|  |  |  |
| --- | --- | --- |
| Лабораторная работа №4  Знакомство с облачными платформами PaaS | Выполнил | Доценко Е.В. |
| Группа | САПР 2.1 |
| Преподаватель | Цыганков В.А. |
| Дата |  |
| Оценка |  |

Цель работы - знакомство с облачными технологиями PaaS на примере создания простых веб-функций в cloud.yandex.ru.

Задачи:

1. Регистрация в облаке, рассмотрение возможностей

2. Рассмотреть пример создания и использования веб-функции

3. Создание и использование своих веб-функций

Предоставленные примеры:

Пример 1 – Простая облачная функция для расчета Евклидова расстояния на Python.

Пример 2 – Облачная функция для расчета меры Левенштейна на C#.

Пример 3 – Облачная функция для работы со списками фигур в Яндекс.cloud на Pythоn.

Модификации примеров:

1. Изменить источник данных по умолчанию для примера с фигурами.

В предоставленном коде источник данных по умолчанию брал данные из строки. Я поменяла его на источник данных с константными значениями для фигур. Причем в параметрах передается количество фигур и подсчитывается их общие площади и периметры.

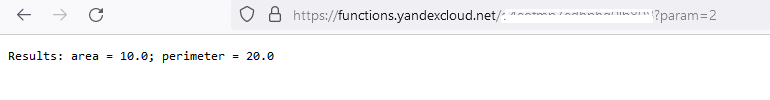


Рисунок 1 - Результат изменения источника данных

1. Изменить параметры обращения к функции / функциям

Я захотела изменить способ передачи данных в body. Вместо текущего формата "a=string1, b=string2", я изменила класс запроса:

public class Request {

public string httpMethod { get; set; }

public string a { get; set; }

public string b { get; set; }

}

А также я поменяла функцию FunctionHandler:

public Response FunctionHandler(Request request) {

// Вместо обработки разделенных по запятой данных

return new Response(200,

StringDistance.LevenshteinDistance(request.a, request.b).ToString());

}

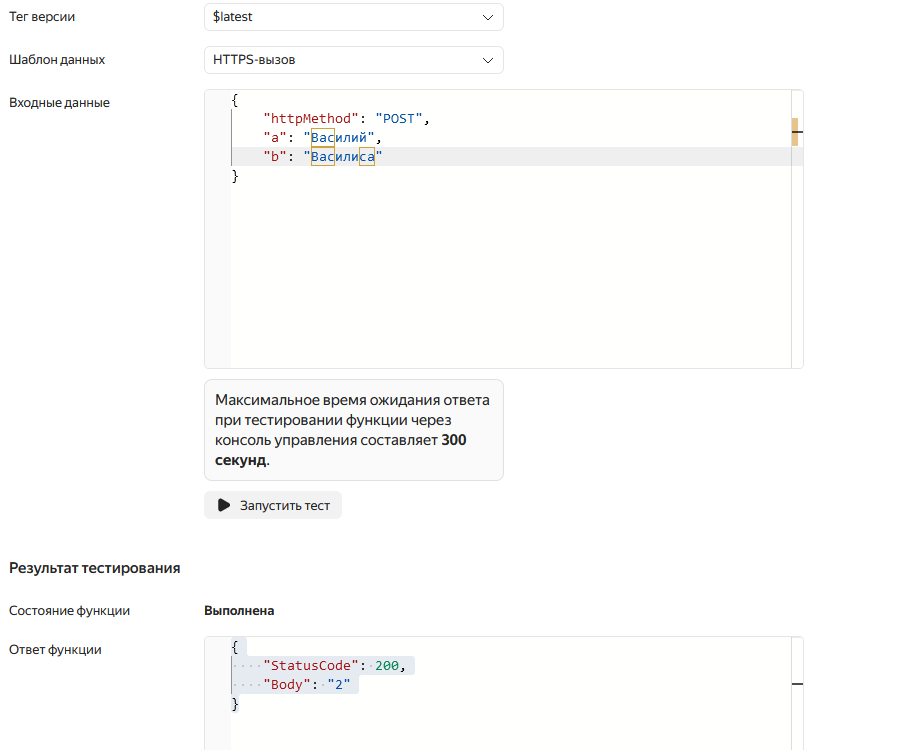


Рисунок 2 - Тест измененного способа передачи данных в тело запроса

1. Реализовать обращение к функции из клиента?

4) Реализовать свою простую веб-функцию.

Я реализовала веб-функцию калькулятора, который в зависимости от задаваемой операции выполняет расчёт.

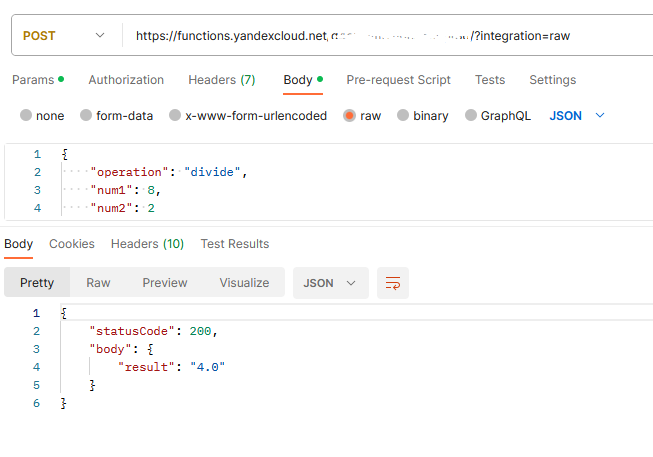


Рисунок 3 - Деление

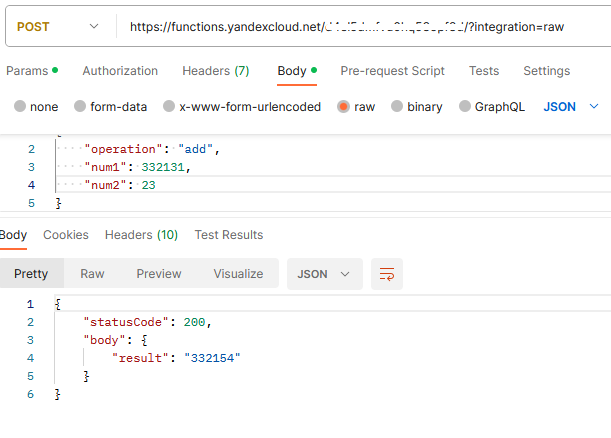


Рисунок 4 - Сложение

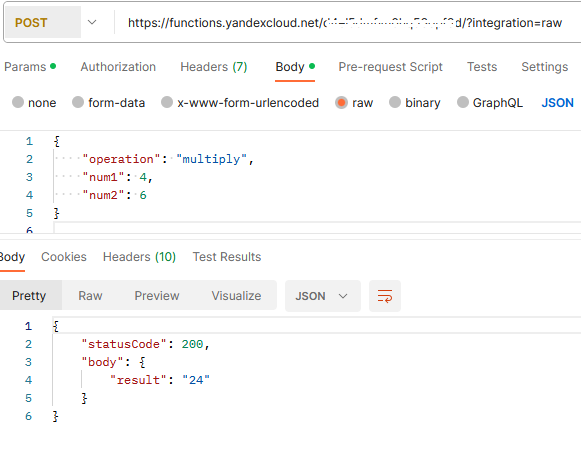


Рисунок 5 - Умножение

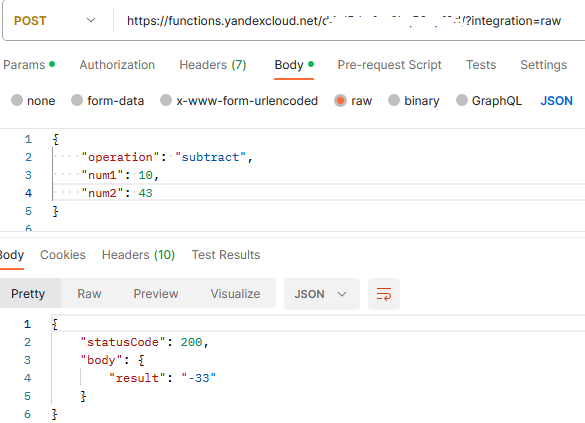


Рисунок 6 - Вычитание

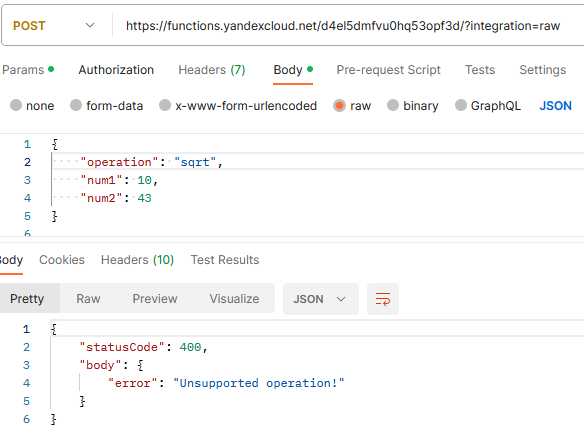


Рисунок 7 - Неподдерживаемая операция

Код функции

import json

def handler(event, context):

    # Получаем параметры из запроса

    operation = event.get('operation')

    num1 = event.get('num1')

    num2 = event.get('num2')

    result = None

    if operation == 'add':

        result = num1 + num2

    elif operation == 'subtract':

        result = num1 - num2

    elif operation == 'multiply':

        result = num1 \* num2

    elif operation == 'divide':

        if num2 != 0:

            result = num1 / num2

        else:

            return {

                'statusCode': 400,

                'body': {'error': 'Division by zero is impossible!'}

            }

    else:

        return {

            'statusCode': 400,

            'body': {'error': 'Unsupported operation!'}

        }

    return {

        'statusCode': 200,

        'body': {'result' : str(result)}

    }

Вывод: Выполнив данную работу, я познакомилась с облачными технологиями PaaS на примере создания простых веб-функций в cloud.yandex.ru