

# Мини ДЗ №2 по КПО.

Магомедов Абдул Омаргаджиевич

БПИ 234

## Предисловие:

Комментарии в коде решил не писать, потому что думал, что тут понятнее объясню с скринами. На github проект в виде папки загрузить не получилось, поэтому просто загрузил архив, который можно скачать. Называется архив ZooManagementTest.zip, это не только тесты, так весь проект называется, внутри будут папки «ZooManagementTest» и «ZooManagementTests». Второй вариант – вот это тесты, ну а первое основная часть. Когда я запускаю проект, у меня ссылка в браузере все нормально открывается, но мне говорили, что у кого то с этим проблемы, так что если при запуске проекта ничего не открылось, нужно перейти по ссылке: <https://localhost:7003/swagger/index.html>. ДЗ сделано на оценку 10, вроде.

## Краткое описание:

При запуске проекта на сайте можно создавать вольеры с определенными типами, после создания вольера в них можно добавить животных, так же с определенными типами, для существующих животных можно создать время кормления, а также процесс feeding/execute, который по логике сам пользователь сайта должен кормить животных в назначенное время. Так же можно перемещать животных из одного вольера в другой, если конечно все удовлетворяет условиям, ну и последнее, в низу сайта есть статистика, где можно посмотреть: кол-во животных, кол-во вольеров, кол-во животных определенного типа, процентное соотношение здоровых/больных животных, кол-во накормленных животных. Далее расскажу более подробно о каждой вкладке.

## Вольеры

Прежде чем создать животных пользователю необходимо создать вольер. Это можно сделать развернув вкладку POST /api/enclosures, далее нажать Try it Out (для взаимодействия с вкладками необходимо всегда

нажимать) и вводить значения, для удобства все поля, в которых нужно что то вводить были сделаны не в формате json, а с помощью query для удобства.

POST /api/enclosures

Parameters

Name	Description
size	integer(\$int32) (query)
maxCapacity	integer(\$int32) (query)
type	string (query)
healthType	string (query)

Execute

Responses

Code	Description	Links
200	OK	No links

Мы видим 4 значения:

size – размер вольера,

maxCapacity – размерность вольера,

type – тип животных, которые будут содержаться в вольере,

healthType – выбрать, для каких животных будет данный вольер, для здоровых или же больных.

Для создания вольера нужно заполнить все поля и нажать Execute. После чего создастся вольер с рандомным ID.

```
{
  "id": "293c267a-157f-4fa7-a7c4-8935ca03865c",
  "type": "Herbivore",
  "healthType": "Healthy",
  "size": 200,
  "maxCapacity": 3,
  "animalIds": []
}
```

После создания вольера можно проверить его существование, развернув уже вкладку GET/api/enclosures, где нажимаю execute он показывает нам все

существующие на данный момент времени вольеры

GET/api/enclosures

Parameters

Cancel

No parameters

ExecuteClear

Responses

Curl

```
curl -X 'GET' \
  https://localhost:7003/api/enclosures' \
  -H 'accept: */*'

```

Request URL

```
https://localhost:7003/api/enclosures

```

Server response

CodeDetails

200

Response body

```
{
  "id": "293c267a-157f-4fa7-a7c4-8935ca83865c",
  "type": "Herbivore",
  "healthType": "Healthy",
  "size": 300,
  "maxCapacity": 3,
  "animalIds": []
}

```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 22 Apr 2025 01:28:48 GMT
server: Kestrel

```

Responses

Code	Description	Links
200	OK	No links

Так же вольер можно удалить, развернув вкладку Delete/api/enclosures, достаточно ввести id вольера, который хотим удалить и нажать execute.

## Животные

Для создания животного нужно нажать Post/api/animals, после чего нам развернется такое окно:

POST/api/animals

Parameters

Try it out

Name	Description
<b>Species</b> * required string (query)	<input type="text" value="Species"/>
<b>Name</b> * required string (query)	<input type="text" value="Name"/>
<b>BirthDate</b> * required string(\$date-time) (query)	Default value : 2025-04-23T12:00:00Z <input type="text" value="2025-04-23T12:00:00Z"/>
<b>Gender</b> * required string (query)	Available values : Male, Female <input type="text" value="Male"/>
<b>FavoriteFood</b> * required string (query)	<input type="text" value="FavoriteFood"/>
<b>Status</b> * required string (query)	Available values : Healthy, Sick <input type="text" value="Healthy"/>
<b>Type</b> * required string (query)	Available values : Predator, Herbivore, Bird, Aquarium <input type="text" value="Predator"/>
<b>EnclosureId</b> * required string(\$uuid) (query)	<input type="text" value="EnclosureId"/>

Каждое поле необходимо заполнить, тут так же как и в вольерах на некоторых полях доступна опция выбора из существующих вариантов, таких как gender, status и type. Gender только Male или Female, status – Healthy или Sick, Type – Тип животного, хищник, травоядное, птица, аквариум. Так же в поле birthdate указано дефолтное значение, чтобы долго не набирать.

В поле EnclosureID вставляем id вольера, который можем получить в Get/api/enclosures. Нажимаем execute и создаем новое животное. Теперь поговорим о несостыковках, есть такие моменты, когда добавление животное в вольер было бы нелогично, именно такие моменты и были предусмотрены, к примеру:

- 1) Если в вольере животные не одного типа (Хищник с Травоядным)
- 2) Если больное животное занести в вольер к здоровым животным, именно поэтому был создан тип для вольеров.
- 3) Ну и если добавлять животное в вольер, когда он уже заполнен.

Во всех этих случаях выдаст ошибку(не системную), где напишется проблема.

Тип животного не соответствует типу вольера.

Данный вольер предназначен только для здоровых животных.

Данный вольер предназначен только для больных животных.

Вольер заполнен. Невозможно добавить новое животное.

Вкладки Get/api/animals и Delete/api/animals работают аналогично вольерам.

# Кормление

Открываем Post/api/feeding/add и видим:

POST /api/feeding/add

Parameters

Name	Description
animalId string(\$uuid) (query)	<input type="text" value="animalId"/>
time string(\$date-time) (query)	<input type="text" value="time"/>
food string (query)	<input type="text" value="food"/>

Execute

Responses

Code	Description	Links
200	OK	No links

В animalID добавляем id животного, которому хотим назначить время кормления, в time нужно заполнить поле так же, как и birthdate в animals, можно скопировать дефолтное значение и переделать. Ну а в food любую еду, это никак контролироваться не может, к сожалению. Нажимаем execute и добавляем опцию кормления животного.

Далее переходим в Post/api/feeding/feed, тут происходит кормление всех записанных животных, нужно просто нажать execute. Кормление не происходит просто так, главный фактор – время. На сайте оно указано как UTC, то есть -3 от МСК. Для примера я сделал кормление на одном и том же животном, но в разное время и сделал execute. Просмотреть накормлено животное или же нет можно в Get/api/feeding/schedule

```
[
  {
    "id": "f1ceb57c-e4d8-469b-8e74-f9886f086644",
    "animalId": "a591171e-cffa-4954-a303-3c3f84e693bb",
    "feedingTime": "2025-04-23T12:00:00Z",
    "foodType": "Grass",
    "isCompleted": false
  },
  {
    "id": "57f48ef3-969e-48b6-bdaf-704edb972032",
    "animalId": "a591171e-cffa-4954-a303-3c3f84e693bb",
    "feedingTime": "2025-04-22T02:00:00Z",
    "foodType": "Grass",
    "isCompleted": false
  },
  {
    "id": "7fd2e97d-cf02-469a-9a65-5102dd4ea4c9",
    "animalId": "a591171e-cffa-4954-a303-3c3f84e693bb",
    "feedingTime": "2025-04-22T01:37:00Z",
    "foodType": "Grass",
    "isCompleted": true
  },
  {
    "id": "b0f701df-5528-470e-a09f-594cfaba9482",
    "animalId": "a591171e-cffa-4954-a303-3c3f84e693bb",
    "feedingTime": "2025-04-22T01:39:00Z",
    "foodType": "Grass",
    "isCompleted": true
  }
]
```

Проверка проводилась в 4:39 МСК.

# Трансфер

Тут только одна вкладка:

POST

/api/transfer

^

Parameters

Try it out

Name	Description
animalId string(\$uuid) (query)	<input type="text" value="animalId"/>
targetEnclosureId string(\$uuid) (query)	<input type="text" value="targetEnclosureId"/>

Responses

Code	Description	Links
200	OK	No links

Соответственно вводим id животного, которого хотим переместить ну и id вольера, в который хотим переместить. Тут так же обработаны исключительные случае, учитывается здоровье животного, а также заполненность вольера, вот сообщения с обработанными случаями:

Данный вольер предназначен только для больных животных.

Данный вольер предназначен только для здоровых животных.

Вольер переполнен. Перемещение невозможно.

Целевой вольер не найден.

В случае же если все условия соблюдены, выводится:

Животное успешно перемещено.

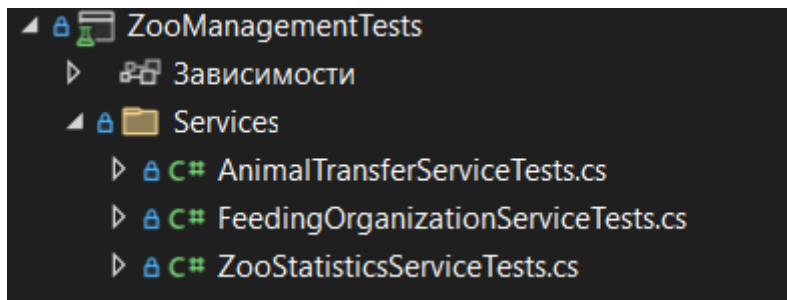
## Статистика

В статистике так же только одна вкладка Get, в которой описана вся информация, пример можно увидеть на картинке:

```
{
  "totalAnimals": 5,
  "totalEnclosures": 3,
  "animalsByType": {
    "Herbivore": 5
  },
  "healthyPercentage": 80,
  "sickPercentage": 20,
  "totalFedAnimals": 2
}
```







## Тесты

Также предусмотрены тесты, которые покрывают 65% кода:

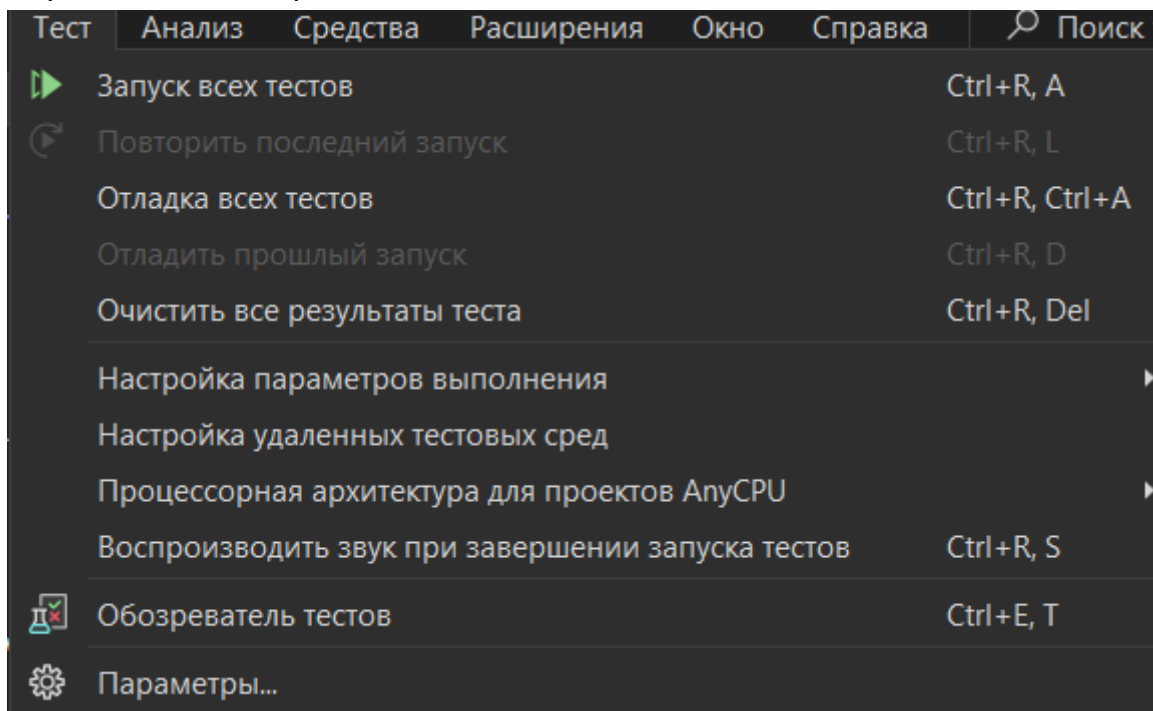


Было создано как отдельный проект. Для успешной проверки тестов необходимо скачать следующие 4 nuget-пакета:

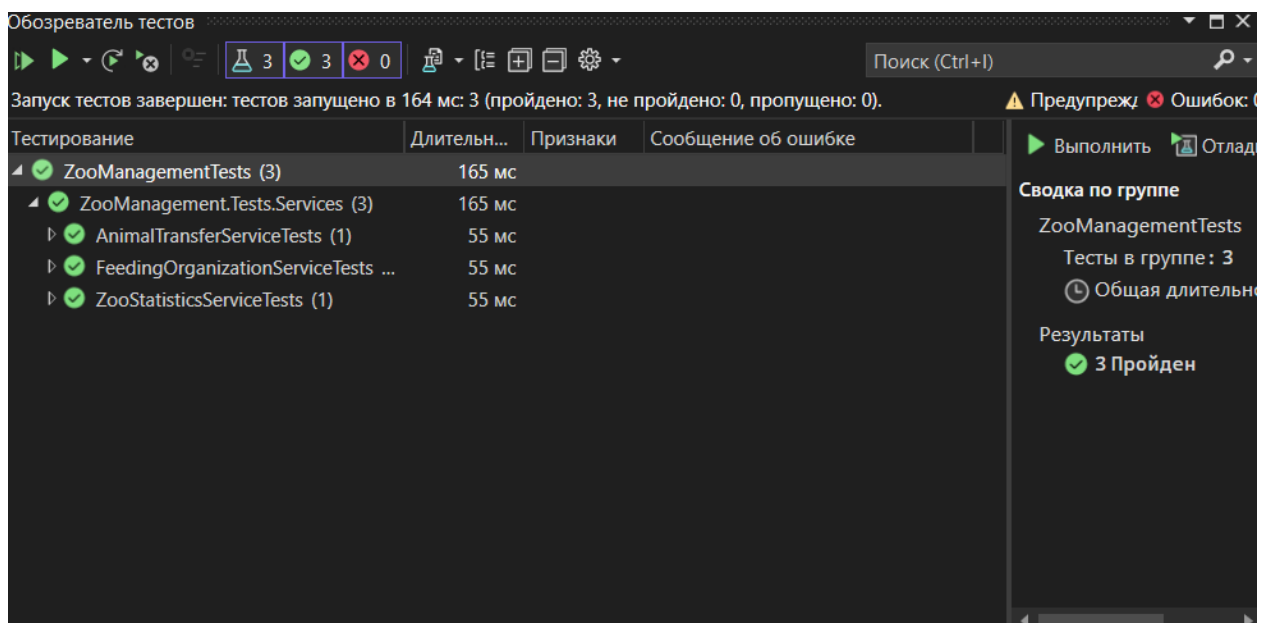
- 1) xunit
- 2) xunit.runner.visualstudio
- 3) Moq
- 4) FluentAssertions

	<b>coverlet.collector</b> от tonerdo Coverlet is a cross platform code coverage library for .NET, with support for line, branch and method coverage.	6.0.0 ✖ 6.0.4 ⬆
	<b>FluentAssertions</b> от Dennis Doomen, Jonas Nyrup, Xceed A very extensive set of extension methods that allow you to more naturally specify the expected outcome of a TDD or	8.2.0
	<b>Microsoft.NET.Test.Sdk</b> от Microsoft The MSbuild targets and properties for building .NET test projects.	17.8.0 17.13.0
	<b>Moq</b> от Daniel Cazzulino, kzu Moq is the most popular and friendly mocking framework for .NET.	4.20.72
	<b>xunit</b> от jnewkirk, bradwilson xUnit.net is a developer testing framework, built to support Test Driven Development, with a design goal of extreme simplicity and alignment with framework features.	2.5.3 2.9.3
	<b>xunit.runner.visualstudio</b> от jnewkirk, bradwilson Visual Studio 2022+ Test Explorer runner for the xUnit.net framework. Capable of running xUnit.net v1.9.2 and v2.0+ tests. Supports .NET 4.6.2 or later, and .NET 6 or later.	2.5.3 3.0.2

Для проверки в Visual Studio в верхней панели нажимаем «Тест», далее переходим «Обозреватель тестов»:



В обозревателе тестов нажимаем запуск, и видим как все тесты были успешно пройдены.



Это все