API здесь, здесь все: Swift, REST, Mongo, JSON, Node и Push Notifications)))

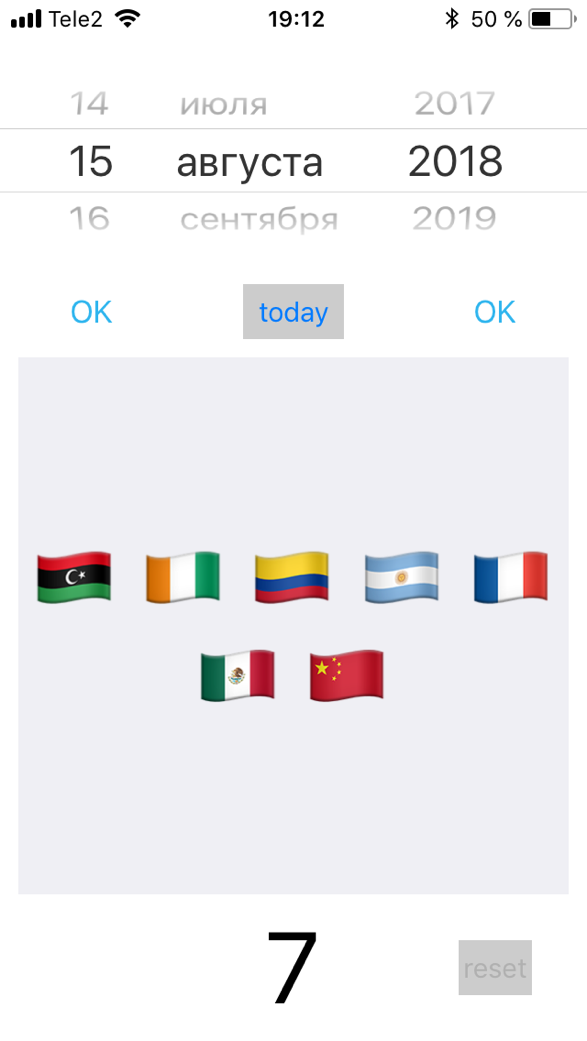
Проект YappM задуман с целью создания приложения для информирования разработчика о количестве и географии пользователей App Store, установивших на свои устройства, созданные разработчиком приложения. Информация доступна как из самого приложения, так и путем доставки push notifications. В настоящее время дорабатывается версия для Apple Watch.

За основу был взят сервис Яндекса – AppMetrica Logs API <https://tech.yandex.ru/appmetrica/doc/mobile-api/concept/about-docpage/>

<https://tech.yandex.ru/appmetrica/doc/mobile-api/logs/request-procedure-docpage/#request-data-processing>

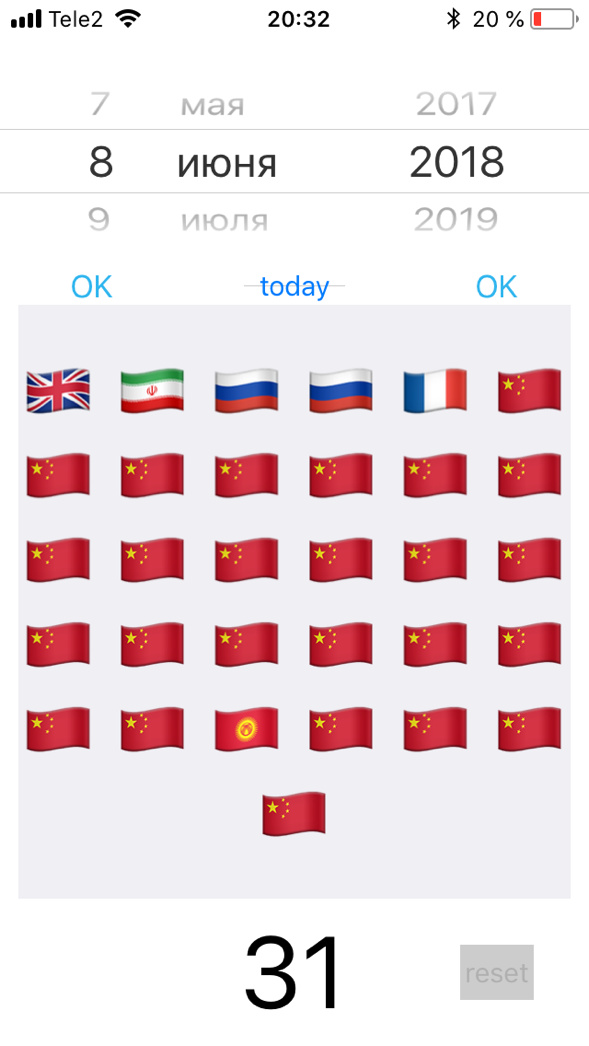
YappM настроен на запрос и ответ по двум моим опубликованным приложениям в App Store, кстати вот они: (<https://itunes.apple.com/ru/app/binatrix/id1296545616?l=ru&ls=1&mt=8> и <https://itunes.apple.com/ru/app/hexastar/id1327719099?mt=8>). Данные получаем за сутки (с 00-00-00 до 23-59-59). Из-за любви к географии меня интересует только страна, где произошла установка приложений. Результат возвращается в виде JSON файла с двухбуквенными кодами стран по ISO.

Конвертируем коды в флаги стран (привет школа!),



помня о том, что символ флага в эмодзи состоит как раз из сочетания специальных букв в Unicode (A + U в примере Австралии).

Мои приложения, к сожалению, не так популярны, поэтому география пользователей не систематизируется (страна – количество установок), а информируется сам факт единичной загрузки в стране:

 О как бывает!

Реализация push notifications на телефоне выполнена с помощью этих руководств (англоязычная и русская версии):

<https://www.raywenderlich.com/584-push-notifications-tutorial-getting-started>

<https://swiftbook.ru/post/tutorials/tutorial-push-notifications/>

При запуске приложения, получаем токен устройства, который сохраняем в телефоне, а так же посылаем его с помощью REST для сохранения в базе MongoDB, развернутой на сервере Linux. В случае смены токена, наша база будет оставаться актуальной, потому что мы отправляем его значение при каждом запуске приложения.

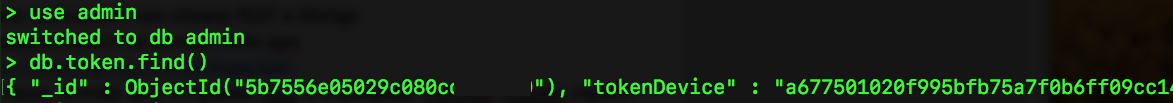
MongoDB устанавливаем по данной инструкции:

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

В коллекции в базе создаем запись и запоминаем номер ключа:

<http://gearmobile.github.io/mongodb/mongodb-databases/>

<http://gearmobile.github.io/mongodb/mongodb-document-create/>



Установим всё необходимое и напишем скрипт для REST:

<https://habr.com/company/ruvds/blog/321104/>

Теперь с помощью запросов и ключа записи в базе, у нас всегда будет доступ с актуальному токену устройства.

Приступаем к главному! Создадим скрипт используя node-apn на сервере:

<https://www.npmjs.com/package/apn>

Это наша основа для отправки push notifications с сервера. Скрипт повторяет запросы так же как и в реализации в телефоне, но с периодичностью 18 минут сохраняя ответы. Раз в 20 минут скрипт сравнивает данные и, если есть изменения по сравнению с прошлой попыткой, посылает нам push notifications, для которого берет с помощью запроса REST из MongoDB токен нашего устройства.

Для автозагрузки и мониторинга работы скриптов используем PM2:

<https://pm2.io/doc/en/runtime/overview/?utm_source=pm2&utm_medium=website&utm_campaign=rebranding>

Postman поможет управлять базой при помощи запросов:

<https://www.getpostman.com>

Не все статьи и руководства современны, не сразу все заработает. Данное описание - для общего представления о проекте. Конкретная реализация работает, требует доработки только часть с Apple Watch.

