

## Лабораторная работа по теме «Серверная шина предприятия»

Для выполнения лабораторной работы вам потребуется скачать пакет NServiceBus из nuget-репозитория.

1. Разрабатываемая система должна состоять из четырех основных частей: Client, Server, Subscriber1 и Subscriber2.

2. В качестве приложения Client необходимо создать MVC-проект по управлению сущностью по вариантам. Должны быть доступны CRUD-операции над моделью. Приложение может считывать данные напрямую из подсистемы DAL, но создавать, редактировать и удалять сущность должно только через серверную шину данных.

3. Приложение Server должно отправлять пересылаемые по шине данные, и перенаправлять их к приложениям-подписчикам. Пересылаемую информацию необходимо логировать и сохранять в лог-файлах. Для этого можно создать собственный логер или использовать готовое решение, например, NLog.

4. Приложение Subscriber1 должно получать данные от приложения Server и сохранять изменения в базе данных, путем вызова необходимых методов из подсистемы DAL.

5. Приложение Subscriber2 должно получать данные от приложения Server и отправлять информацию на email администратору системы.

6. Сборка DAL (Data Access Layer) инкапсулирует логику работы с БД. Сборка должна одновременно использоваться приложениями Client (для считывания данных) и Subscriber1 (для создания, обновления и удаления данных). Для взаимодействия с БД используйте библиотеку Entity Framework 6 (подход Code First) и паттерны Repository и UnitOfWork.

7. Всю общую логику, которая необходима для работы нескольким приложениям, следует вынести в отдельные модули (сборки).

8. Все проекты должны находится в одном решении.

9. При разработке системы используйте локальный MSSQL сервер. Таблицу (или таблицы) для хранения данных необходимо назвать с использованием следующего формата {ФИ\_TableName}, где ФИ – первые буквы имени и фамилии на английском языке.

10. После проверки работоспособности системы и тестирования следует заменить локальный MSSQL сервер на облачный.

11. Разверните разработанную систему на веб-сервере IIS.

Полезные ссылки:

<https://docs.particular.net/>

<https://docs.particular.net/nservicebus/>

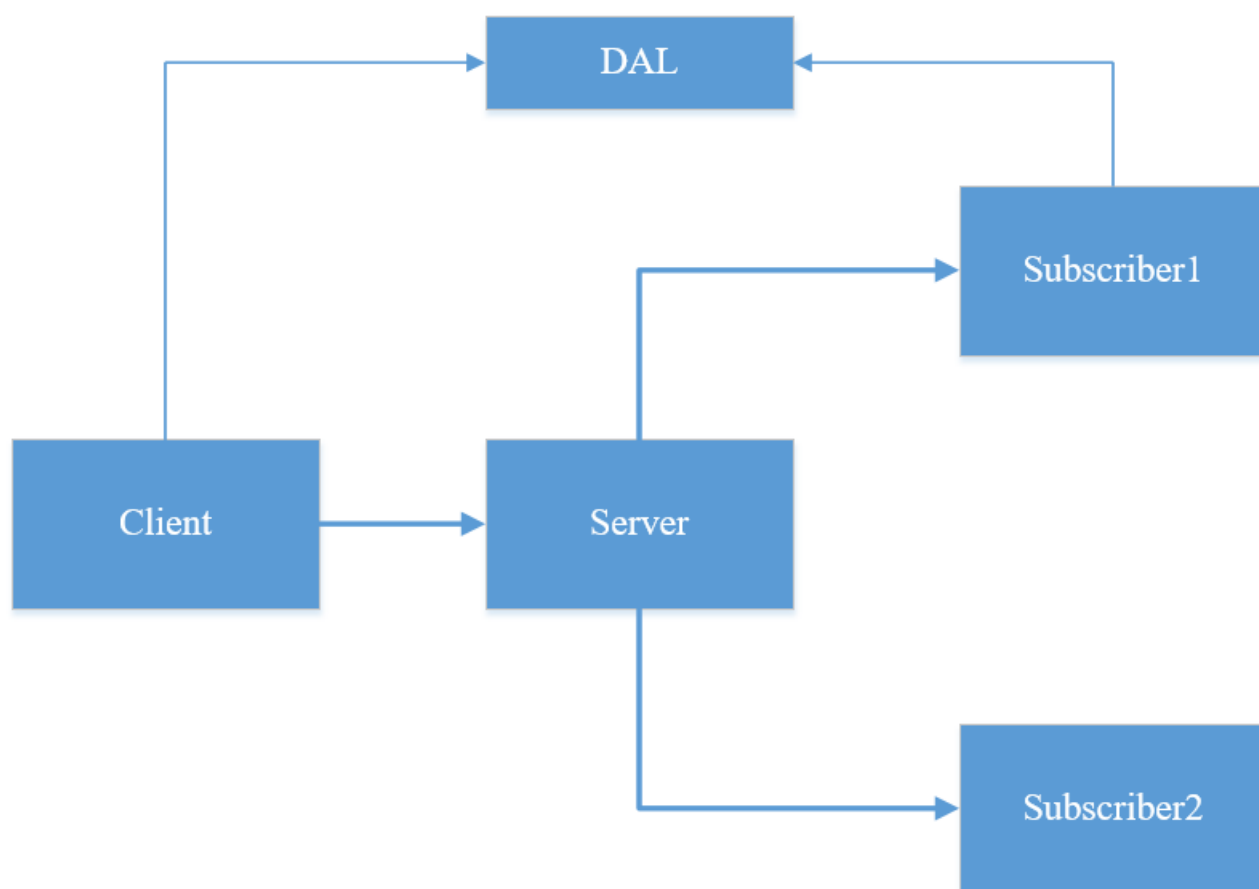
<https://docs.particular.net/samples/>

<https://github.com/NLog/NLog>

[https://msdn.microsoft.com/library/swas0fwc\(v=vs.110\).aspx](https://msdn.microsoft.com/library/swas0fwc(v=vs.110).aspx)

<https://www.mailinator.com/>

Схема взаимодействия между частями системы:



Вариант	Задание
1, 8	Класс «Рабочий». Возможные поля: ФИО, возраст, специальность, стаж, зарплата, пол, место работы и др.
2, 9	Класс «Счет». Возможные поля: номер, тип вклада, PIN, баланс, дата создания счета, пользователь, история изменений счета (список операций пополнения/снятия денег со счета), и т.д.
3, 10	Класс «Дисциплина». Возможные поля: название, количество лекций, количество лабораторных, наличие курсового проекта, вид контроля, количество слушателей, лектор и т.д.
4, 11	Класс «Книга». Возможные поля: название, область науки, количество страниц, издательство, тип переплета, наличие CD, DVD, автор и т.д.
5, 12	Класс «Растение». Возможные поля: название, описание, вид, возраст, класс, размер соцветия, класс опасности, ареал произрастания, область применения

	(в пищевой промышленности, в получении лекарственных препаратов, в сельском хозяйстве...), первооткрыватель (ученый) и т.д.
6, 13, 15	Класс «Компьютер». Возможные поля: количество ядер процессора, тип процессора, частота, наличие технологии Hyper-Threading, разрядность архитектуры, производитель, видеокарта, размер и тип ОЗУ, размер и тип жесткого диска, и т.д.
7, 14, 16	Основной объект «Самолет». Возможные поля: номер, класс, авиакомпания, члены экипажа (список), количество мест, рейсы, год выпуска, производитель и т.п. Наследуемые классы: военный самолет, гражданский самолет.