

УДК 004.43

**К. С. Левин, Р. А. Кузьменко**

студенты кафедры вычислительных систем и сетей

**А. В. Аксенов**

научный руководитель

## СОЗДАНИЕ И ПРИМЕНЕНИЕ МНОГОПОТОЧНОГО СЕРВИСА API С РАЗВЕРТЫВАНИЕМ НА СЕРВЕР В ДОКЕР-КОНТЕЙНЕРЕ

В современном мире существует огромное количество разных языков программирования для написания сложных и больших сервисов, готовых обслуживать ни одну тысячу человек одновременно. И зачастую очень трудно выбрать технологии, на которых будут держаться такие сервисы. Чтобы лучше разобраться в этой проблеме, необходимо изучить рынок, а затем сравнить, что больше подходит для выполнения конкретных критерий:

1. Большой размер проекта.
2. Отказоустойчивость при работе с данными.
3. Безопасность.
4. Скорость разработки и эксплуатации.
5. Большие нагрузки.

Комбинация технологий, таких как Go, Docker и PostgreSQL, выполняет поставленные критерии. Например, язык Go – это язык общего назначения с широкими возможностями и понятным синтаксисом, а также с надежной, хорошо документированной стандартной библиотекой, которая предоставляет удобные подходы к самой разработке. Поэтому он является идеальным языком для быстрого изучения и написания безопасных программ.

Docker – это открытая платформа для разработки, доставки и эксплуатации приложений. Он разработан для более быстрого выкладывания ваших приложений. С помощью docker вы можете отделить ваше приложение от вашей инфраструктуры и обращаться с ней, как управляемым приложением. Docker помогает выкладывать ваш код быстрее, быстрее тестировать, и уменьшить время между написанием и запуском кода. Все это делается с помощью легкой платформы контейнерной виртуализации, используя процессы и утилиты, которые помогают управлять и выкладывать приложения.

Для того чтобы создать и запустить контейнер используется подход Dockerfile, который использует обычный DSL с инструкциями для построения образов Docker.

На примере реализуемого проекта Dockerfile выглядит так:

```
#имя базового образа
FROM golang:1.10
MAINTAINER orderfood
#установка стороннего пакета govendor
RUN go get -u github.com/kardianos/govendor
#идем в папку
WORKDIR /go/src/github.com/orderfood/api_of
ADD .
#скачиваем зависимые пакеты проекта благодаря govendor
RUN govendor sync
# идем в папку
WORKDIR /go/src/github.com/orderfood/api_of/cmd/api
#сборка проета
RUN go build
#установка прав доступа
RUN chmod +x /go/src/github.com/orderfood/api_of/cmd/api/api
#конфигурация
ENTRYPOINT ["/go/src/github.com/orderfood/api_of/cmd/api/api"]
```

PostgreSQL – это СУБД, которая использует реляционную модель для своих баз данных и поддерживает стандартный язык запросов SQL. Она предоставляет множество различных возможностей, достаточно надежна и имеет хорошие характеристики по производительности. Работает практически на всех UNIX-платформах, включая UNIX-подобные системы, такие как FreeBSD и Linux. Ее можно применять на Windows NT Server и Windows 2000 Server. Кроме того, PostgreSQL свободно распространяется и имеет открытый исходный код.

Язык Go выбран после сравнения его с конкурирующими языками по пятибалльной системе (таблица).

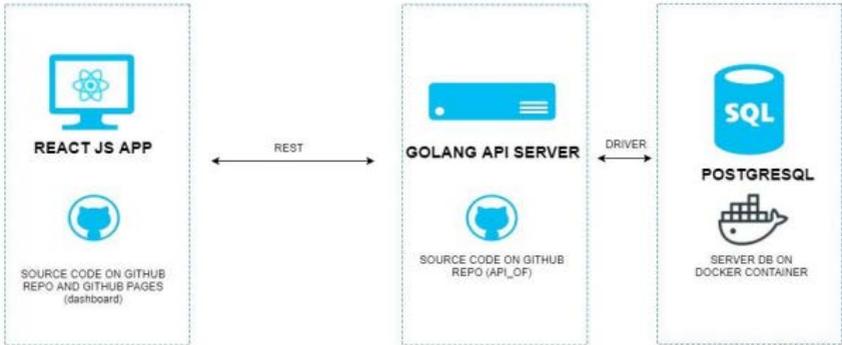


Рис. 1. Схема взаимодействия стека технологий

Таблица

Сравнение языков программирования

Показатели	Go	NodeJs	Python
Производительность	5	4	3
Удобство	5	5	5
Масштабируемость	5	4	4
Популярность	5	3	4
Совместимость	5	4	3

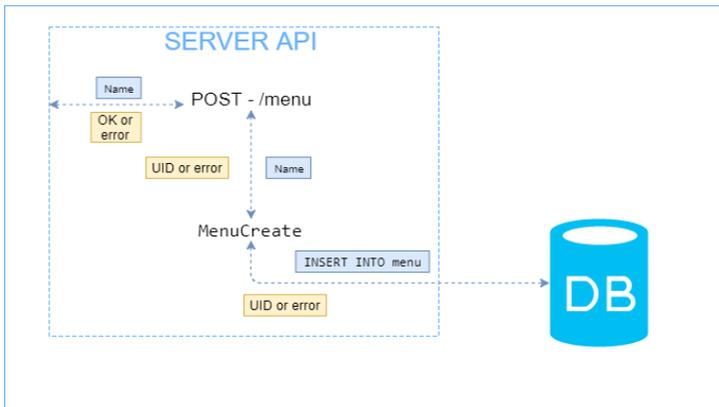


Рис. 2. Пример использования REST на серверной части проекта

Для большой производительности важно не только выбранная технология, но и способы, применяемые в передаче данных. Существует множество различных подходов к обмену данными между клиентом и сервером, обладающих различными свойствами, преимуществами и недостатками. Сегодня наиболее популярным является REST (Representational State Transfer).

REST был введен и определен в 2000 г. Roy Fielding в его докторской диссертации. REST – это архитектурный стиль для проектирования распределенных систем, обладающий несколькими принципами:

1. Ресурсы позволяют легко понять структуру каталогов URI.
2. Представления передают JSON или XML в качестве представления данных объекта и атрибутов.
3. Сообщения используют HTTP методы явно (например, GET, POST, PUT и DELETE).
4. Отсутствие состояния взаимодействий не сохраняет контекст клиента на сервере между запросами.

В результате выбор такого стека технологий позволяет легко и быстро создать требуемый проект, удовлетворяющий всем критериям.

#### **Библиографический список**

1. Webber J., Parastatidis S., Robinson I. REST in Practice. Hypermedia and Systems Architecture. O'Reilly Media, 2010.
2. Как и для чего использовать Docker. URL: <https://guides.hexlet.io/docker/> (дата обращения: 12.04.2018).
3. An Introduction to Programming in Go. URL: <https://www.golang-book.com/books/intro> (дата обращения: 12.04.2018).