

Инверсия управления (IoC) — это шаблон проектирования, в котором поток управления программы инвертируется. IoC обычно используется для того, чтобы отделить компоненты вашего приложения, поменять местами реализации зависимостей, имитировать зависимости и сделать ваше приложение модульным и тестируемым.

ViewModel — это модель представления. Задачей модели представления является отображение данных, необходимых для представления.

👉 **AutoMapper** позволяет быстро производить проецирование одной модели на другую.

Аутентификация — это проверка введенных данных на подлинность. Логика сравнения данных при аутентификации во всех приложениях плюс-минус одинакова: сравнивается введенный логин и пароль с логином и паролем, который хранится в базе данных. И в нашем сервисе аутентификации мы это сделаем.

Авторизация — предоставление определённому лицу или группе лиц прав на выполнение определённых действий.

Если при аутентификации мы определяли, может ли пользователь использовать наш сервис или нет, то при авторизации мы будем проверять, имеет ли пользователь доступ к определенным функциям нашего сервиса.

Фильтры в ASP.NET Core подразделяются на 5 типов. Каждый тип имеет свою характерную особенность:

👉 **Фильтры авторизации.** Фильтры данного типа определяют, авторизован ли пользователь для выполнения запроса.

👉 **Фильтры ресурсов.** Фильтры данного типа запускаются сразу после фильтров авторизации (если таковые присутствуют) и очень полезны для кэширования и повышения производительности.

- 👉 **Фильтры действий.** Фильтры данного типа запускаются непосредственно до и после вызова метода действия.
- 👉 **Фильтры исключений.** Фильтры данного типа занимаются обработкой исключений до заполнения тела ответа.
- 👉 **Фильтры результатов.** Фильтры данного типа запускаются до и после выполнения результатов действий. Они запускаются только в том случае, если метод выполнен успешно.

Middleware — это промежуточное программное обеспечение, выполняющееся в конвейере приложения. Так же, как и в фильтры, в Middleware мы можем поместить логику обработки запроса. Единственное различие между фильтрами и Middleware состоит в том, что Middleware имеет доступ только к данным HttpContext. Фильтры, например, могут получить доступ к данным маршрутизации, а Middleware — нет.

На практике очень часто Middleware в ASP.NET Core используется при реализации аутентификации на основе токенов JWT.