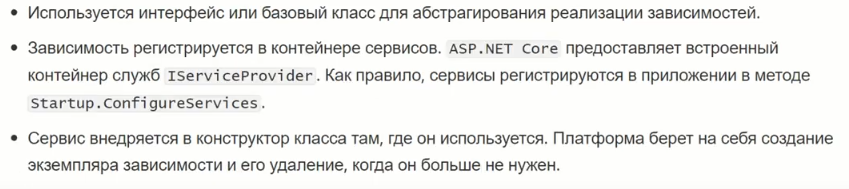
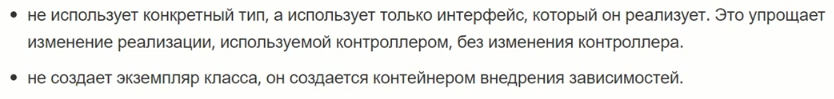
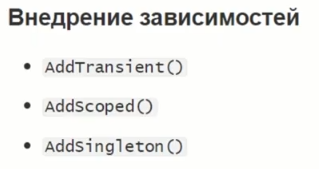


Внедрение зависимостей в свою очередь устраняет эти проблемы следующим образом:

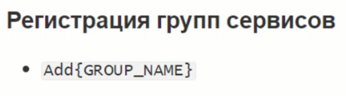


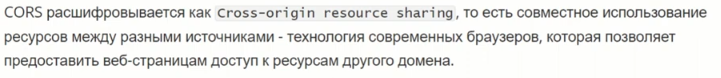


Добавлять зависимости в ASP.Net Core можно с помощью:

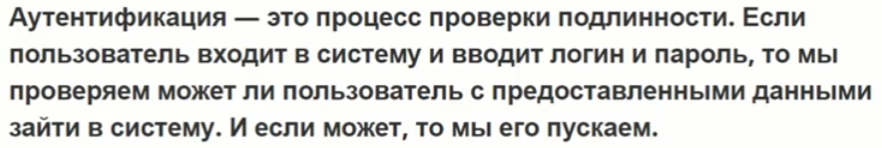


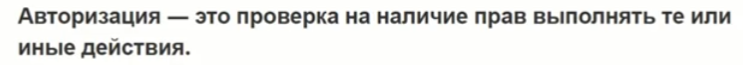




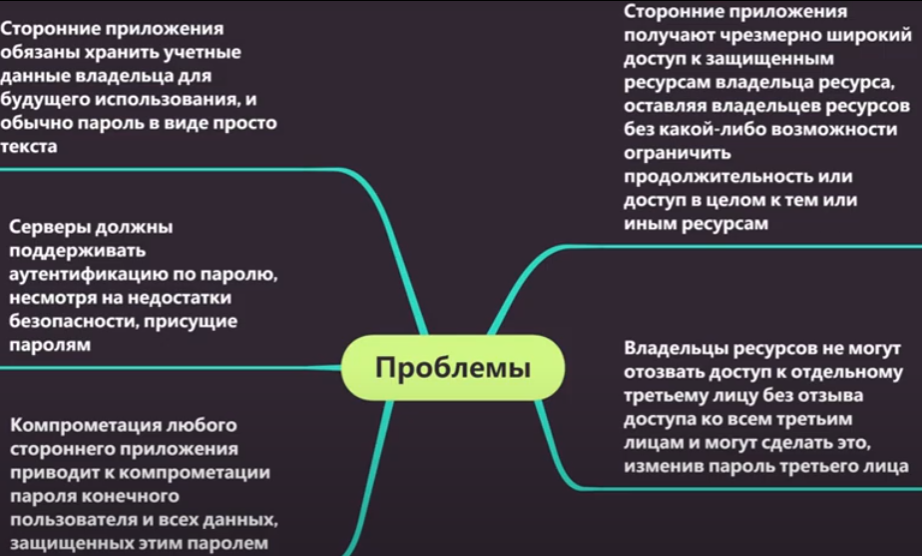


Middleware это фрагмент кода в конвейере приложения, используемый для обработки запросов и ответов. Как правило каждая Middleware обрабатывает входящие запросы и передает выполнение следующему Middleware для дальнейшей обработки. Middleware обычно инкапсулируется в класс и предоставляется с помощью метода расширения.

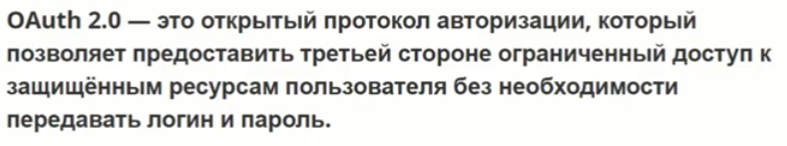


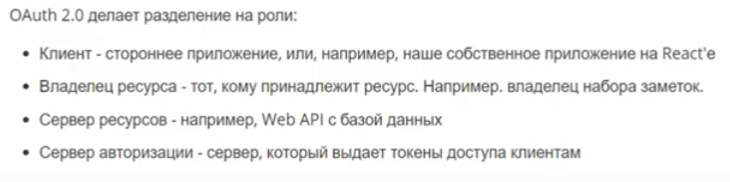


Проблемы авторизации и аутентификации:

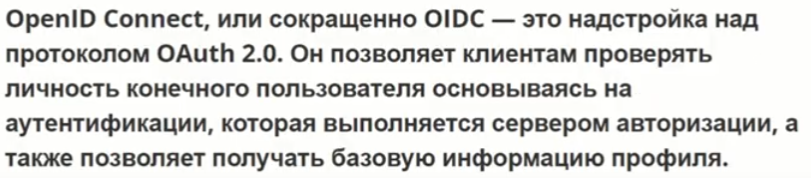


Эти проблемы можно решить путем внесения дополнительного слоя авторизации и отделением ролей клиента от роли владельца ресурса, именно такой подход использует специально разработанный протокол OAuth 2.0





Клиент запрашивает доступ к ресурсам, контролируемым владельцем ресурса и расположенным на сервере ресурсов и доступ предоставляется не по имени пользователя и паролю, а по специальному токену доступа, который выдается каждому пользователю отдельно спец. Сервером авторизации.

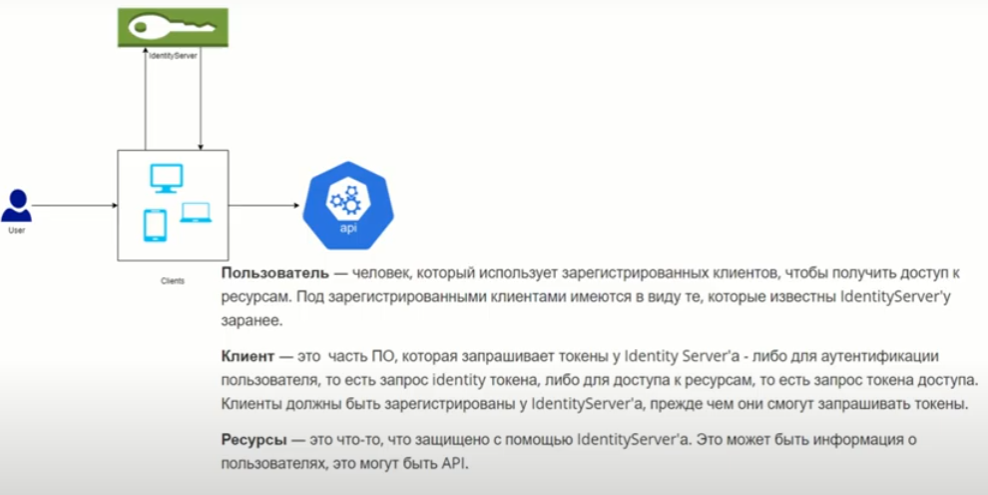


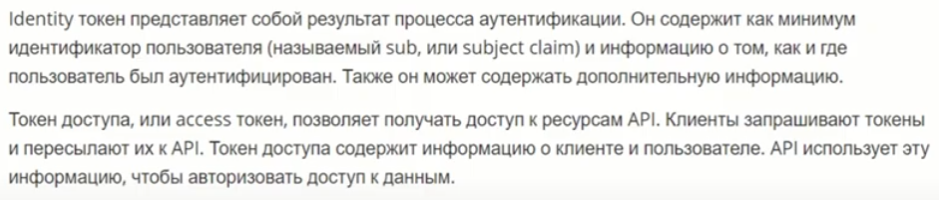
Для работы исп. REST Full API, описанный в спецификации, также в OpenID Connect определены доп. Механизмы для надежного шифрования и цифровой подписи. OIDC позволяет разработчикам аутентифицировать своих пользователей между сайтами и приложениями без необходимости иметь пароли ко всем из них, благодаря открытому стандарту OpenID, который и предоставляет пользователю возможность создания единой учетной записи для аутентификации на множестве не связанных между собой интернет ресурсов, используя услуги третьих лиц.



IdentityServer позволяет нам возможности OAuth 2.0 и OIDC. Пользователь, используя любой поддерживаемый клиент, входит в приложение, которое запрашивает токен доступа у сервера авторизации. Сервер авторизации возвращает токен доступа клиенту, и уже с ним клиент идет к серверу ресурсов.

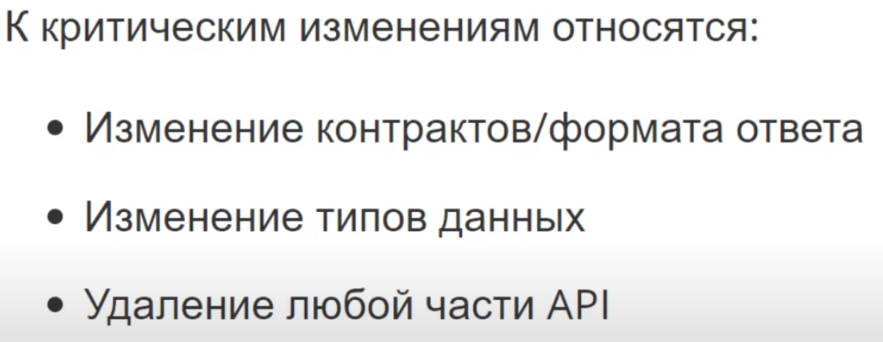
Итог:





Версионирование.

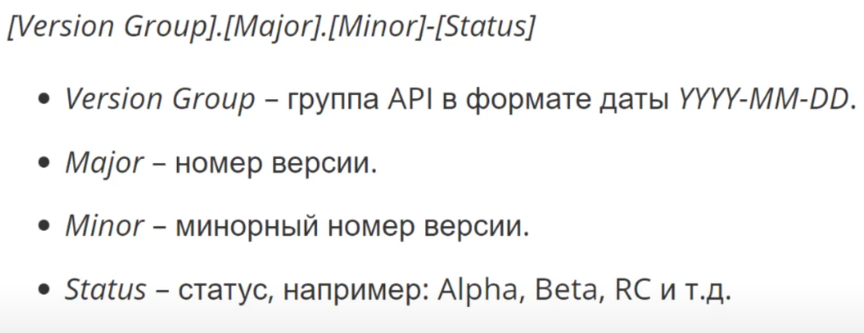
Версию Api необходимо обновлять после внесения критических изменений:



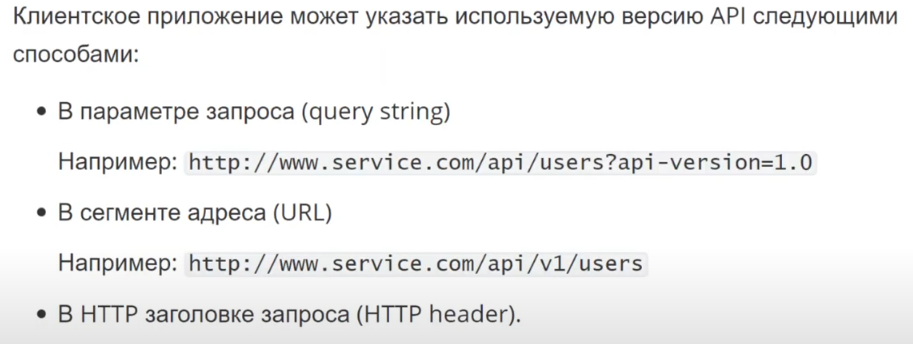
При разработке Api веб приложения часто возникает вопрос создания и поддержки нескольких версий Api. Один из самых простых способов сделать это, использовать библиотеку:



Рассматриваемая библиотека поддерживает следующий формат номера версий:



Каждая часть опциональная, т.е. самый короткий номер версии может состоять только из любой одной части



Кроме того библиотека позволяет разработчику реализовать свой вариант номера версии Api. По умолчанию значение ожидается в параметре запроса с именем api-version, или в сегменте Url. Для второго варианта потребуется доп. Конфигурация контроллера.