Лабораторная работа №3

GCP: App Engine

# Цель работы

Знакомство с App Engine, Pub/Sub.

# Задание для выполнения

Напишите микросервис **MS3**, реализующий функциональность биллинговой системы, в которую приложение **MS2** будет отправлять данные каждый раз, когда происходит «Регистрация события».

Связь между двумя сервисами – с использованием Pub/Sub.

Задача микросервиса **MS3** – обработать/оценить полученные данные *(случайным образом, не надо реальный биллинг строить)* и вернуть в MS2 результат: положительный или отрицательный – разрешено действие или нет. **MS2** в зависимости от этого должно произвести некие действия, например, отменить заказ или удалить товар из корзины.

# Что такое App Engine?

Прежде всего обозначим, что такое Google App Engine (GAE). Это платформа для запуска ваших веб-приложений, таких как Heroku. Но они немного отличаются, если присмотреться. Heroku — универсальная платформа для облачных вычислений, как и AWS. Но, опять же, они разные.

Google запустил GAE в 2008 году, когда облачные вычисления еще были в зачаточном состоянии. Amazon был впереди, когда Google начал сдавать в аренду свою IT-инфраструктуру в 2006 году. Но с GAE Google очень рано предложил сложную платформу как услугу (PaaS), которая соответствовала Amazon со своим сервисом Elastic Beanstalk в 2011. Что особенного в GAE?

GAE — полностью управляемая платформа приложений. Я не знаю платформы, которая приближается к полному пакету GAE: управление журналом, доставка почты, масштабирование, memcache, манипуляция изображениями, распределенные задания Cron, балансировка нагрузки, управление версиями, очередь задач, поиск, анализ производительности, облачная отладка, сеть доставки контента — и это без упоминания о вспомогательных службах, появившихся в облаке Google, таких как SQL, BigQuery, хранение файлов… список можно продолжить.

Используя Google App Engine, вы можете запустить приложение поверх лучшей в мире (возможно) инфраструктуры. Кроме того, вы получаете функциональность «из коробки», которая потребует по крайней мере, дюжины дополнений от третьих сторон на Heroku или нескольких недель установки. Это призыв GAE.

Достойные упоминания приложения, которые работают на GAE, включают Snapchat и Khan Academy.

# SDK

Каждое приложение должно применять специальный SDK (Software Development Kit) для использования API, предлагаемых GAE. SDK огромен. Например, загрузка Java SDK составляет примерно 190 МБ. Некоторые JAR не нужны в большинстве случаев использования, а некоторые нужны только во время разработки.

SDK — не только мост в мир Google App Engine, он также служит симулятором локальной машины. Практически для каждого сценария использования GAE API есть шаблон, который вы можете использовать как основу для вашего проекта.Что это означает? Во-первых, при запуске приложения локально вы будете очень близки к тому, как оно будет вести себя на производстве. Во-вторых, вы можете легко написать тесты интеграции с API. Как правило, это может вас далеко завести; разница между производственным и заглушенным поведением довольно невелика.

Для работы вам придется скачать и установить SDK. Также для используемого вами языка необходимы клиентские библиотеки. Например, для PHP они могут быть установлены одним вызовом **composer require google/cloud**

# Pub/Sub

Издатель-подписчик (англ. publisher-subscriber или англ. pub/sub) — поведенческий шаблон проектирования передачи сообщений, в котором отправители сообщений, именуемые издателями (англ. publishers), напрямую не привязаны программным кодом отправки сообщений к подписчикам (англ. subscribers). Вместо этого сообщения делятся на классы и не содержат сведений о своих подписчиках, если таковые есть. Аналогичным образом подписчики имеют дело с одним или несколькими классами сообщений, абстрагируясь от конкретных издателей.

Во многих реализациях шаблона издатель-подписчик издатель отправляет сообщения посреднику, который может быть брокером сообщений или шиной. В таком случае подписчики регистрируют подписку с этим брокером, осуществляющим фильтрацию. Брокер, как правило, осуществляет хранение сообщений и пересылку для маршрутизации сообщения от издателя к подписчику. Кроме того, брокер может устанавливать приоритеты сообщениям в очереди сообщений перед их маршрутизацией.

Google Pub/Sub - это служба асинхронного обмена сообщениями, которая отделяет службы, создающие события, от служб, обрабатывающих события.

Вы можете использовать Pub/Sub как промежуточное ПО, ориентированное на обмен сообщениями, или как прием и доставку событий для конвейеров потоковой аналитики.

Pub/Sub предлагает надежное хранилище сообщений и доставку сообщений в реальном времени с высокой доступностью и стабильной производительностью в любом масштабе.