NET Core. Этот объект также доступен через свойство HttpContext класса контроллера. То есть следующие вызовы будут обращаться к одному и тому же объекту:

```
var ctx1 = ControllerContext.HttpContext;
var ctx2 = HttpContext;
```

Объект HttpContext инкапсулирует всю информацию о запросе. В частности, он определяет следующие свойства:

- **Request**: содержит собственно информацию о текущем запросе.
- **Response**: управляет ответом
- **User**: представляет текущего пользователя, который обращается к приложению
- **Session**: объект для работы с сессиями

## Response

Свойство **HttpContext.Response** представляет объект **HttpResponse** и позволяет управлять ответом на запрос, в частности, устанавливать заголовки ответа, куки, отправлять в выходной поток некоторый ответ. Этот же объект доступен через свойство **Response** класса Conroller. Среди свойств объекта Response можно выделить следующие:

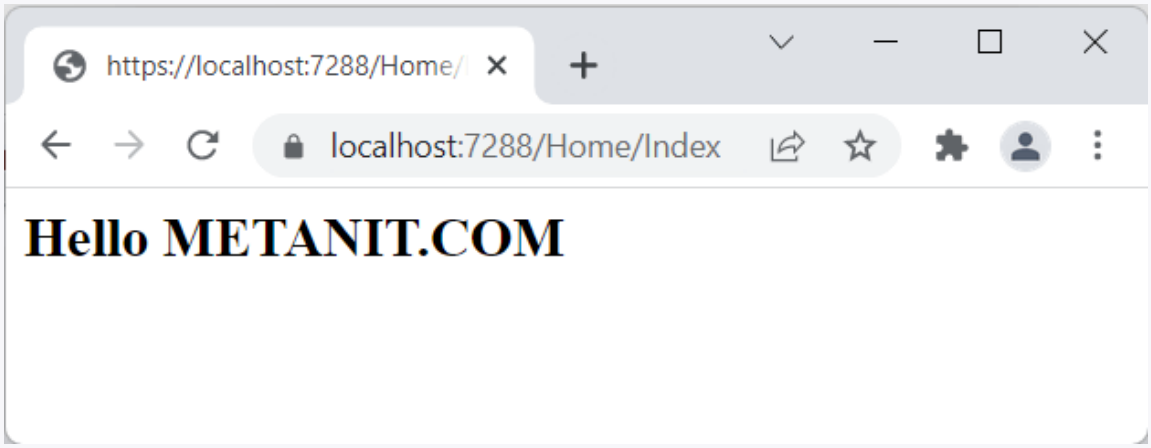
- **Body**: объект Stream, который применяется для отправки данных в ответ пользователю
- **Cookies**: куки, отправляемые в ответе
- **ContentType**: MIME-тип ответа
- **Headers**: коллекция заголовков ответа
- **StatusCode**: статусный код ответа

С помощью объекта Response мы можем отправить настроить параметры ответа и отправить его клиенту. Например, отправим из действия контроллера данные клиенту:

```
using Microsoft.AspNetCore.Mvc;

namespace MvcApp.Controllers
{
    public class HomeController : Controller
    {
        public async Task Index()
        {
            Response.ContentType = "text/html;charset=utf-8";
            await Response.WriteAsync("<h2>Hello METANIT.COM</h2>");
        }
    }
}
```

Здесь в методе Index вначале устанавливается заголовок Content-Type, а затем с помощью метода `WriteAsync()` отправляется некоторое простейшее содержимое в виде строки с кодом html. Поскольку метод `WriteAsync()` - асинхронный, то к нему можно применить оператор await, а метод-действие Index определен как асинхронный. Соответственно при обращении к этому действию мы увидим в браузере это содержимое:




## Request

Свойство **HttpContext.Request** представляет объект **HttpRequest** и предоставляет разнообразную информацию о запросе. Этот же объект

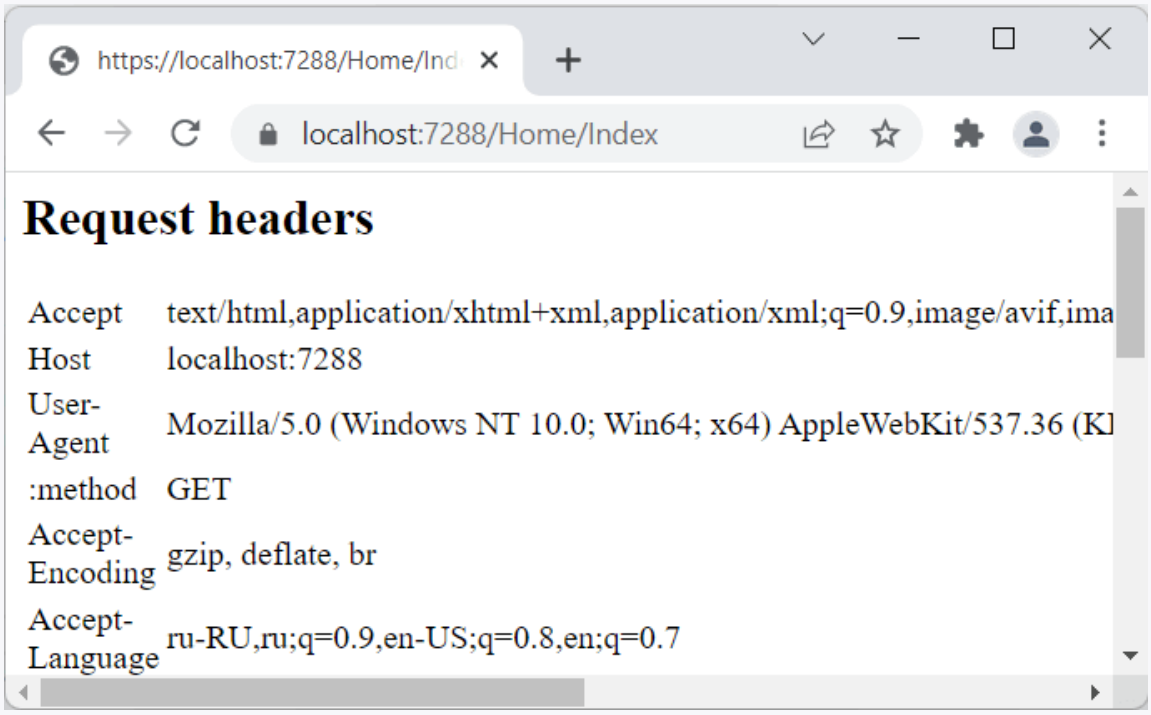
доступен через свойство **Request** класса контроллера. Среди свойств объекта Request можно выделить следующие:

- **Body**: объект Stream, который используетя для чтения данных запроса
- **Cookies**: куки, полученные в запросе
- **Form**: коллекция значений отправленных форм
- **Headers**: коллекция заголовков запроса
- **Path**: возвращает запрошенный путь - строка запроса без домена и порта
- **Query**: возвращает коллекцию переданных через строку запроса параметров
- **QueryString**: возвращает ту часть запроса, которая содержит параметры. Например, в запросе *http://localhost:52682/Home/Index?alt=4* это будет `?alt=4`

Вся основная информация нам доступна из заголовков. Например, получим из запроса все заголовки и выведем их в браузере:

```
using Microsoft.AspNetCore.Mvc;

namespace MvcApp.Controllers
{
    public class HomeController : Controller
    {
        public async Task Index()
        {
            Response.ContentType = "text/html;charset=utf-8";
            System.Text.StringBuilder tableBuilder = new("<h2>Reques
            foreach (var header in Request.Headers)
            {
                tableBuilder.Append($"<tr><td>{header.Key}</td><td>{
            }
            tableBuilder.Append("</table>");
            await Response.WriteAsync(tableBuilder.ToString());
        }
    }
}
```




Назад Содержание Вперед

Помощь сайту

**YooMoney:**
410011174743222

Перевод на карту
**Номер карты:**
4048415020898850

**Номер карты:**
4890494751804113

Вконтакте | Телеграм | Twitter | Канал сайта на youtube | Помощь сайту

Контакты для связи: metanit22@mail.ru