

# Тестовое задание

Необходимо разработать сервис заказов. Сервис должен позволять управлять заказами и сохранять свое состояние в базу данных. Необходимо реализовать следующий функционал:

- Создание заказа
- Изменение / редактирование заказа
- Удаление заказа
- Получение заказа по идентификатору

Заказ представляем собой комбинацию шапки заказа и строк заказа. Шапка состоит из следующих полей:

- идентификатор заказа, GUID
- статуса заказа, пречисление: новый, ожидает оплату, оплачен, передан в доставку, доставлен, завершен
- дата создания, DateTime

Строка заказа представляет собой идентификатор товара (GUID) и количество (целое не отрицательное число).

Пример в формате json:

```
{
  id: "ddfea950-d878-4bfe-a5d7-e9771e830cbd",
  status: "New",
  created: "2023-01-01 23:21.12",
  lines: [
    { id: "ffee5ad4-c7e5-48d3-ae4-b1593a4e1863", qty: 12},
    { id: "15c68982-ed2e-4460-a4c3-66804d9aed65", qty: 1},
  ]
}
```

При реализации сервиса необходимо использовать следующие технологии:

- asp.net web api (<https://learn.microsoft.com/en-gb/aspnet/core/fundamentals/apis?view=aspnetcore-7.0>)
- asp.net dependency injection framework (<https://learn.microsoft.com/en-gb/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-7.0>)

- Entity Framework (<https://learn.microsoft.com/en-gb/aspnet/core/data/entity-framework-6?view=aspnetcore-7.0>)
- postgresql и npgsql(<https://www.postgresql.org> & <https://www.npgsql.org>)

Сервис должен реализовывать следующий контракт:

```

Создание заказа
POST /orders
{
  id: "ddfea950-d878-4bfe-a5d7-e9771e830cbd",
  lines: [
    { id: "fffee5ad4-c7e5-48d3-ae4-b1593a4e1863", qty: 12},
    { id: "15c68982-ed2e-4460-a4c3-66804d9aed65", qty: 1},
  ]
}
====>
{
  id: "ddfea950-d878-4bfe-a5d7-e9771e830cbd",
  status: "New",
  created: "2023-01-01 23:21.12",
  lines: [
    { id: "fffee5ad4-c7e5-48d3-ae4-b1593a4e1863", qty: 12},
    { id: "15c68982-ed2e-4460-a4c3-66804d9aed65", qty: 1},
  ]
}

-----

Изменение / редактирование заказа
PUT /orders/ddfea950-d878-4bfe-a5d7-e9771e830cbd
{
  status: "Paid",
  lines: [
    { id: "fffee5ad4-c7e5-48d3-ae4-b1593a4e1863", qty: 12},
    { id: "15c68982-ed2e-4460-a4c3-66804d9aed65", qty: 1},
  ]
}
====>
{
  id: "ddfea950-d878-4bfe-a5d7-e9771e830cbd",
  status: "Paid",
  created: "2023-01-01 23:21.12",
  lines: [
    { id: "fffee5ad4-c7e5-48d3-ae4-b1593a4e1863", qty: 12},
    { id: "15c68982-ed2e-4460-a4c3-66804d9aed65", qty: 1},
  ]
}

-----

Удаление заказа
DELETE /orders/ddfea950-d878-4bfe-a5d7-e9771e830cbd
==>
OK, 200

```

```
-----  
  
Получение заказа по идентификатору  
GET /orders/ddfea950-d878-4bfe-a5d7-e9771e830cbd  
==>  
{  
  id: "ddfea950-d878-4bfe-a5d7-e9771e830cbd",  
  status: "Paid",  
  created: "2023-01-01 23:21.12",  
  lines: [  
    { id: "ffee5ad4-c7e5-48d3-ae4-b1593a4e1863", qty: 12},  
    { id: "15c68982-ed2e-4460-a4c3-66804d9aed65", qty: 1},  
  ]  
}
```

### Критерии приемки

- новый заказ создается в статусе “новый” и дата создания проставляется как текущая дата в UTC
- невозможно создать заказ без строк
- количество по строке заказа не может быть отрицательным
- количество по строке заказа не может быть 0
- невозможно удалить заказ дважды или удалить не созданный заказ
- идентификатор заказа и дату создания нельзя редактировать
- заказы в статусах “оплачен”, “передан в доставку”, “доставлен”, “завершен” нельзя редактировать
- заказы в статусах “передан в доставку”, “доставлен”, “завершен” нельзя удалить
- невозможно получить по идентификатору удаленный или не созданный заказ

### Определение готовности

- сервис реализован в полном объеме согласно заданию
- критерии приемки выполняются
- сервис хранит свое состояние в базе данных и данные не теряются после рестарта

- схема базы данных описана в миграциях и автоматически разворачивается при старте сервиса
- написана инструкция по запуску сервиса
- сервис опубликован на GitHub в публичном репозитории

Срок выполнения задания - 1 неделя с момента получения.