



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения
(ИиППО)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4
по дисциплине

«Проектирование и разработка серверных частей интернет-ресурсов»

Выполнил студент группы ИКБО-20-23

Комисарик М.А.

Принял Ассистент кафедры ИиППО

Благирев М.М.

Практическая работа выполнена

«__»_____202__ г.

(подпись студента)

«Зачтено»

«__»_____202__ г.

(подпись руководителя)

Москва 2025

ХОД РАБОТЫ

```
location /api/ {  
    rewrite ^/api/(.*)$ /api.php/$1 break;  
    proxy_pass http://apache_backend;  
    proxy_buffering off;  
    proxy_http_version 1.1;  
    proxy_set_header Host $host;  
}
```

Рисунок 1 – Новый блок location в файле nginx.conf

```
1 <?php  
2 header( header: 'Content-Type: application/json; charset=utf-8');  
3 require_once '../includes/db_connect.php';  
4  
5 $method = $_SERVER['REQUEST_METHOD'];  
6 if (!key_exists( key: 'PATH_INFO', $_SERVER))  
7 {  
8     echo "Enter the query";  
9     return;  
10 }  
11 $request = explode( separator: '/', trim($_SERVER['PATH_INFO'], characters: '/'));  
12 $entity = $request[0] ?? '';  
13 $id = $request[1] ?? null;  
14  
15 if (!isset($_SERVER['PHP_AUTH_USER']))  
16 {  
17     header( header: 'WWW-Authenticate: Basic realm="Autoservice API");  
18     http_response_code( response_code: 401);  
19     echo json_encode(['error' => 'Требуется аутентификация.']);  
20     exit;  
21 }  
22 if ($_SERVER['PHP_AUTH_USER'] != 'admin' || $_SERVER['PHP_AUTH_PW'] != 'admin123')  
23 {  
24     http_response_code( response_code: 403);  
25     echo json_encode(['error' => 'Неверные учетные данные.']);  
26     exit;  
27 }
```

Рисунок 2 – api.php, часть 1

```

29     try
30     {
31         switch ($method)
32         {
33             case 'GET':
34                 if ($entity == 'services')
35                 {
36                     if ($id)
37                     {
38                         $statement = $pdo->prepare( query: "SELECT * FROM services WHERE id = ?");
39                         $statement->execute([$id]);
40                         $service = $statement->fetch( mode: PDO::FETCH_ASSOC);
41                         if ($service)
42                         {
43                             echo json_encode($service);
44                         } else
45                         {
46                             http_response_code( response_code: 404);
47                             echo json_encode(['error' => 'Услуга не найдена.']);
48                         }
49                     } else
50                     {
51                         $statement = $pdo->query( query: "SELECT * FROM services");
52                         $services = $statement->fetchAll( mode: PDO::FETCH_ASSOC);
53                         echo json_encode($services);
54                     }
55                 }
56                 break;

```

Рисунок 3 – api.php, часть 2

```

57     case 'POST':
58         $input = json_decode(file_get_contents( filename: 'php://input'), associative: true);
59         if ($entity == 'services')
60         {
61             $statement = $pdo->prepare( query: "INSERT INTO services (name, description, price, duration, category) VALUES (?, ?, ?, ?, ?)");
62             $statement->execute([$input['name'], $input['description'], $input['price'], $input['duration'], $input['category']]);
63             http_response_code( response_code: 201);
64             echo json_encode(['message' => 'Услуга успешно создана.', 'id' => $pdo->lastInsertId()]);
65         }
66         break;
67     case 'PUT':
68         $input = json_decode(file_get_contents( filename: 'php://input'), associative: true);
69         if ($entity == 'services' && $id)
70         {
71             $statement = $pdo->prepare( query: "UPDATE services SET name=?, description=?, price=?, duration=?, category=? WHERE id=?");
72             $statement->execute([$input['name'], $input['description'], $input['price'], $input['duration'], $input['category'], $id]);
73             echo json_encode(['message' => 'Услуга успешно обновлена.']);
74         }
75         break;
76     case 'DELETE':
77         if ($entity == 'services' && $id)
78         {
79             $statement = $pdo->prepare( query: "DELETE FROM services WHERE id=?");
80             $statement->execute([$id]);
81             echo json_encode(['message' => 'Услуга успешно удалена.']);
82         }
83         break;

```

Рисунок 4 – api.php, часть 3

```

84         default:
85             http_response_code( response_code: 405);
86             echo json_encode(['error' => 'Метод не разрешен.']);
87             break;
88     }
89 }
90 catch (Exception $e)
91 {
92     http_response_code( response_code: 500);
93     echo json_encode(['error' => 'Внутренняя ошибка сервера: ' . $e->getMessage()]);
94 }

```

Рисунок 5 – api.php, часть 4

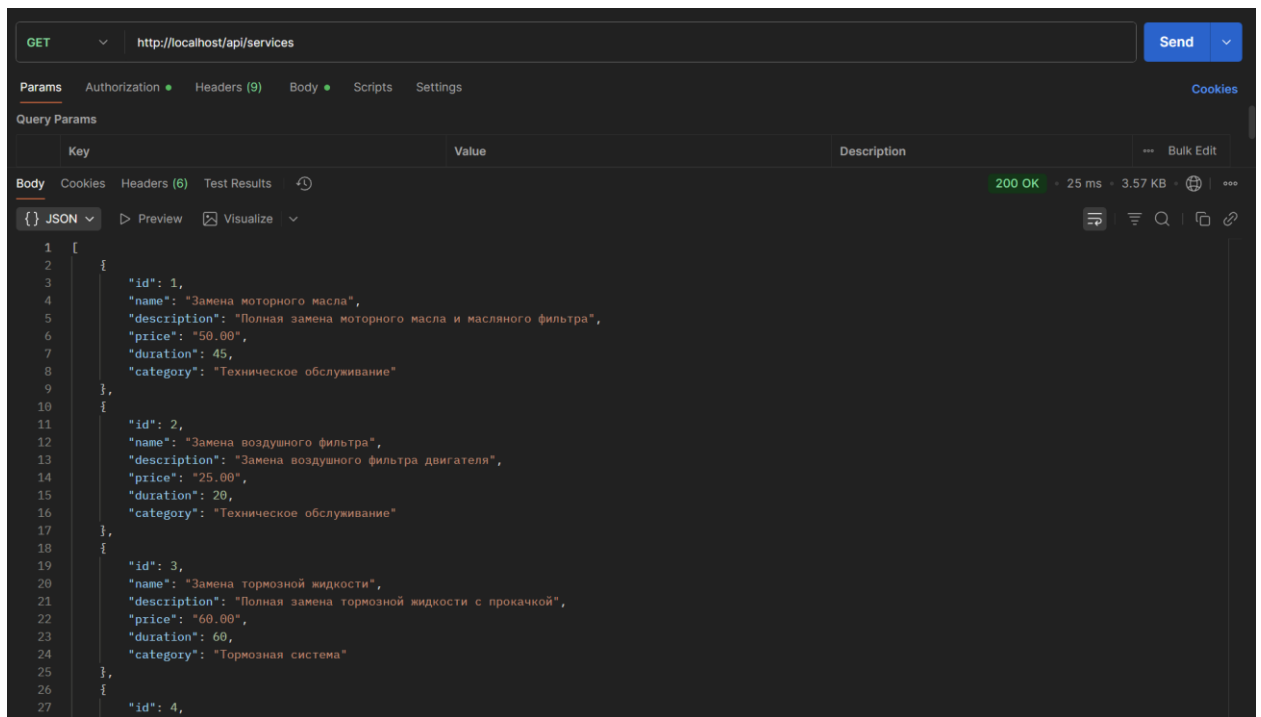


Рисунок 6 – GET запрос на получение всех услуг

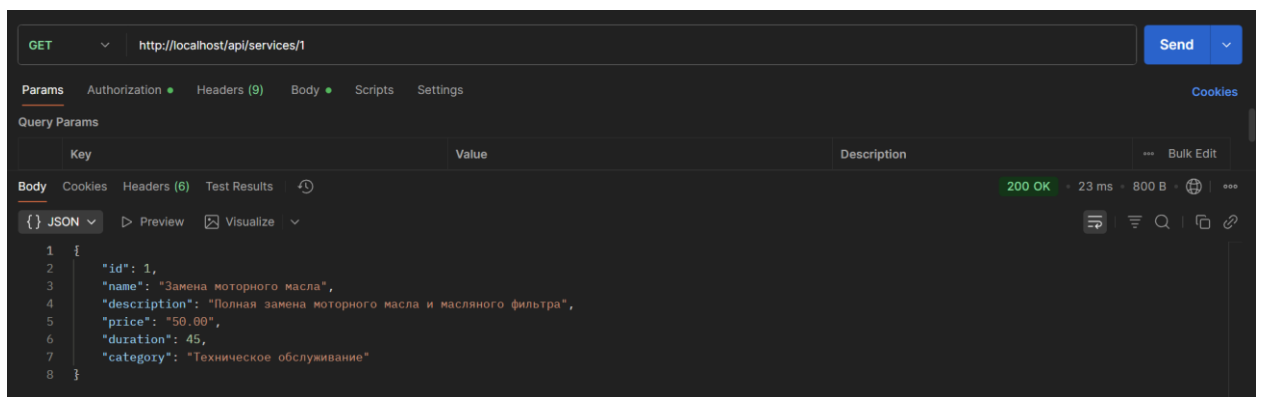


Рисунок 7 – GET запрос на получение услуги с id 1

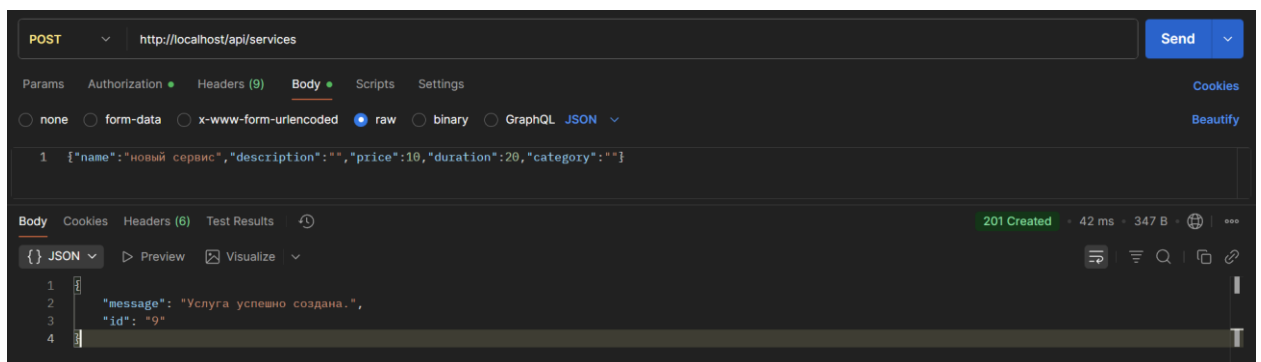


Рисунок 8 – POST запрос на создание услуги

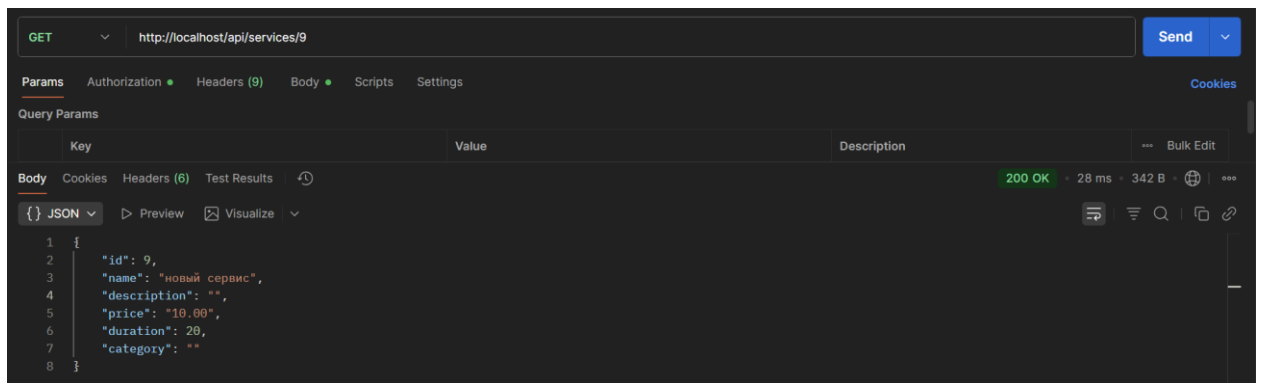


Рисунок 9 – Получение добавленной услуги

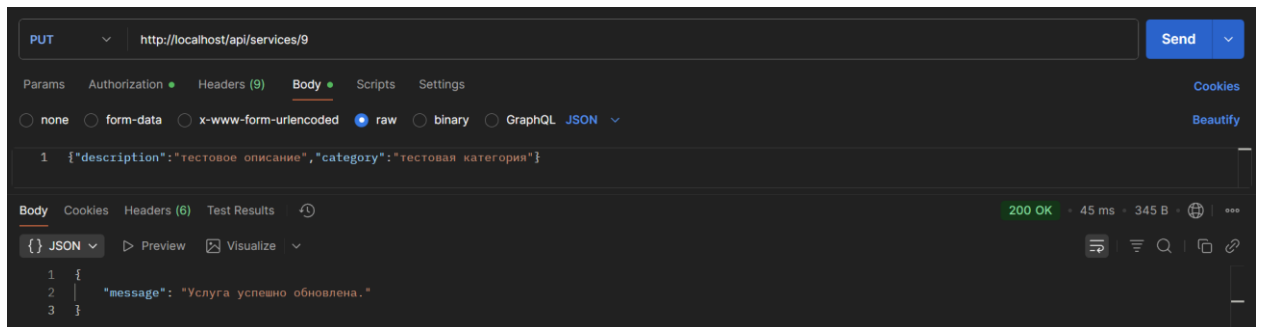


Рисунок 10 – PUT запрос на изменение новой услуги

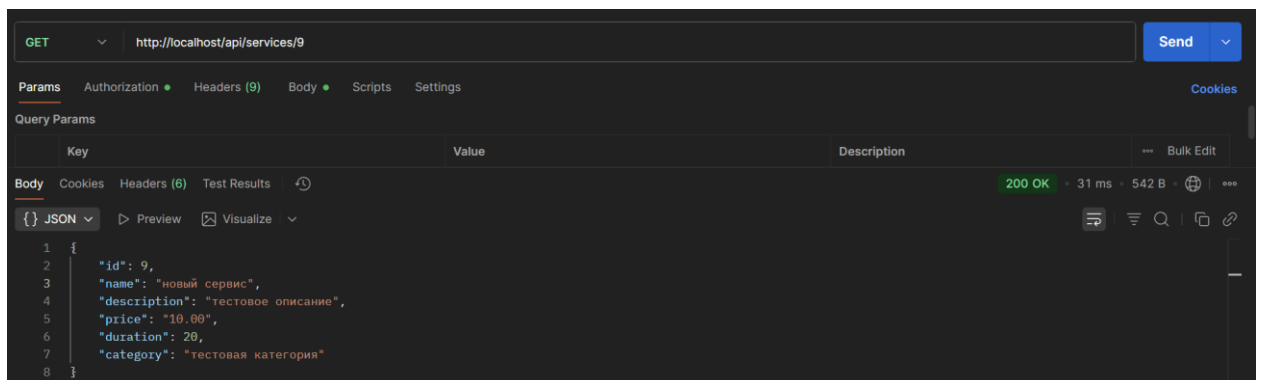


Рисунок 11 – Получение измененной услуги

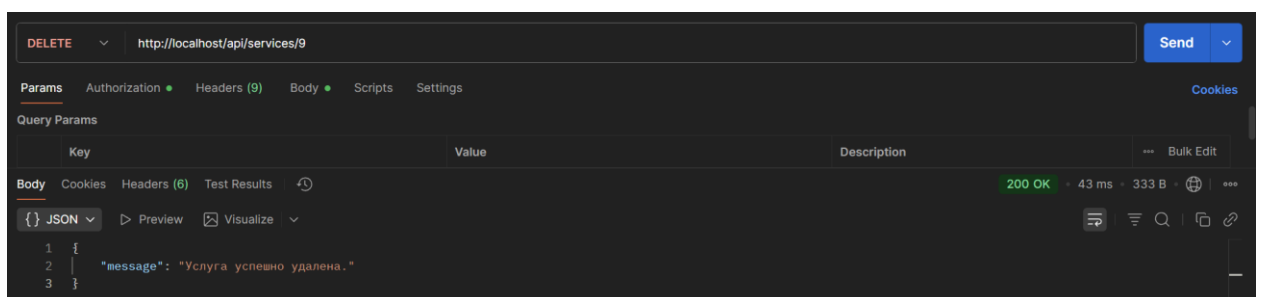


Рисунок 12 – Удаление новой услуги

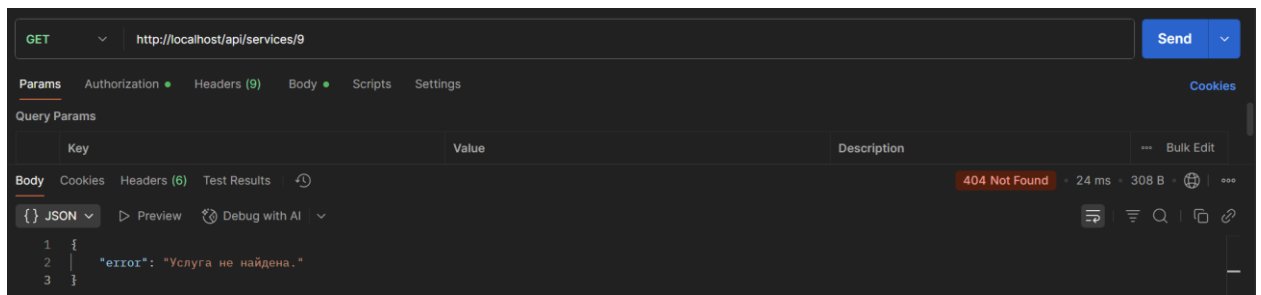


Рисунок 13 – Проверка успешности удаления новой услуги