

# Конспект по изученному материалу по ASP.NET WEB API

**Контроллеры** - это по сути классы, которые привязаны к определённому маршруту.

**[Route("/[controller]")]** - эта строчка есть в начале каждого нормального контроллера. Имеется ввиду, что методы этого контроллера будут доступны по пути: "домен/название контроллера"

Но это только путь к самому контроллеру, сам по себе он ничего не даёт, по сути это библиотека методов, которые и используются для общения с сервером (То есть это просто класс по факту, такая же логика, библиотека методов)

**Методы и есть сами запросы.**

Каждый метод должен возвращать ActionResult, то есть результат запроса. Это как ответ на запрос. Дальше идёт тег, который обозначает какой именно запрос был сделан. Пример: public ActionResult Get (Это запрос на получение информации)

Также перед методом который связан с запросом (всё таки бывают и те что не связаны), должен стоять атрибут, который тоже указывает для чего нужен этот метод. В кавычках рядом с атрибутом может быть указан маршрут, который привязан к этому методу. Без маршрута метод будет выполняться всегда в дериктории контроллера. То есть если нет маршрута у метода Get, но в библиотеке был сделан запрос Get, то вызовется именно он. А если есть

маршрут, то тогда надо обратиться и к нему тоже. Вот пример такого атрибута:  
`[HttpGet("getByName")]`

Далее в аргментах метода могут указываться параметры, которые обязательно нужно передать, чтобы получить позитивный ответ на запрос. Это может быть любая переменная либо же экземпляр класса. Если это экземпляр класса, то тогда надо указать все его обязательные свойства (указанные как Required).

Если запрос передаётся в формате json, то в аргументах его нужно заранее расшифровать. Для этого применяется такая запись: **[FromBody] User user**

Также это будет значить что метод ожидает получить конкретно json формат.

**Swagger используется для тестирования всех этих запросов.** По сути swagger отображает все контроллеры, а также их внутренние методы. И благодаря свагеру можно тестировать эти методы в интерактивном режиме. Там есть удобная панелька и ты просто нажимаешь на метод, вводишь аргументы и смотришь результат запроса. Это всё нужно по сути просто для тестирования запросов.

Сама логика запроса, то есть что должно быть после получения запроса и всех присущих ему аргументам, делается уже при помощи кода вручную.

**Логика пишется внутри метода.** Самое главное это в конце вернуть `ActionResult` с ответом. Этот ответ появится у того кто кинул запрос.