**КОНКУРСНОЕ ЗАДАНИЕ**

**Цифровые возможности для бизнеса**

**(2-7 декабря 2020, Межвузовский чемпионат)**

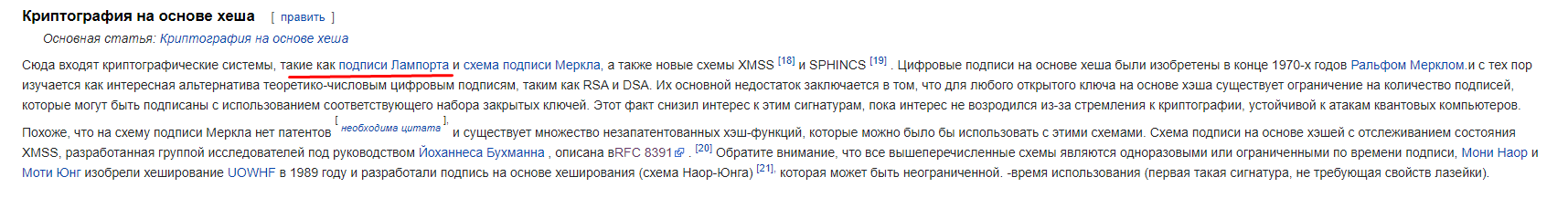
**3 ссесия**

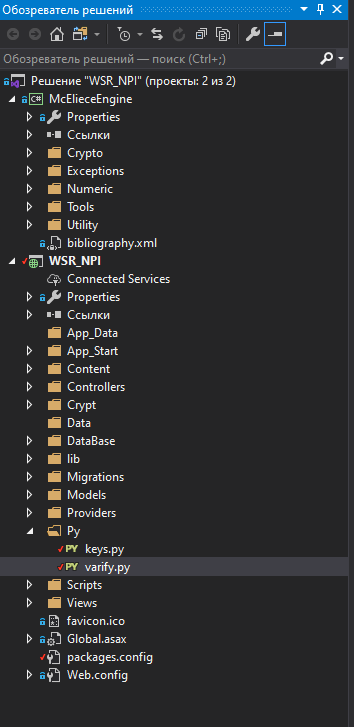
**Модуль 3. Информ. Безопасность**

**Общая идея:** улучшение итогового продукта с точки зрения безопасности. Предлагается рад магистральных сценариев атак с возможным путем решения проблемы, а также несколько дополнительных.

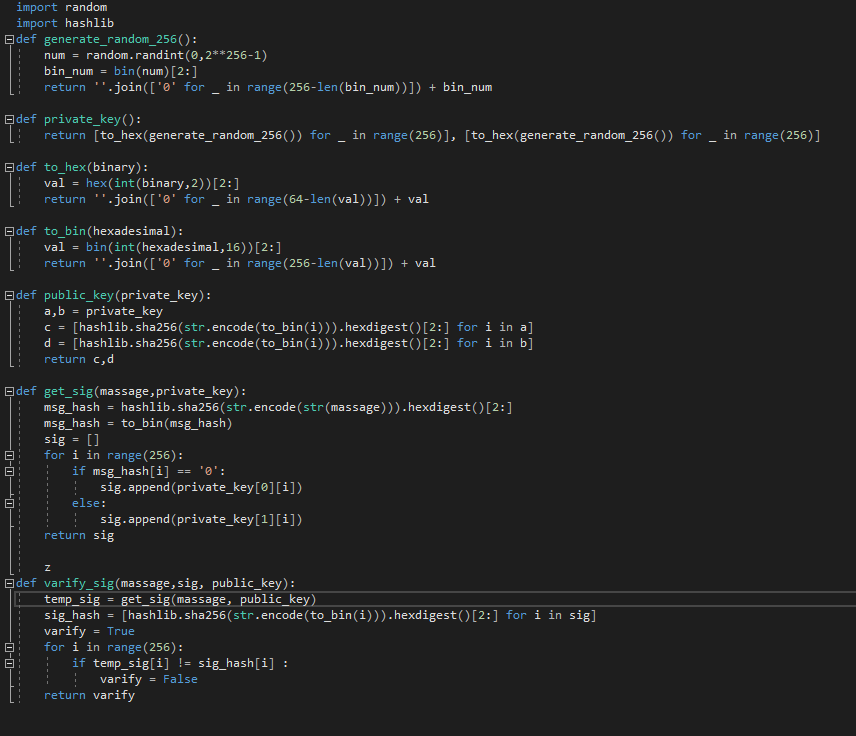
1. **Задание**: необходимо обеспечить механизм, который позволит предотвратить возможность отправки заказа в блокчейн стороннему лицу. Реализовано с помощью интеграции ЭЦП, благодаря чему менеджер может использовать ЭЦП для подписи каждого заказа
2. **Дополнительная задача:** необходимо улучшить механизм, реализованный в прошлой задаче, таким образом, чтобы исключить возможность компрометации блокчейн-хранилища посредством квантовых атак. Реализовано с помощью интеграции пост-квантового алгоритма подписи Лампорта.

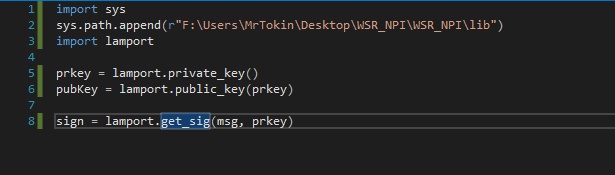
Для реализации взяли – подпись Лампорта, которая является пост-квантовым алгоритмом подписи согласно википедии https://en.wikipedia.org/wiki/Post-quantum\_cryptography#Hash-based\_cryptography

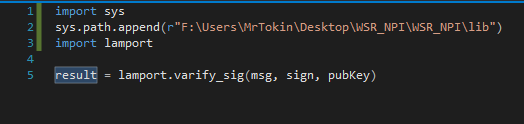




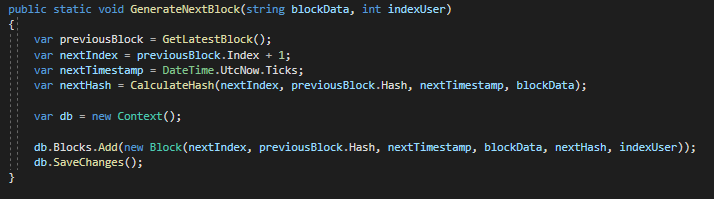
Код подписи алгоритмом Лампорта



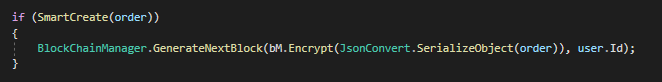




Раньше код для записи блока в блокчейн был следующим:



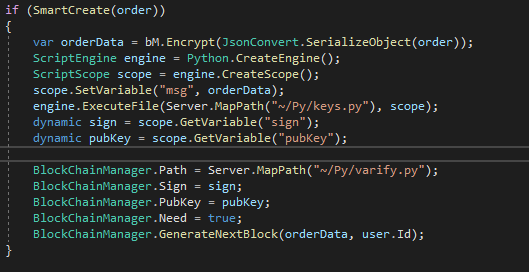
А код для записи информации о заказе в блок следующим:

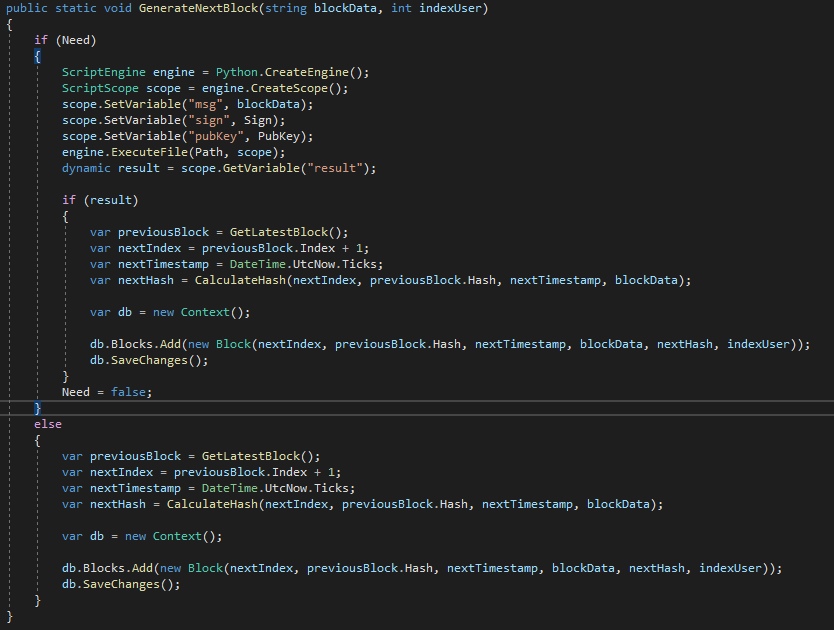


После реализации подписи Лампорта:

код для записи информации о заказе в блок следующим:

Мы видим, что генерируются ключи и подпись



код для записи блока в блокчейн :

Теперь перед записью блока в блокчей – проверяется цифровая подпись Лампорта и в случае успеха – блок помещается в блокчейн.

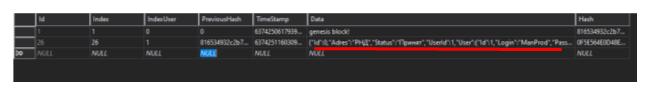
1. **Задача:** необходимо обеспечить механизм, который позволит предотвратить возможность изменения статуса заказа стороннему лицу. Реализована авторизация.

Данный функционал был реализован в 1 сессии, у нас в коде контроллеров, которые отображают страницы – стоят атрибуты [Authorize] – данные атрибуты на стороне сервера – проверяют от кого поступил запрос и если данный пользователь не прошел авторизацию, то страницы не будут отображены!

* **Дополнительная задача:** необходимо улучшить механизм, реализованный в прошлой задаче, таким образом, чтобы исключить возможность его компрометации посредством квантовых атак. Реализована с помощью использования пост-квантового алгоритма шифрования McEliece

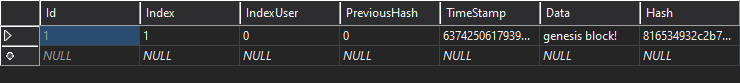
Использовался алгоритм шифрования McEliece

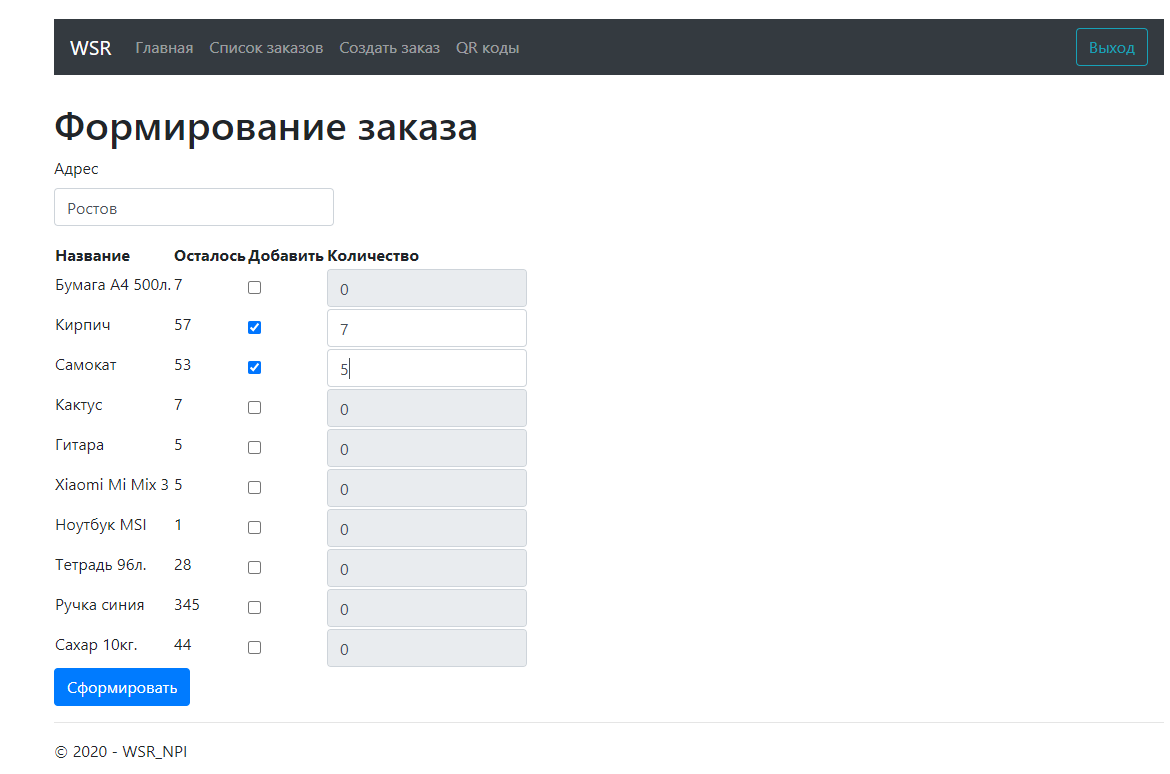
Раньше информация о заказе в блокчейне хранилась в открытом виде:

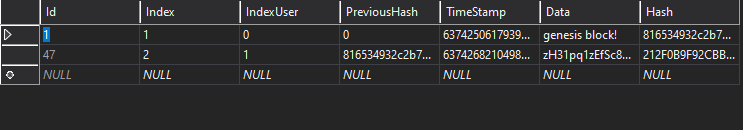


Теперь реализация с использование алгоритма шифрования заказа:

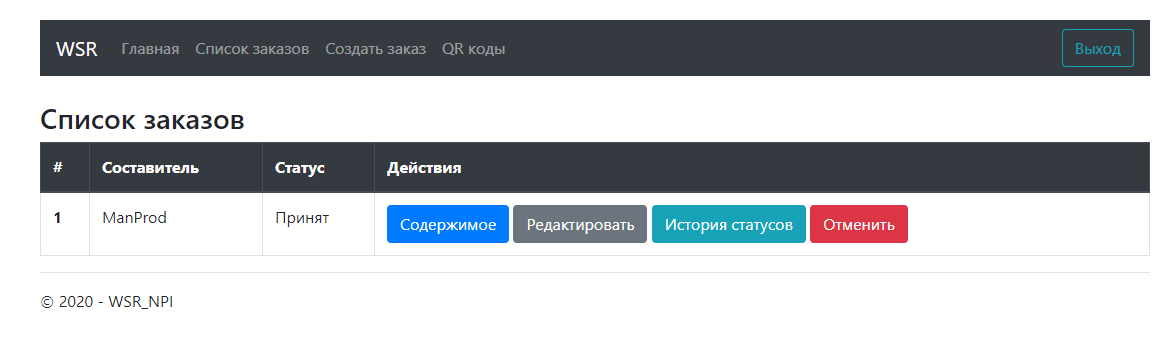
Пустой блокчейн



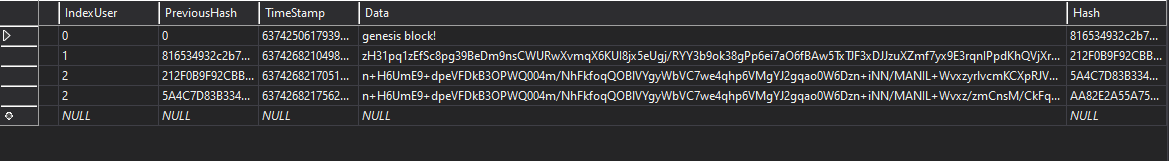


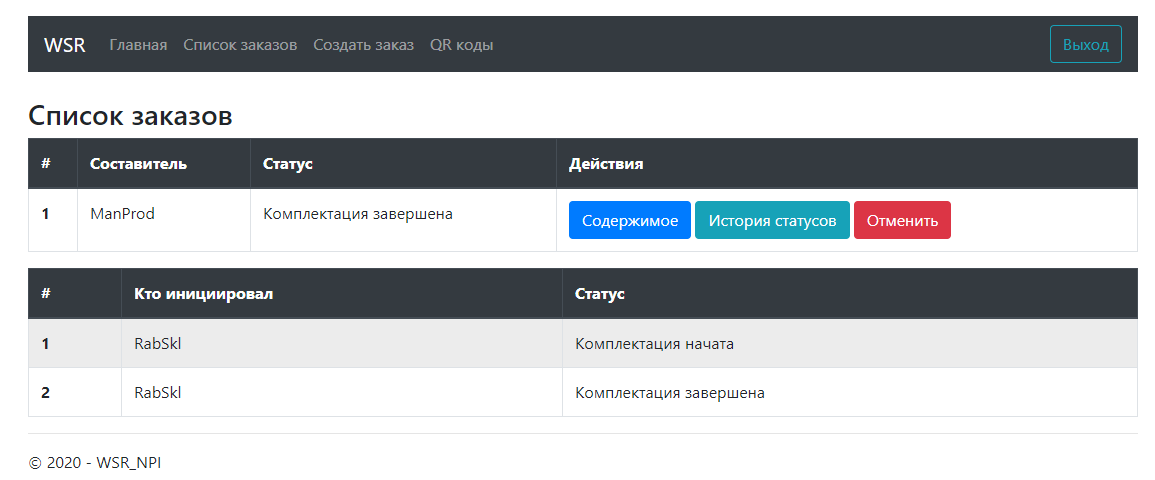


Как мы видим теперь информация о заказе и его статусах хранится в зашифрованном виде. Для более удобного хранения результат шифрования преобразуется в base64.



Далее проводим 2 операции смены статуса



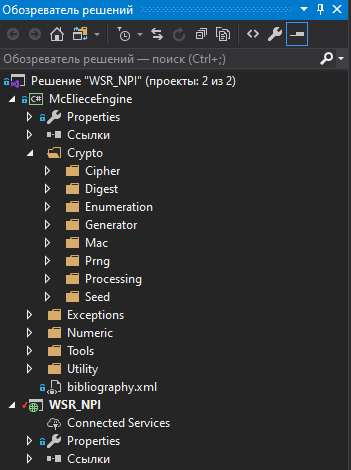


Получаем информацию о смене статусов в расшифрованном виде

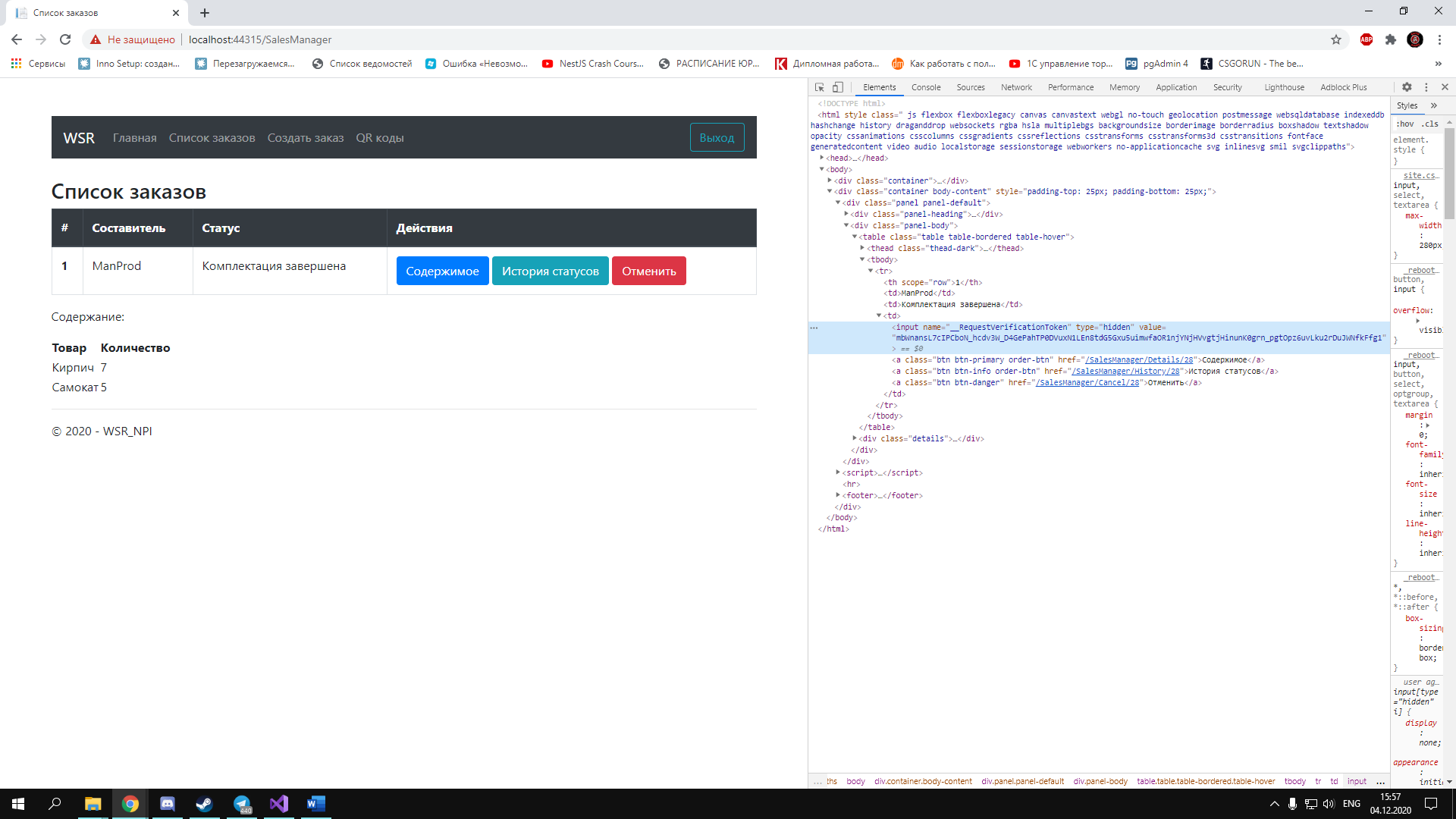
Код менеджера шифрования



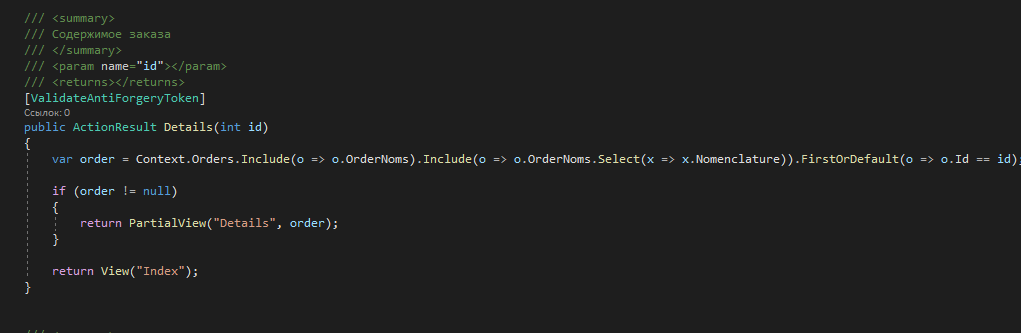
Данный класс использует следующий модуль шифрования:



1. **Задача:** необходимо обеспечить механизм, который позволит предотвратить возможность просмотра информации о заказе стороннему лицу. Реализована с помощью использования токена для доступа к информации о заказе. Функционал проиллюстрирован на рисунках.



На html страницы было добавлено скрытое поле с токеном авторизации, через dom дерево страницы в браузере мы можем видеть это поле с его значением



Теперь при вызове метода – отображения детали о заказе – стоит атрибут ValidateAntiForgeryToken  
Он проверяет токен со страницы, если кто-то попытается получить данные без токена, то страница не пройдет валидацию и данные не будут получены!

При обращении с мобильной версии используется работа с токеном, которая подразумевает работу с доступом по токену, который хранится в бд.

