**25. Фильтры конечной точки**

**Понятие:**  
Фильтры конечной точки — это механизмы, которые позволяют перехватывать и модифицировать запросы и ответы в ASP.NET Core. Они применяются к маршрутам и предоставляют возможность выполнять различные действия до и после обработки запроса конечной точкой (методом контроллера).

**Принципы действия:**

* **Обработка до выполнения**: Фильтры могут выполнять код перед вызовом метода конечной точки. Это позволяет, например, проверять аутентификацию, авторизацию или выполнение валидации данных.
* **Обработка после выполнения**: Фильтры могут выполнять код после завершения метода конечной точки. Это может быть полезно для логирования, изменения ответа или обработки ошибок.
* **Управление порядком выполнения**: Фильтры могут быть настроены на выполнение в определенном порядке, что позволяет контролировать логику применения фильтров.

**26. Констрейнт маршрута**

**Понятие:**  
Констрейнт маршрута — это условие или правило, которое ограничивает или определяет, какие значения могут соответствовать определенному сегменту маршрута. Это позволяет фильтровать запросы на основе параметров, переданных в URL.

**Принципы действия:**  
Констрейнты применяются к параметрам в маршруте и могут использоваться для проверки, соответствуют ли переданные значения ожиданиям (например, числовые или строковые форматы). Если значение не соответствует констрейнту, маршрут не будет соответствовать запросу.

**27. Известные констрейнты маршрутов и их принципы действия**

* **int**: Ожидает, что параметр будет целым числом. Например, в маршруте {id:int} параметр id должен быть числом.
* **string**: Ожидает, что параметр будет строкой. Это значение по умолчанию, и если не указано, будет проверка на строку.
* **bool**: Ожидает, что параметр будет булевым значением (true или false).
* **datetime**: Ожидает, что параметр будет датой и временем в формате, поддерживаемом .NET.
* **length**: Позволяет задать минимальную и/или максимальную длину строки. Например, {name:length(1,10)} требует, чтобы строка name имела длину от 1 до 10 символов.
* **regex**: Позволяет использовать регулярные выражения для более сложных проверок значений параметров. Например, {id:regex(^\d{3}$)} требует, чтобы id состоял ровно из трех цифр.