**Таблица маршрутов** — это структура данных, которая хранит все маршруты, определенные в приложении. Каждый маршрут может быть представлен в виде ключ-значение, где ключом является URI, а значением — соответствующий обработчик.

**Конечная точка (endpoint)** — это исполняемый кода приложения, обрабатывающий запрос.

Конечные точки определяются в приложении и настраиваются при его запуске. Процесс сопоставления конечных точек может извлекать значения из URL-адреса запроса и предоставлять эти значения для обработки запроса.

Шаблон маршрута, который сопоставляется с конечной точкой, может иметь параметры. Параметры имеют имя и определяются в шаблоне маршрута внутри фигурных скобок: {название\_параметра}

Система маршрутизации сопоставляет маршруты с запросом в том порядке, в котором они определены. Поэтому если мы определяем маршруты, которые могут пересекаться, то вначале надо определять те маршруты, которые имеют больший приоритет или более специфичные. А больше общие или стандартные маршруты должны определяться ближе к концу.

Сопоставление адреса URL или **URL matching** представляет процесс сопоставления запроса с конечной точкой. Данный процесс основывается на пути запроса и полученных в запросе заголовках. Данный процесс проходит ряд этапов:

1. Сначала выбираются все конечные точки, шаблон маршрута которых совпадает с путем запроса
2. Далее из полученного на предыдущем этапе набора конечных точек удаляются те, которые не соответствуют ограничениям маршрута
3. Затем из полученного на предыдущем этапе набора конечных точек удаляются те, которые не удовлетворяют политике объекта MatcherPolicy (вкратце: класс **MatcherPolicy** позволяет определить порядок сравнения конечных точек и адреса URL)
4. И в самом конце применяется объект **EndpointSelector** для выбора из полученного на предыдущем этапе списка конечной точки, которая в конечном счете и будет обрабатывать запрос

Приоритет конечных точек зависит от двух факторов:

* Порядок следования в наборе конечных точек
* Приоритетность шаблона маршрута

Приоритетность шаблонов маршрута зависит от специфичности шаблона. Специфичность шаблона определяется на основе следующих критериев:

* Шаблон маршрута с большим количеством сегментов более специфичен, чем шаблон меньшим количеством сегментов
* Сегмент с текстовым литералом (статический сегмент) более специфичен, чем сегмент с параметром маршрута
* Сегмент с параметром, к которому применяется ограничение маршрута, более специфичен, чем сегмент с параметром без ограничения
* Комплексный сегмент более специфичен, чем сегмент с параметром с ограничением
* Параметр catch-all (параметр, который соответствует неопределенному количеству сегментов) наименее специфичен

Если в конечном счете осталось две и более конечных точек, которые соответствуют запрошенному адресу, и соответственно система маршрутизации не может выбрать, какая из этих конечных точек должна обрабатывать маршрут, то генерируется исключение.

**UseRouting** добавляет соответствие маршрута в конвейер ПО промежуточного слоя. Это ПО промежуточного слоя обращается к набору конечных точек, определенных в приложении, и выбирает наиболее подходящее на основе запроса.

**UseEndpoints** добавляет выполнение конечной точки в конвейер ПО промежуточного слоя. Он запускает делегат, связанный с выбранной конечной точкой.

Фреймворк MVC позволяет использовать в приложении маршрутизацию на основе атрибутов. Такой тип маршрутизации еще называется **Attribute-Based Routing**. Атрибуты предоставляют более гибкий способ определения маршрутов. Маршруты, определенные с помощью атрибутов, имеют приоритет по сравнению с маршрутами, определенными в классе Startup.

В качестве параметра атрибут Route принимает шаблон URL, с которым будет сопоставляться запрошенный адрес.

Если в проекте планируется использовать только маршрутизацию на основе атрибутов, то в классе Startup мы можем не определять никаких маршрутов.

От всех параметров шаблона маршрутов в атрибутах отличаются два параметра controller и action, которые ссылаются соответственно на контроллер и его действие. При использовании их надо помещать в квадратные скобки, а не в фигурные, как другие параметры: [Route("[controller]/[action]/{id?}")].

С помощью атрибутов можно задать несколько маршрутов для одного метода.