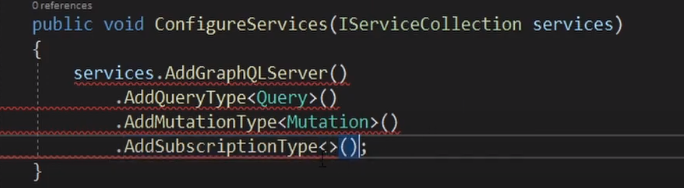
# Query, mutation, subscription

Каждый надо зарегать в сервисах



# Subscriptions

App.UseRouting()

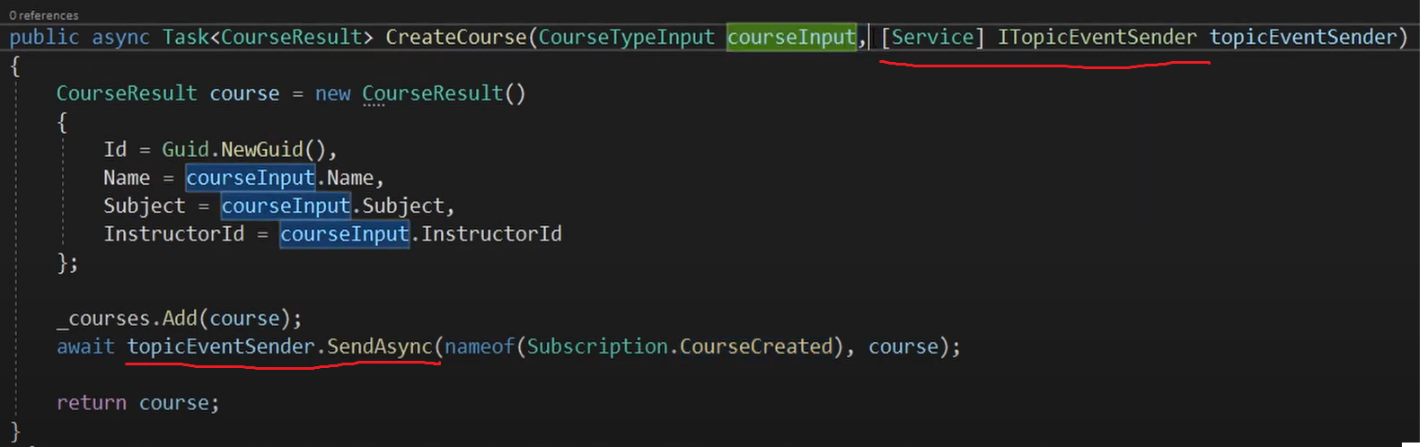
**App.UseWebSockets()**

App.UseEndpoints(ep =>

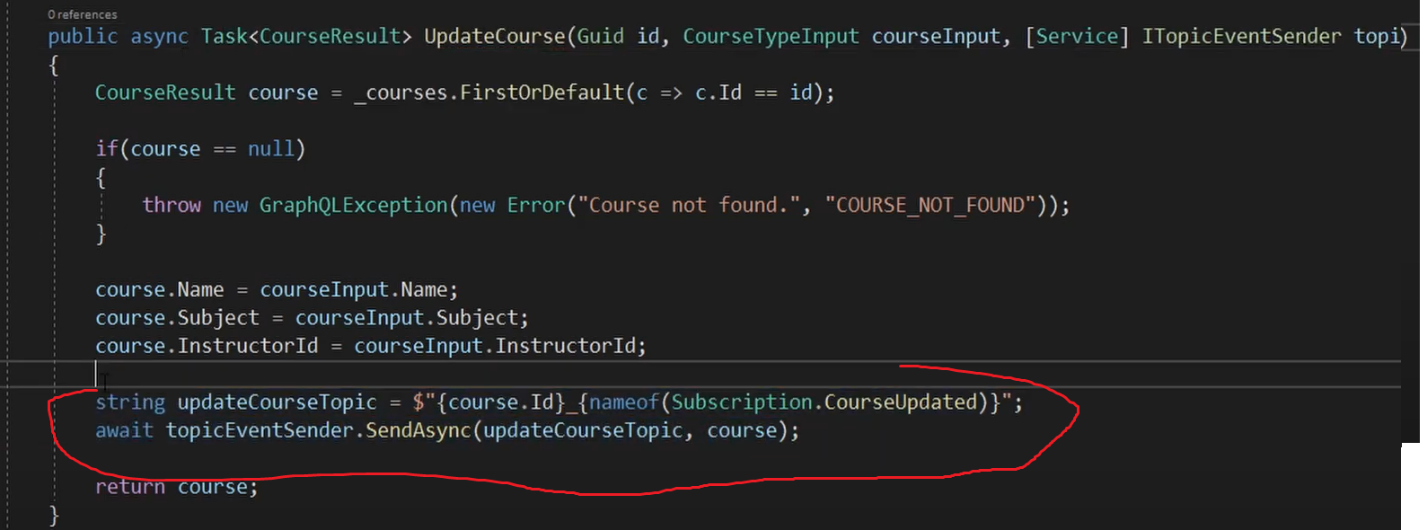
Ep.MapGraphQL();

}

eventSender можно передать как в класс, так и в метод атрибутом [Service]

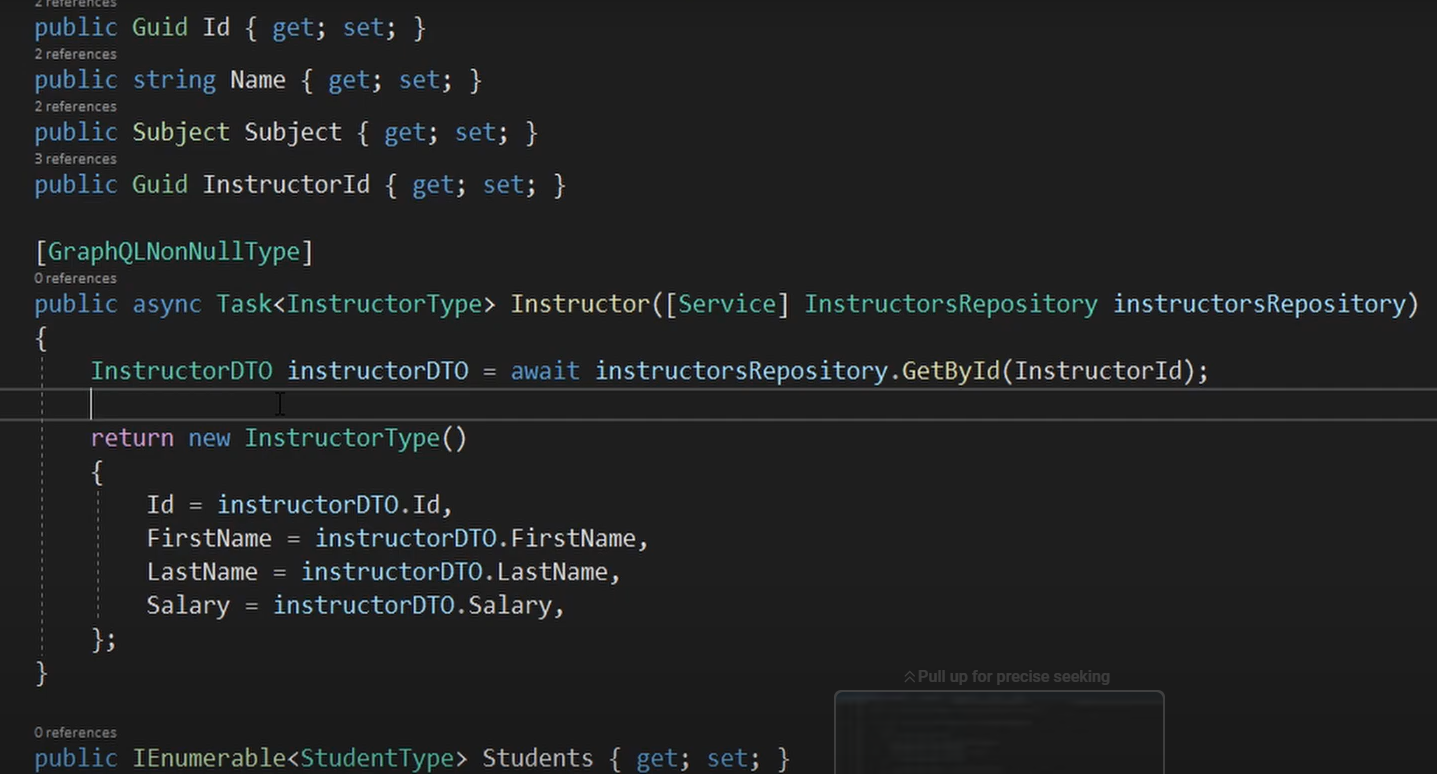


Публикация ивента в конкретный топик

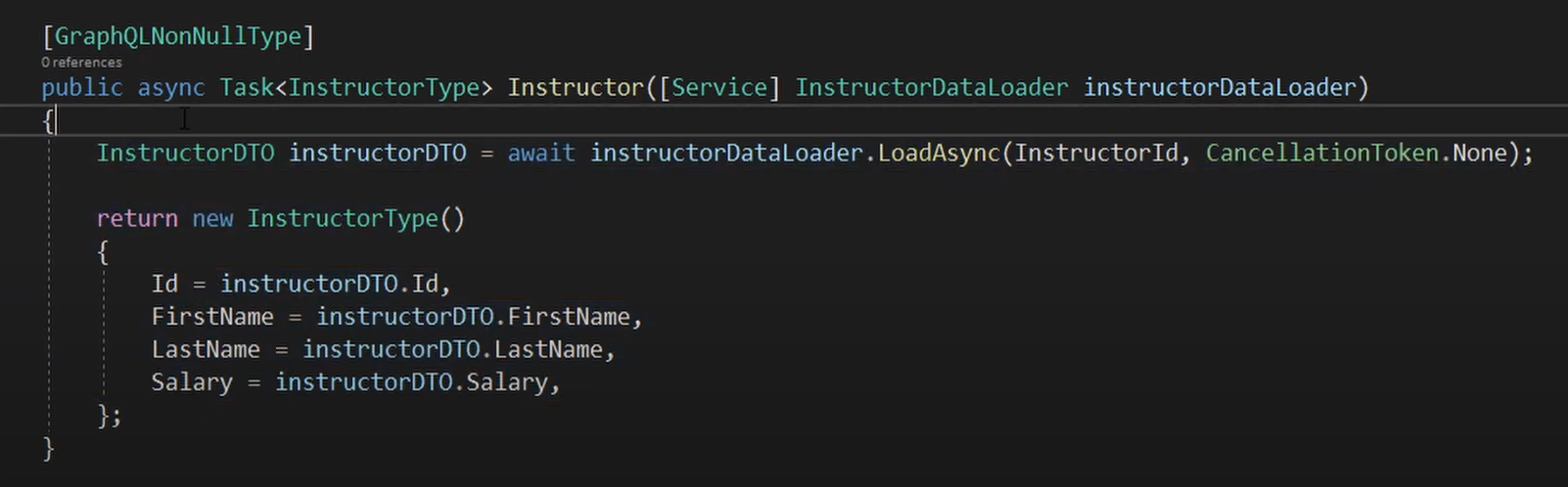


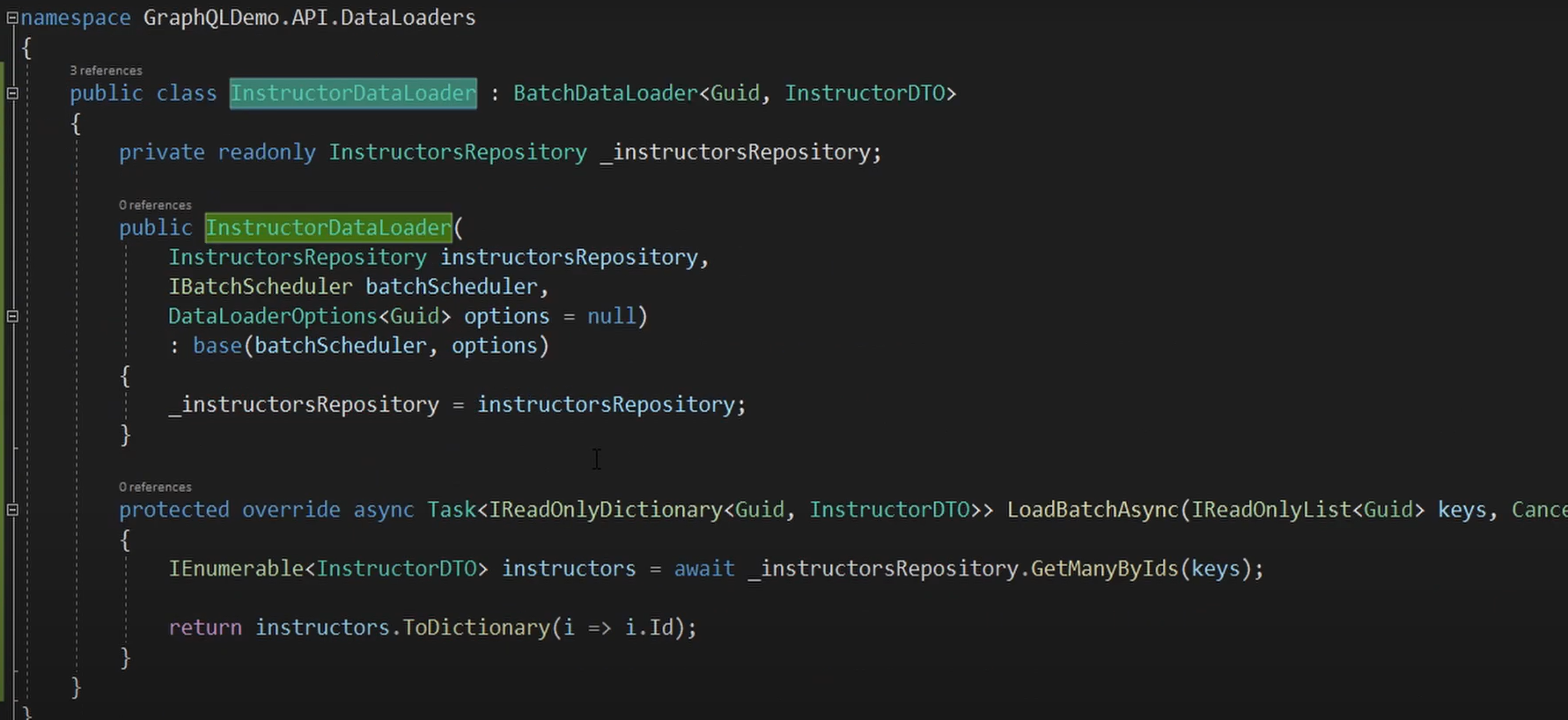
Реализация опционального Include

Но есть N+1 проблема – будет выполнен 1 запрос для списка курсов и для каждого курса по запросу для получения инструктора.

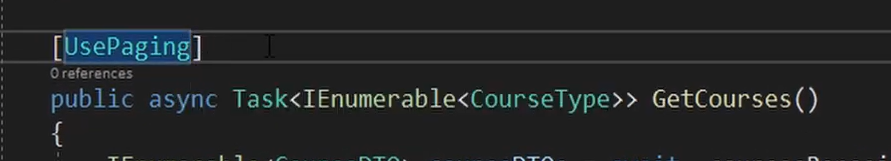


Чтобы решить N+1 нужно добавить DataLoader





Пагинация



В запросе появляется

Edges{

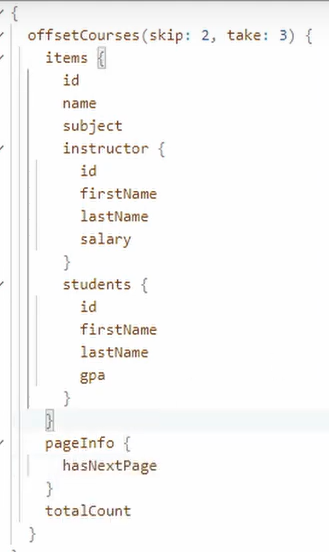
Nodes{

Запрос

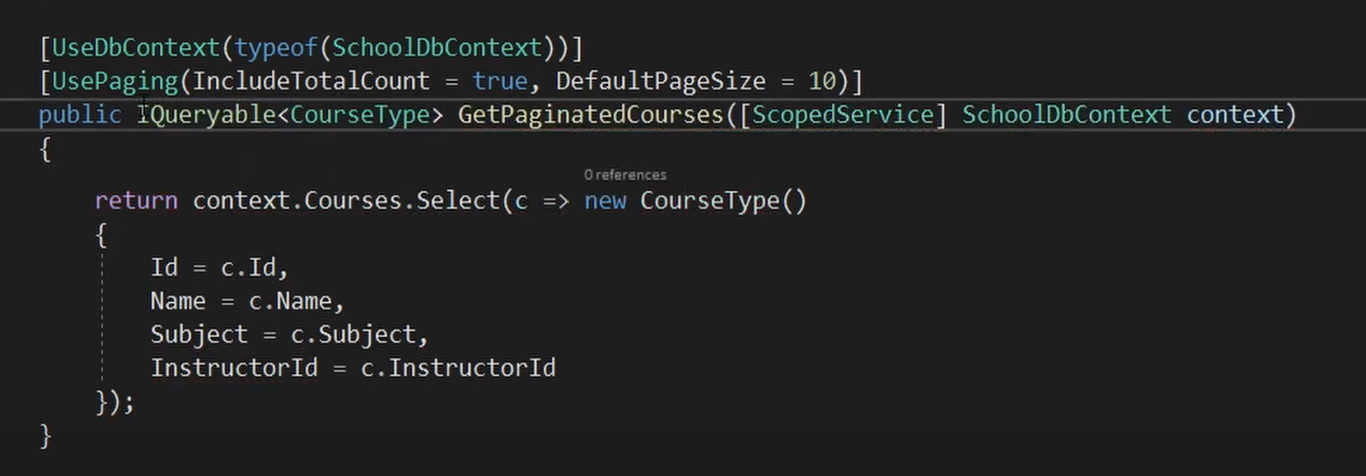
}

}

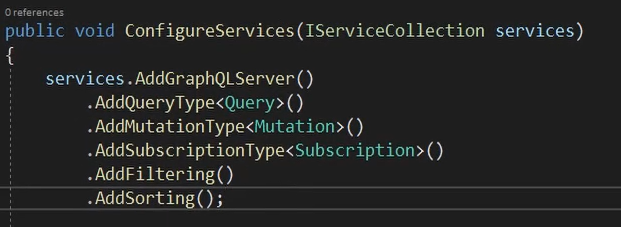
UsePaging предпочтительнее, но есть UseOffsetPaging.

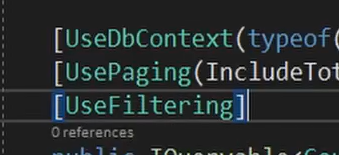


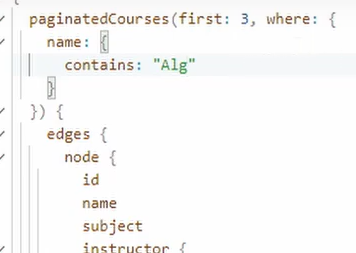
В таком случае \/ вернется IQueriable, и атрибут UsePagination сам выполнить выборку



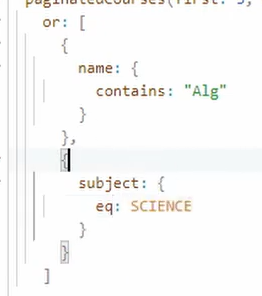
Фильтрация



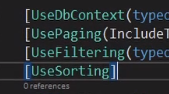


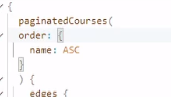


Or – аналоги свича



Сортировка





Projections

Проекции используются чтобы GrapQL сам сделал джоины.

Применяются проекции к методам, которые возвращают IQueriable.