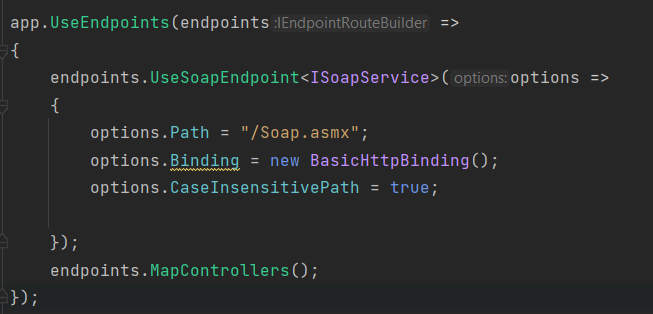
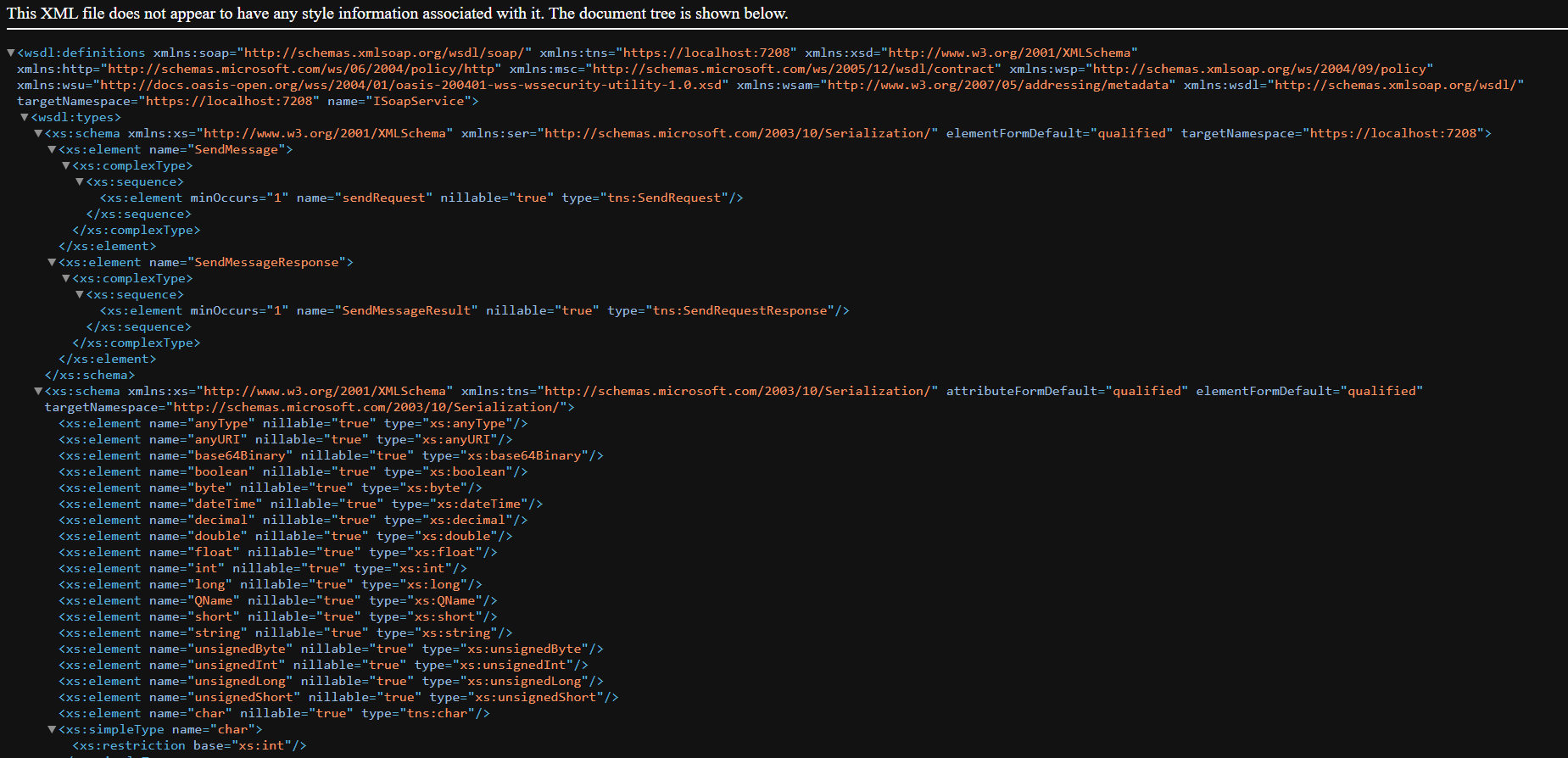
2ая SOAP

Передача данных ведется через .xnl файлы, в остальном та же хуйня, что и REST обычный.

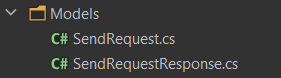
В program.cs билдим проект.

Тут у нас мы указываем, какой адрес будет слушать наш SOAP 

Если перейдем по ссылке «http://localhost:5084/Soap.asmx», то увидим это: 

Тут дофига всего, но от сюда можно наверное что-то достать)

В моделях наши запросы/ответы как их использовать решительно не понятно, но оно есть (если попытаться их отправить в клиенте, то также ловишь ошибку), но наверное, надо как-то через них, благо я их хотя бы написал.

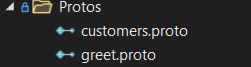


В файле SoapController наш клиент, он в запросе обращается на дефолтный адрес и пытается что-то достать. (НИХУЯ ОН НЕ ДОСТАЕТ)



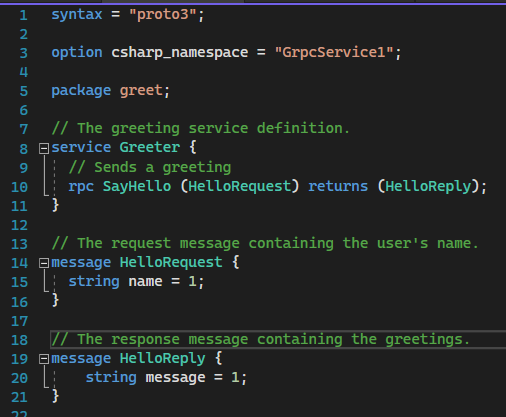
3ая gRPC

Суть простая: все данные для обмена находятся в контрактах описанных в .proto файлах

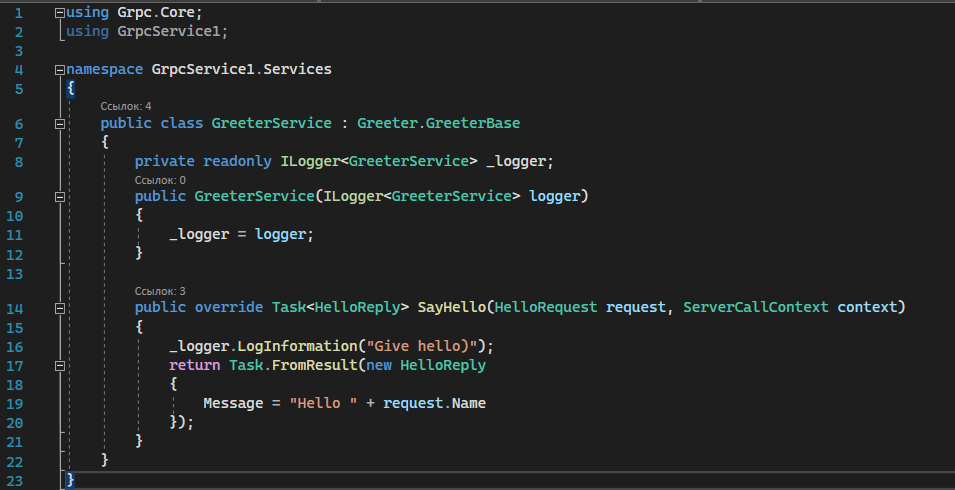


В них описана модель принимаемая(SmthRequest) и передаваемая(SmthReply), и реакция на взаимодействие (service SomeService).

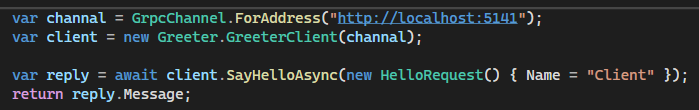
В сервисе мы указываем, что получаем и что отдадим.



После создания контракта ребилдим проект и генерится несколько доп файлов. (В нашем случаем Greeter). В нем есть метод, который нам нужно переопределить (тут SayHello) мы его название указали в .proto файле.



На клиенте все работает также, только теперь мы создаем не сервис и метод для ответа, а клиента, который будет отправлять запрос на сервис.



Создаем канал, в нем указываем путь нашего сервака

Потом создаем клиента (перед этим нужно еще проект ребилднуть, а то файлы не сгенерируются!!!!)

Ну и отправляем запрос. В reply наш ответ с сервера.

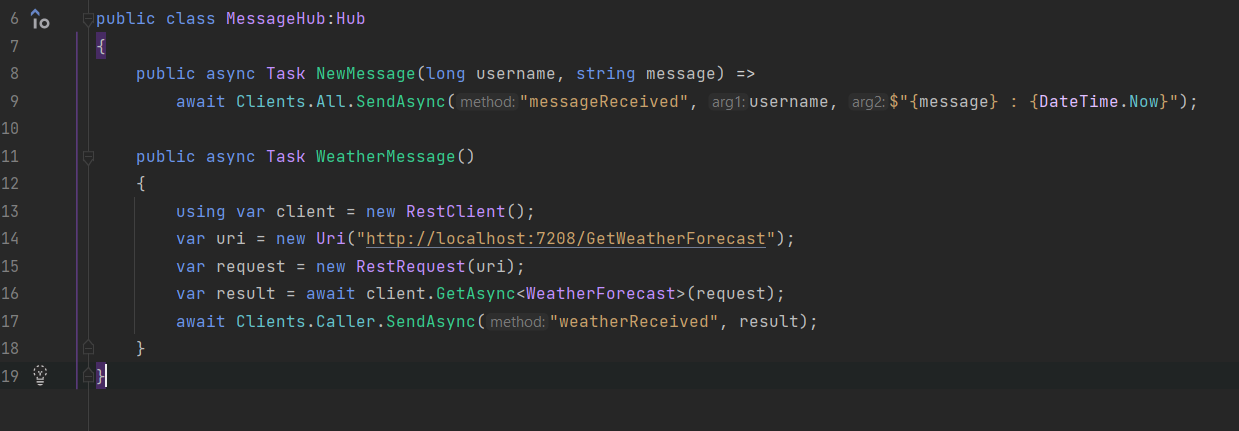
4ая SignalR

SignalR -либа для веб сокетов. Веб сокет - это штука для постоянного обмена сообщениями.

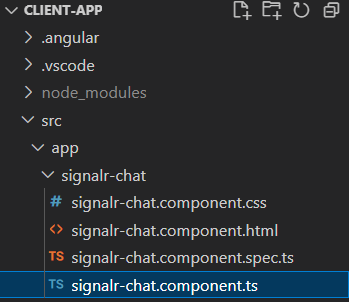
На сервере мы выделяем отдельный адрес, который будет слушать наш веб сокет. В данном случае MessageHub.



Тут описан наш сокет серверный. В первом методе мы просто говорим от кого приняли сообщение и отправляем обратно всем подписанным на этот адрес. Во втором методе он просто обращается в другой контроллер и колит его метод и отправляет результат том, кто к нам обратился.



В клиенте (Client/client-app) есть скрипт, который подписывается на сокет и выводит инфу(на самом деле нихуя он не выводит)



Там ничего интересного нет, просто открой и покажи корсуну, он блюванет, сделает умный вид и зачтет лабу.