Инструкция

Contents

[Инструкция по запуску проекта на локальном компьютере. 1](#_Toc141907720)

[Структура папок. 4](#_Toc141907721)

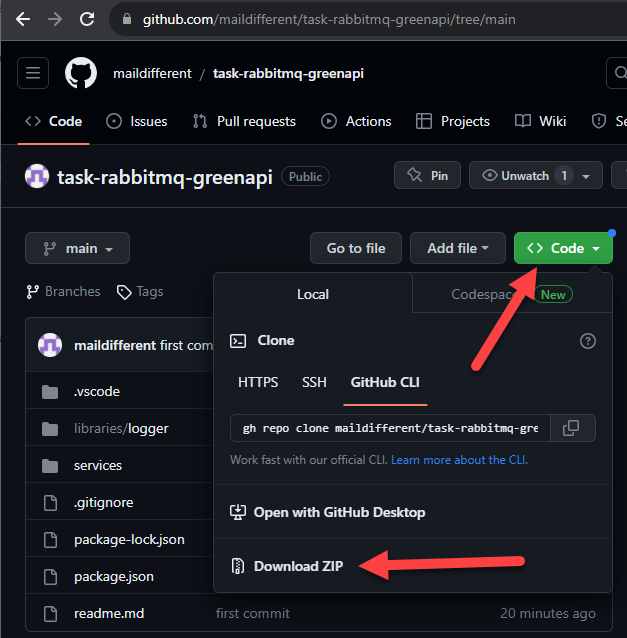
[Описание функциональности. 5](#_Toc141907722)

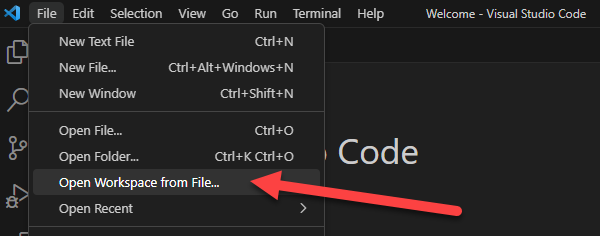
# Инструкция по запуску проекта на локальном компьютере.

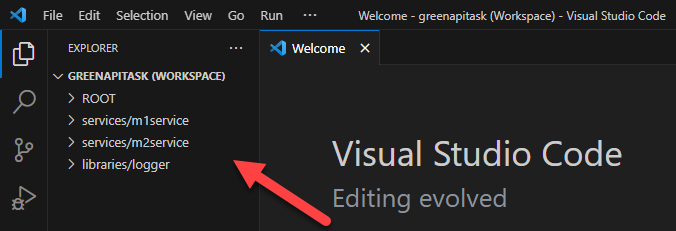
На компьютере должен быть установлен RabbitMQ.

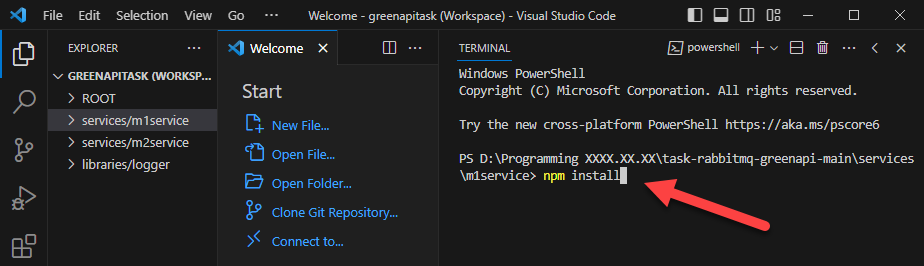
Установка стандартная со стандартными опциями.

https://www.rabbitmq.com/download.html

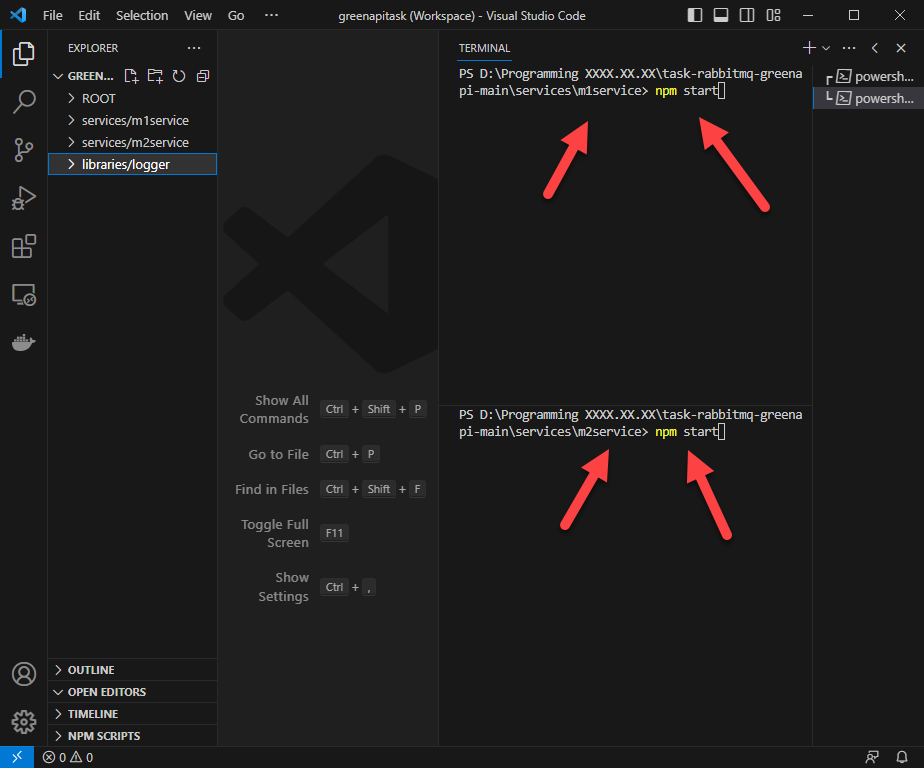
Загружаем и распаковываем проект по ссылке:  
https://github.com/maildifferent/task-rabbitmq-greenapi/  


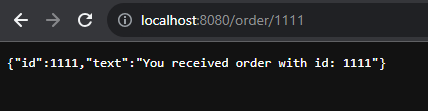
Через Visual Studio Code открываем workspace, который лежит в папке:  
\task-rabbitmq-greenapi-main\.vscode\greenapitask.code-workspace  


Workspace выглядит следующим образом:  


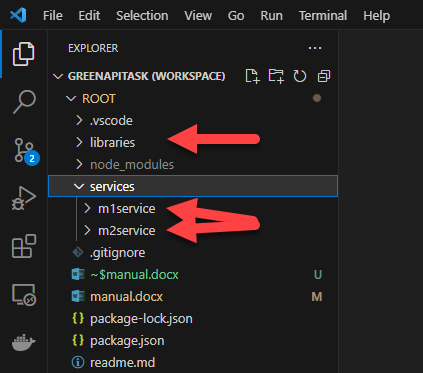
В каждой из четырёх папок:  
- root  
- m1service  
- m2service  
- logger  
Запускаем терминал и выполняем команду npm install  


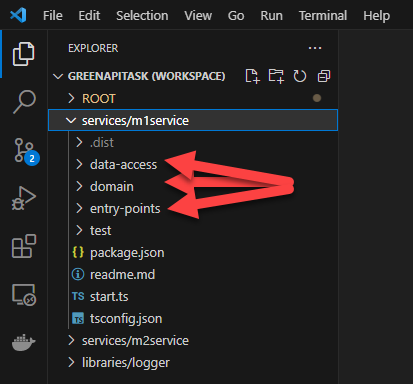
Аналогично в каждой из трёх папок:  
- m1service  
- m2service  
- logger  
Запускаем терминал и выполняем команду tsc

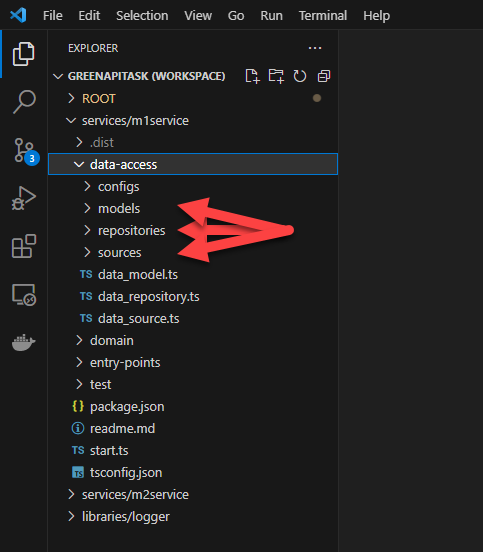
В двух терминалах запускаем команду NPM START в папках: m1service и m2service:  


В браузере проходим по ссылке  
http://localhost:8080/order/1111  
  
И видим ответ от сервера: «{"id":1111,"text":"You received order with id: 1111"}»  
Вместо цифр 1111 можно указать какие-то другие цифры.

# Структура папок.

Структура верхнего уровня разделена по бизнес компонентам (m1service, m2service).  
Отдельно выделена общая функциональность в папке libraries.  


В бизнес компонентах трёхуровневый подход:  
- entry-points – контроллеры  
- domain – логика  
- data-access – доступ к данным  


Доступ к данным разделен на уровни:  
- sources (источники: БД, другие сервисы…)  
- models  
- repositories (простые обёртки для использования моделей и источников)  
  
Сейчас в models и repositories по сути нет логики, показал их здесь просто ради правильности подхода.

Соответственно вся логика работы с внешними сервисами (в нашем случае с сервисом М2) находится в папке sources.

# Описание функциональности.

Обращение к сервису М1 по ссылке http://localhost:8080/order/1111 моделирует как будто мы запрашиваем данные заказа 1111.

Основная логика сервиса М1 находится в папке \services\m1service\data-access\sources

На каждый запрос формируется объект request, который сохраняется в MAP «requests» с уникальным ID.

Сервис М1 отправляет запрос данных в очередь client2server с указанием correlationId и callback очереди.

На callback очереди стоит listener, который обрабатывает полученные ответы.

Сервис М2 всего лишь возвращает объект в виде «{"id":1122,"text":"You received order with id: 1122"}», где 1122 – это номер id заказа, данные по которому должен вернуть сервис М2.