Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №1**

Дисциплина: «Тестирование программного обеспечения»

Выполнила:

студентка группы ИВТАСбд-41

Микка Е. И.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск,

2024

**Задание**

Данной лабораторной работе необходимо разработать API-сервер, который обрабатывает API-запросы. Разработанные запросы необходимо протестировать с помощью программы Postman.

**Ход выполнения работы**

Данный код реализует простой API для управления записями с использованием CherryPy и SQLite.

При создании экземпляра класса Record происходит инициализация базы данных SQLite с именем records.db и создание таблицы records, если она еще не существует. Таблица содержит два поля: id (уникальный идентификатор) и name (имя записи).

Метод create.

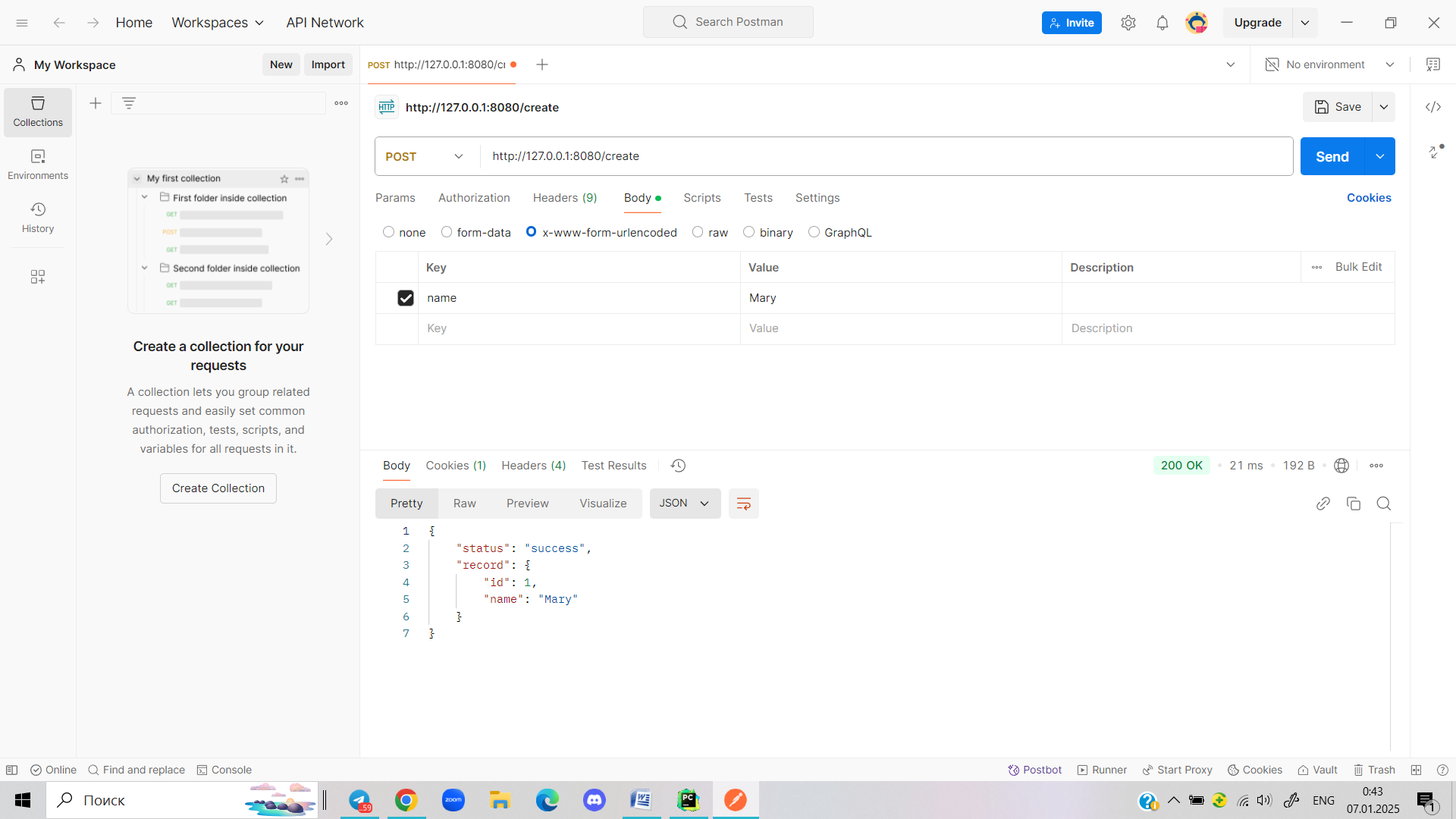
Принимает параметр name, добавляет новую запись в таблицу и возвращает статус операции вместе с созданной записью.

Рис 1. Пример работы метода create

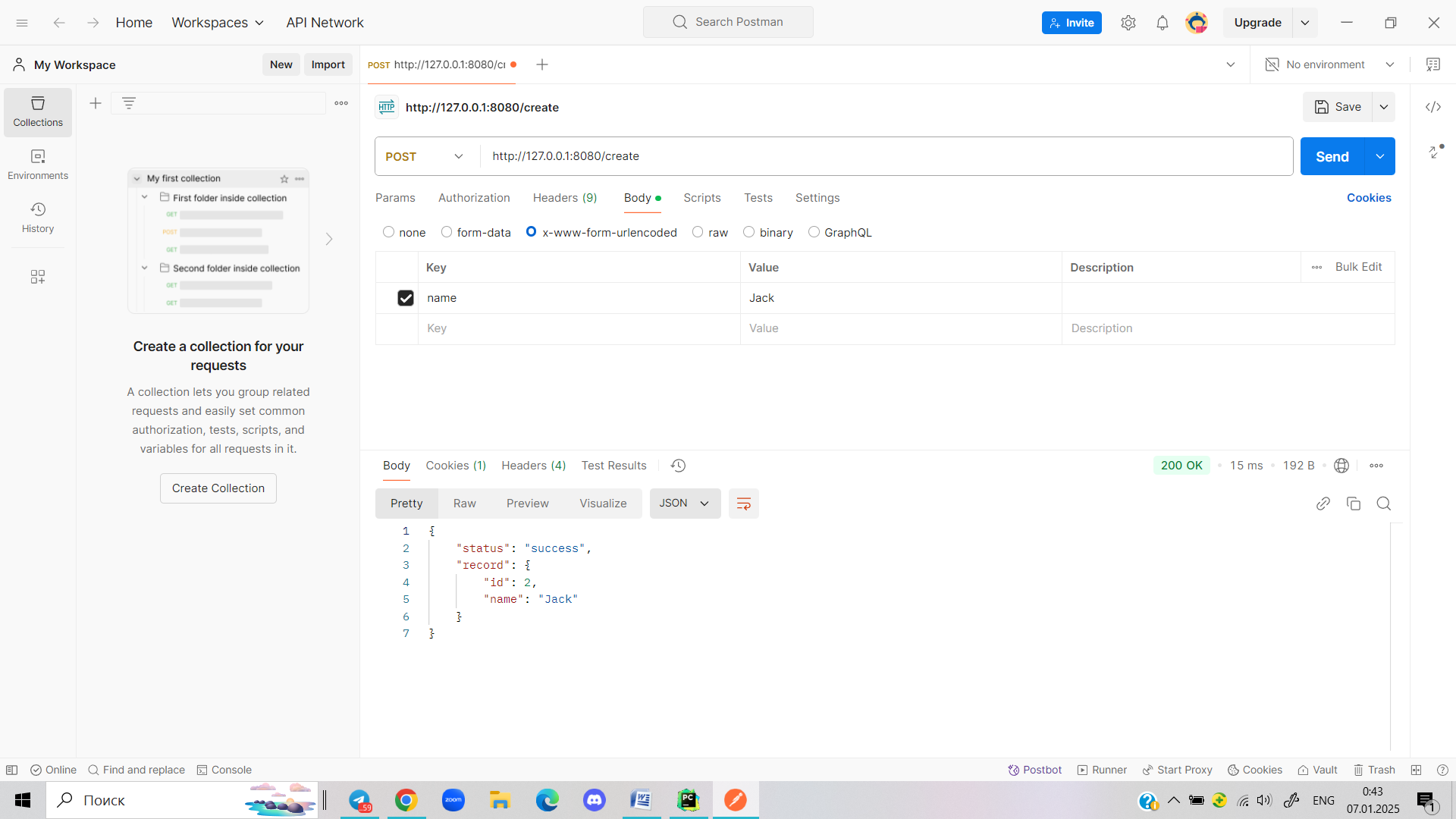


Рис 2. Пример работы метода create

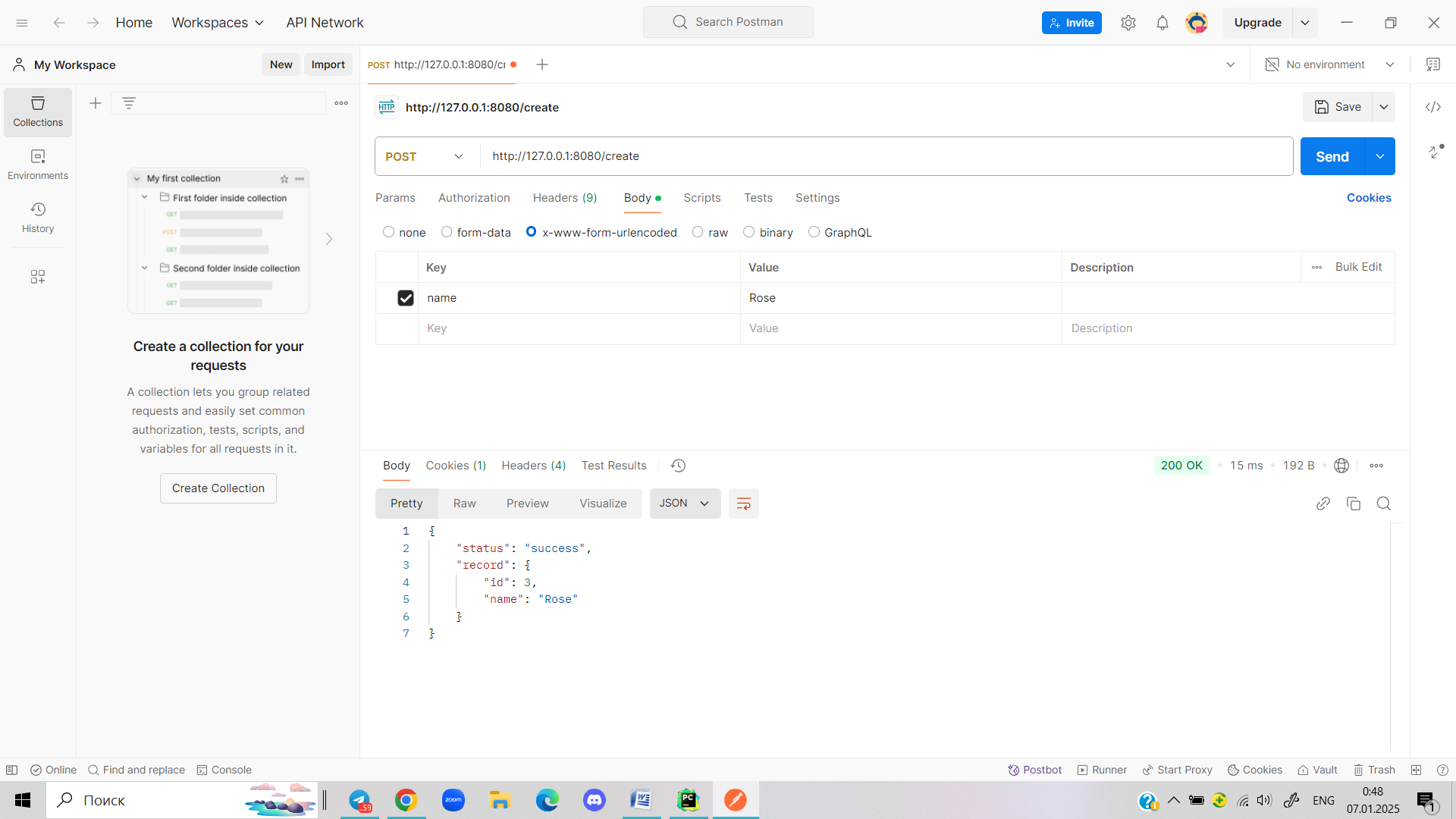


Рис 3. Пример работы метода create

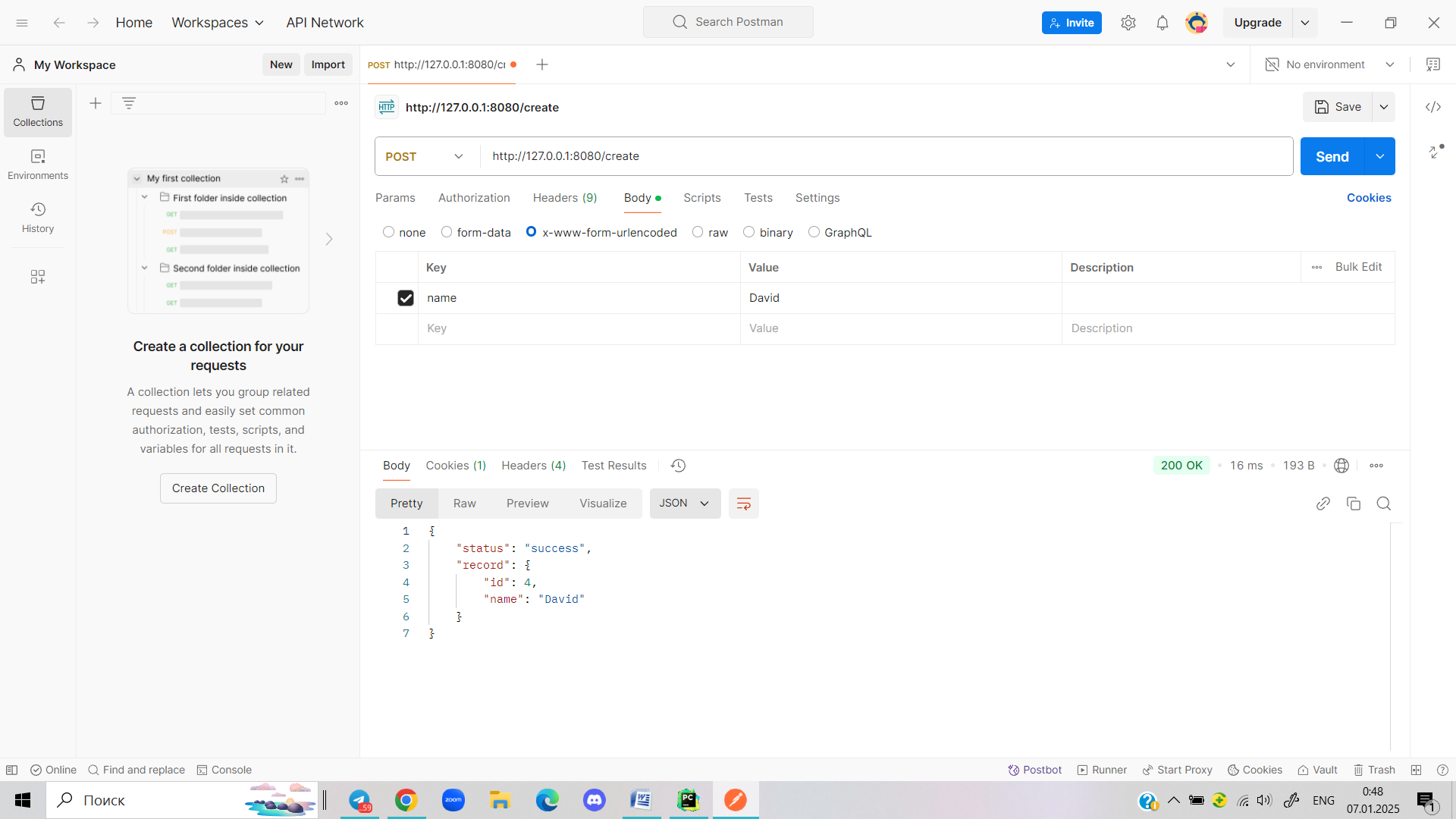


Рис 4. Пример работы метода create

Метод read.

Принимает record\_id, извлекает соответствующую запись из базы данных и возвращает ее. Если запись не найдена, возвращает ошибку 404.

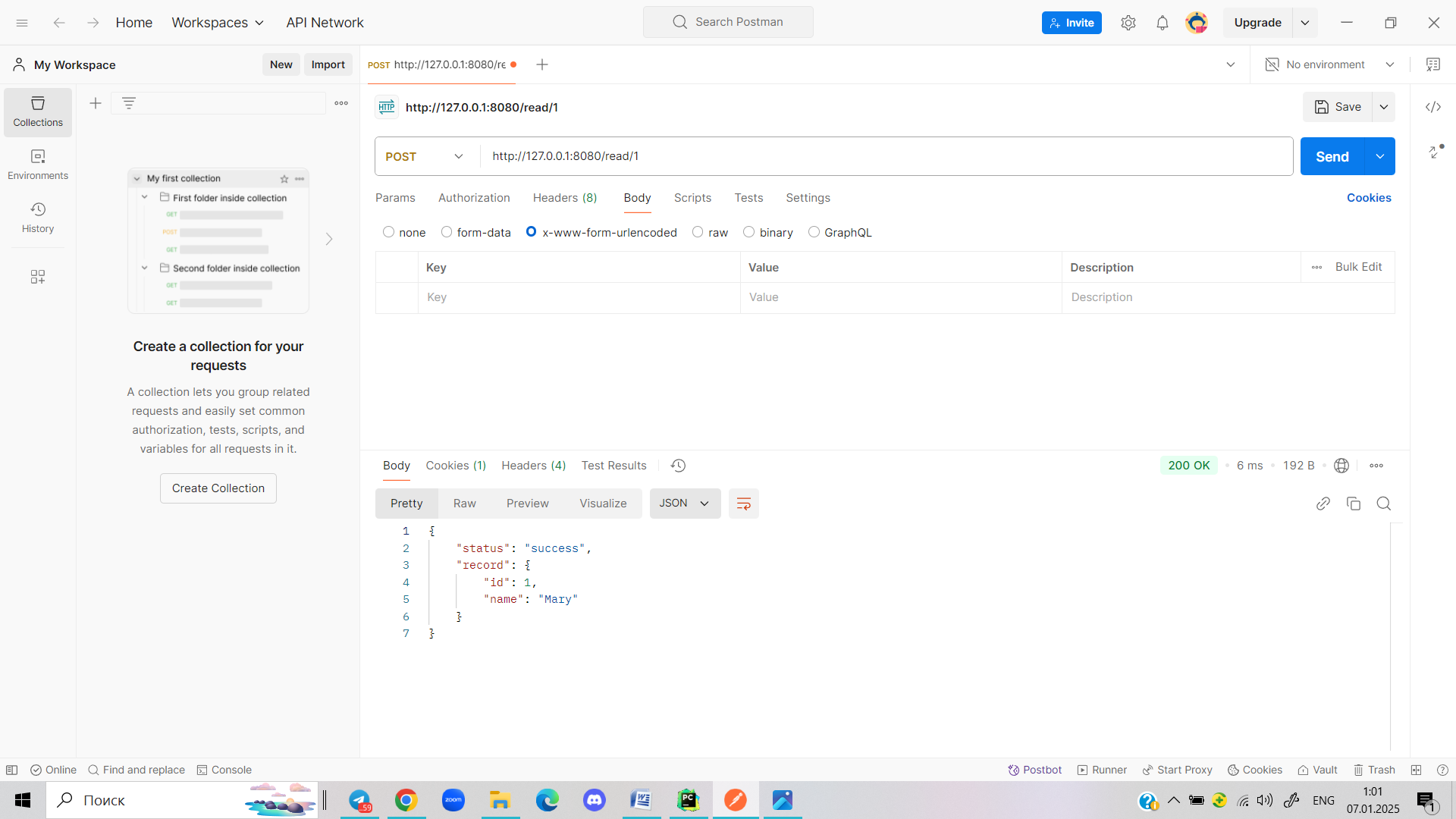


Рис 5. Пример работы метода read.

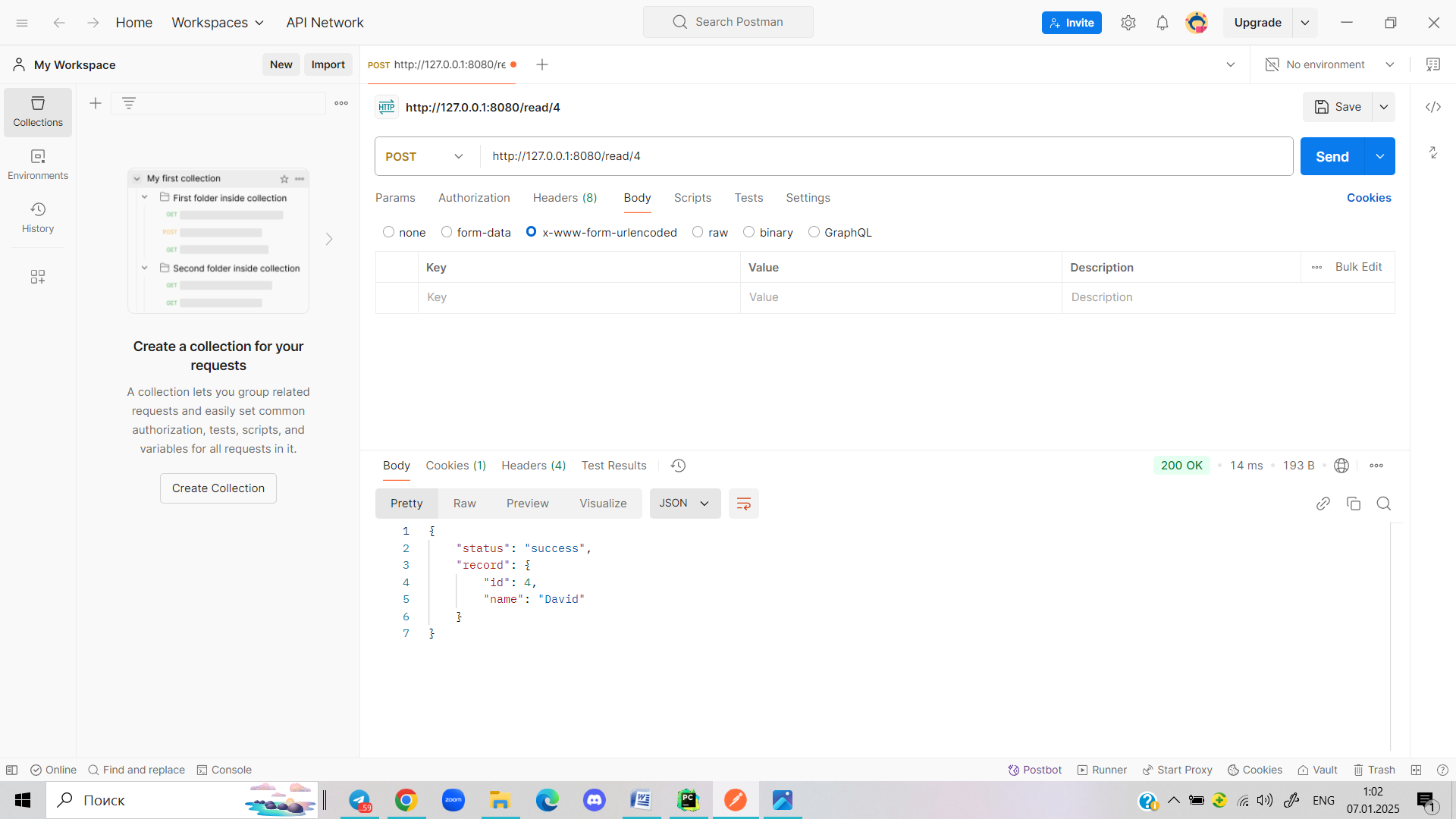


Рис 6. Пример работы метода read.

Метод update.

Принимает record\_id и новое имя name, обновляет запись в базе данных. Если запись не найдена, возвращает ошибку 404.

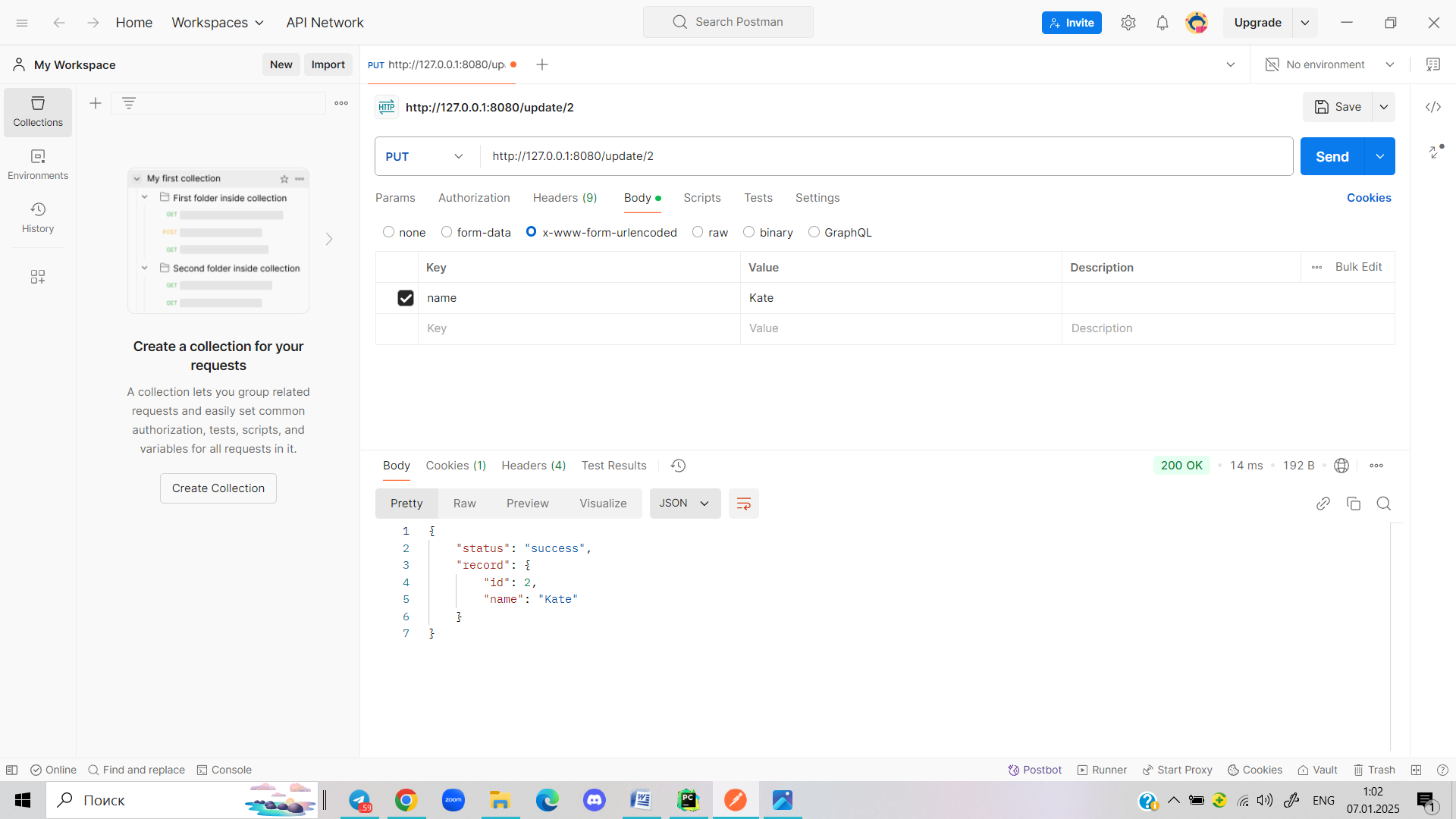


Рис 7. Пример работы метода update.

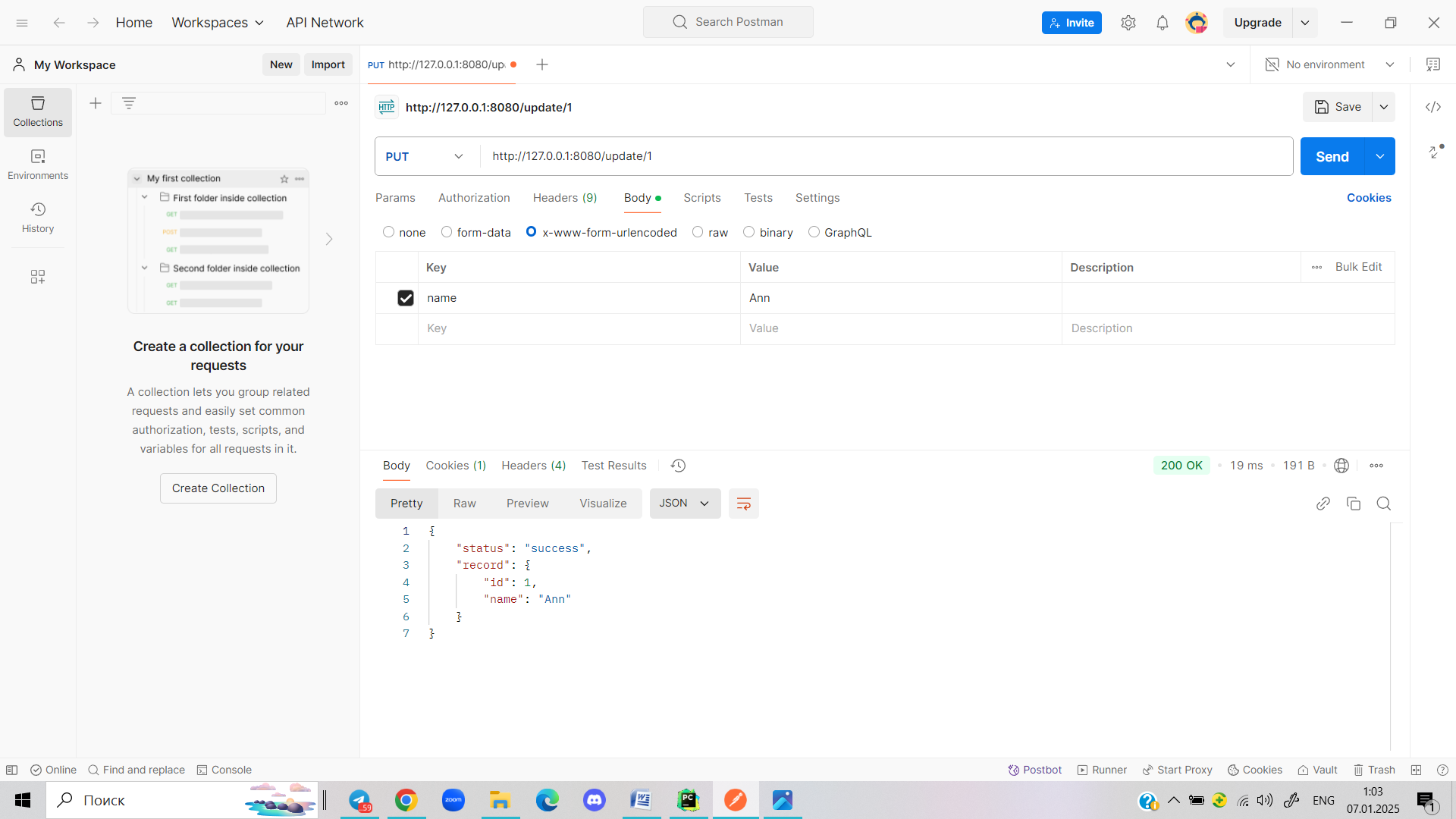


Рис 8. Пример работы метода update.

Метод delete.

Принимает record\_id, удаляет соответствующую запись из базы данных и возвращает статус операции. Если запись не найдена, возвращает ошибку 404.

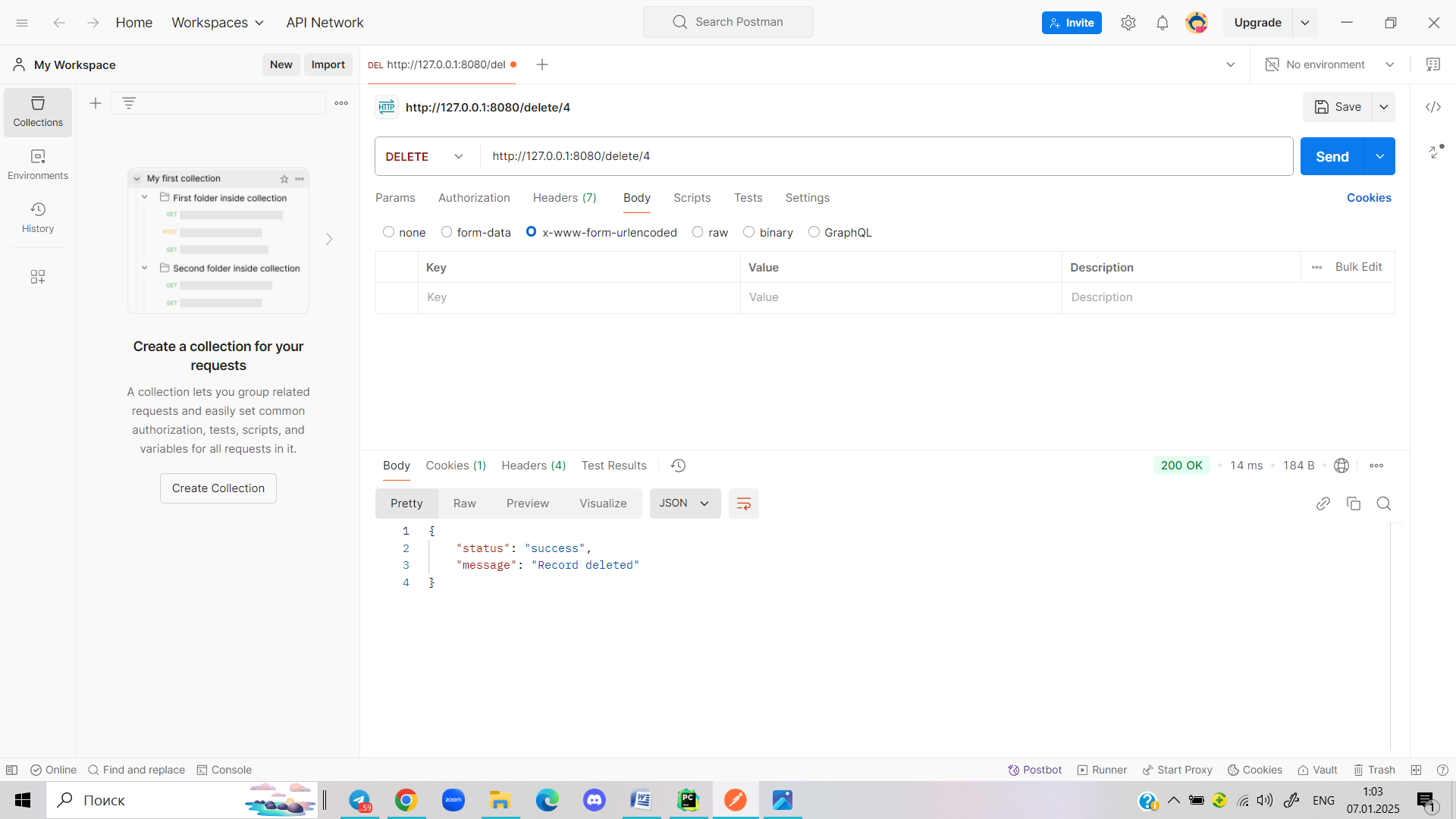


Рис 9. Пример работы метода delete.

**Приложение**

import cherrypy  
import sqlite3  
import json  
  
class Record:  
 def \_\_init\_\_(self):  
 self.db\_name = 'records.db'  
 self.create\_table()  
  
 def create\_table(self):  
 with sqlite3.connect(self.db\_name) as conn:  
 cursor = conn.cursor()  
 cursor.execute('''  
 CREATE TABLE IF NOT EXISTS records (  
 id INTEGER PRIMARY KEY AUTOINCREMENT,  
 name TEXT NOT NULL  
 )  
 ''')  
 conn.commit()  
  
 def get\_db\_connection(self):  
 conn = sqlite3.connect(self.db\_name)  
 conn.row\_factory = sqlite3.Row   
 return conn  
  
 @cherrypy.expose  
 @cherrypy.tools.json\_out()  
 def index(self):  
 return {"message": "Welcome to the Records API. Use /create, /read, /update, and /delete to manage records."}  
  
 @cherrypy.expose  
 @cherrypy.tools.json\_out()  
 def create(self, name):  
 with self.get\_db\_connection() as conn:  
 cursor = conn.cursor()  
 cursor.execute('INSERT INTO records (name) VALUES (?)', (name,))  
 conn.commit()  
 record\_id = cursor.lastrowid  
 return {'status': 'success', 'record': {'id': record\_id, 'name': name}}  
  
 @cherrypy.expose  
 @cherrypy.tools.json\_out()  
 def read(self, record\_id):  
 record\_id = int(record\_id)  
 with self.get\_db\_connection() as conn:  
 cursor = conn.cursor()  
 cursor.execute('SELECT \* FROM records WHERE id = ?', (record\_id,))  
 record = cursor.fetchone()  
 if record:  
 return {'status': 'success', 'record': dict(record)}  
 else:  
 return {'status': 'error', 'message': 'Record not found'}, 404  
  
 @cherrypy.expose  
 @cherrypy.tools.json\_out()  
 def update(self, record\_id, name):  
 record\_id = int(record\_id)  
 with self.get\_db\_connection() as conn:  
 cursor = conn.cursor()  
 cursor.execute('UPDATE records SET name = ? WHERE id = ?', (name, record\_id))  
 if cursor.rowcount == 0:  
 return {'status': 'error', 'message': 'Record not found'}, 404  
 conn.commit()  
 return {'status': 'success', 'record': {'id': record\_id, 'name': name}}  
  
 @cherrypy.expose  
 @cherrypy.tools.json\_out()  
 def delete(self, record\_id):  
 record\_id = int(record\_id)  
 with self.get\_db\_connection() as conn:  
 cursor = conn.cursor()  
 cursor.execute('DELETE FROM records WHERE id = ?', (record\_id,))  
 if cursor.rowcount == 0:  
 return {'status': 'error', 'message': 'Record not found'}, 404  
 conn.commit()  
 return {'status': 'success', 'message': 'Record deleted'}  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = Record()  
 cherrypy.tree.mount(app)  
 cherrypy.config.update({'server.socket\_host': '127.0.0.1',  
 'server.socket\_port': 8080})  
 cherrypy.engine.start()  
 cherrypy.engine.block()