|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №5**

по дисциплине «Интерфейсы прикладного программирования»

**Тема практической работы:**

**Проектирование и реализация простейшего REST API с помощью NodeJS (CRUD APP)**

**Студент группы** ИКБО-01-21 Маров Герман Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** Зарипов Е.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Москва 2023

**ОГЛАВЛЕНИЕ**

[Цель работы 3](#_Toc115999382)

[Ход работы 3](#_Toc115999383)

[Вывод 16](#_Toc115999384)

# Цель работы

Целью данной практической работы является знакомство с архитектурными принципами REST и создание простейшего REST API в стеке MEAN.

Задание: используя материалы практической работы и предложенный стек технологий, необходимо создать и развернуть на сервере Heroku приложение для управления списком телефонных контактов. Стоит обратить внимание на то, что используемый стек технологий очень активно развивается, а также сама облачная PaaS система Heroku может вводить определённые ограничения на поддержку используемых в данном проекте технологий, поэтому могут возникать определённые трудности со сборкой проекта, при попытке простого копирования исходников кода из данной практической работы. По согласованию с преподавателем, можно выбрать и другой стек технологий, а также и другую облачную систему.

# Ход работы

На рисунке 1 показана стартовая страница cyclic.sh, на котором будет запущен сервер.

Так на листинге 1 представлен исходный код примерного приложения на node.js.

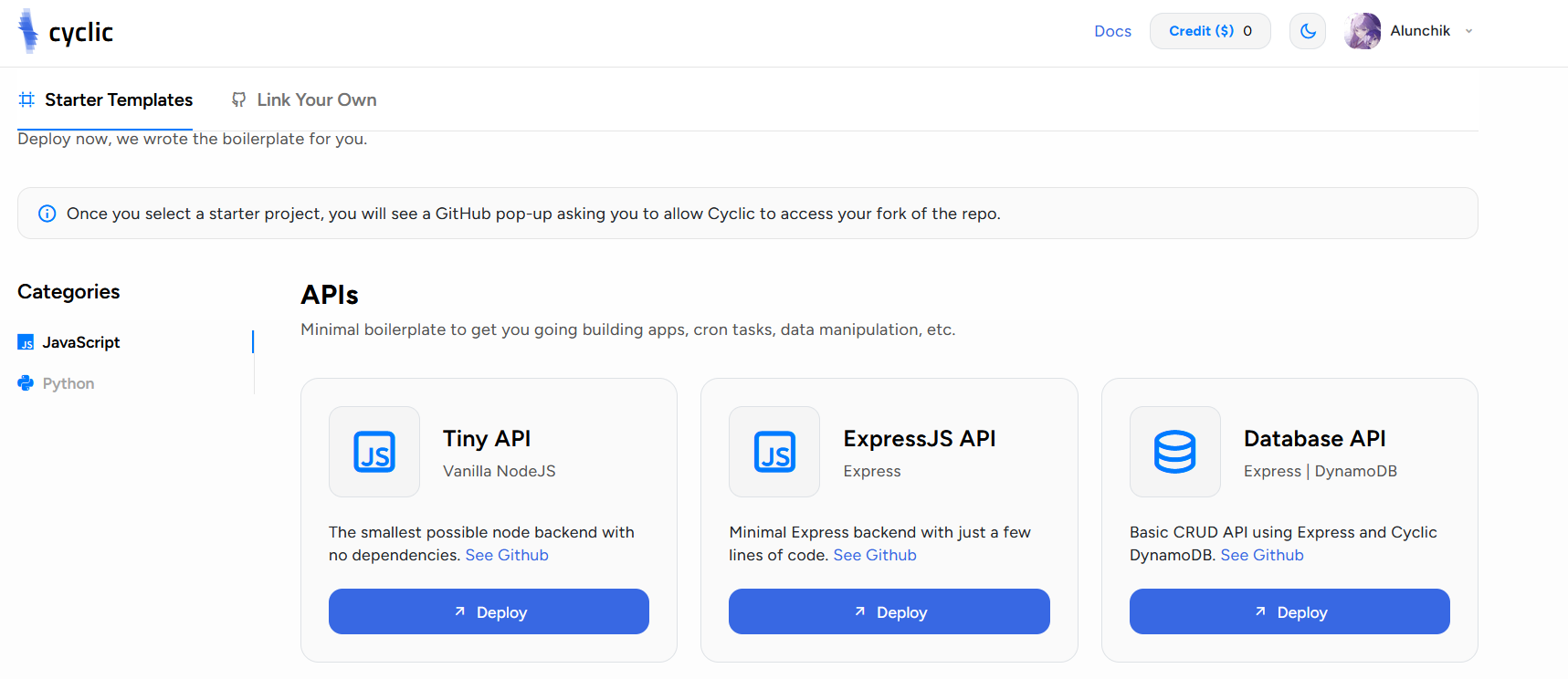
****

Рисунок 1 – Стартовая страница cyclic.sh

Листинг 1 – Исходный код приложения

const express = require('express')

const app = express()

const db = require('@cyclic.sh/dynamodb')

app.use(express.json())

app.use(express.urlencoded({ extended: true }))

app.post('/:col/:key', async (req, res) => {

console.log(req.body)

const col = req.params.col

const key = req.params.key

console.log(`from collection: ${col} delete key: ${key} with params ${JSON.stringify(req.params)}`)

const item = await db.collection(col).set(key, req.body)

console.log(JSON.stringify(item, null, 2))

res.json(item).end()

})

// Delete an item

app.delete('/:col/:key', async (req, res) => {

const col = req.params.col

const key = req.params.key

console.log(`from collection: ${col} delete key: ${key} with params ${JSON.stringify(req.params)}`)

const item = await db.collection(col).delete(key)

console.log(JSON.stringify(item, null, 2))

res.json(item).end()

})

// Get a single item

app.get('/:col/:key', async (req, res) => {

const col = req.params.col

const key = req.params.key

console.log(`from collection: ${col} get key: ${key} with params ${JSON.stringify(req.params)}`)

const item = await db.collection(col).get(key)

console.log(JSON.stringify(item, null, 2))

res.json(item).end()

})

// Get a full listing

app.get('/:col', async (req, res) => {

const col = req.params.col

console.log(`list collection: ${col} with params: ${JSON.stringify(req.params)}`)

const items = await db.collection(col).list()

console.log(JSON.stringify(items, null, 2))

res.json(items).end()

})

// Catch all handler for all other request.

app.use('\*', (req, res) => {

res.json({ msg: 'no route handler found' }).end()

})

// Start the server

const port = process.env.PORT || 3000

app.listen(port, () => {

console.log(`index.js listening on ${port}`)

})

На данном этапе разработанное приложение уже развернуто на хостинге cyclic.sh и доступно всем пользователям сети Интернет. На рисунках 2-6 продемонстрирован его функционал.

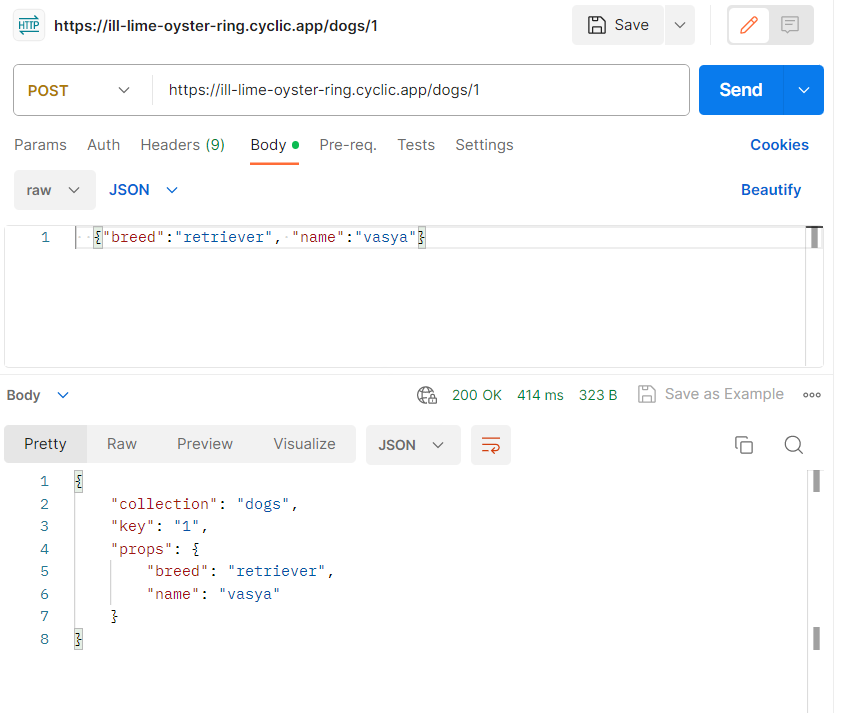


Рисунок 2 – Добавление новой собаки

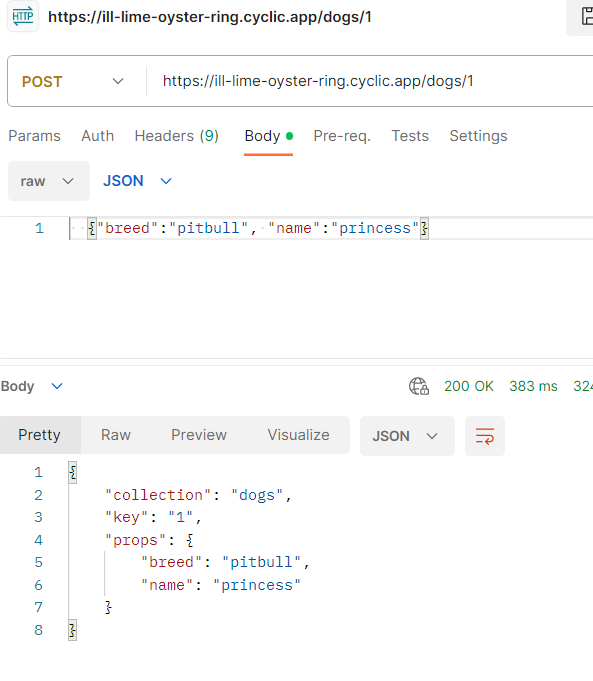


Рисунок 3 – Добавление новой собаки

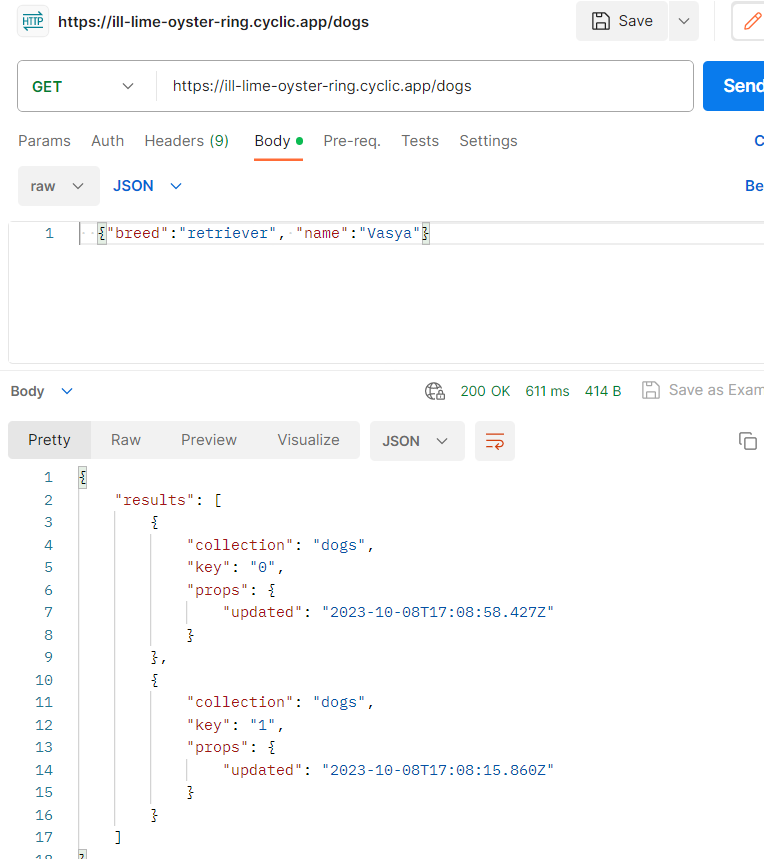


Рисунок 4 – Получение всех собак

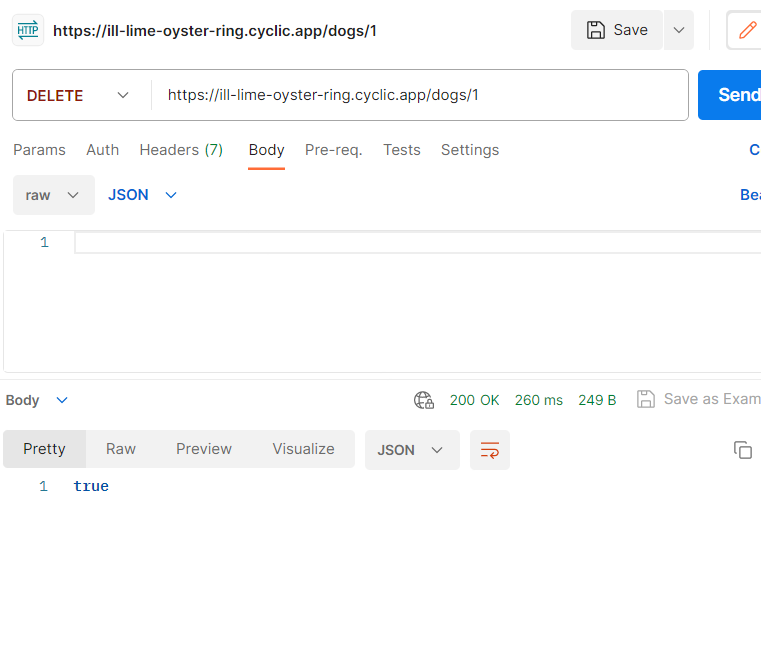


Рисунок 5 – Удаление собаки по id

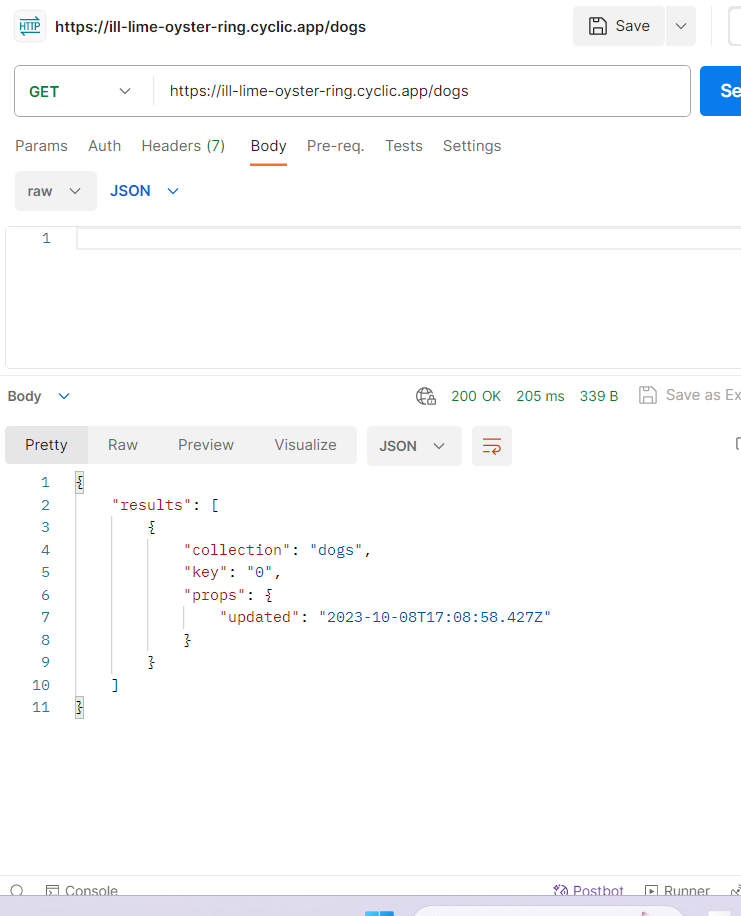


Рисунок 6 – Получение всех собак после удаления

# Вывод

В результате выполнения данной практической работы были получены навыки по проектированию простейшего RESTful API, представляющее из себя базовое приложение для управления списком телефонных контактов, и его развертыванию на сервере cyclic.