Собираем свой Google Static Maps

Миша Баранов, PiterJS #24

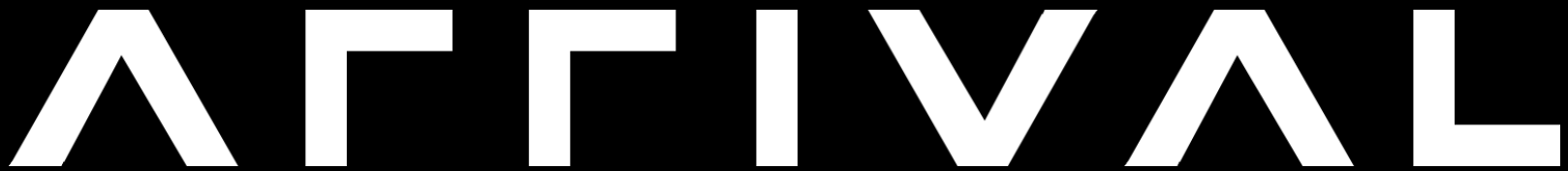
Михаил Баранов

Старший разработчик
в ARRIVAL Software
DevOPS в Веб-стандартах

Михаил Баранов (@_h4_) | Twitter

The latest Tweets from Михаил Баранов (@_h4_). Сказочный долбоёб. Sankt-Peterburg

twitter.com


ARRIVAL


ARRIVAL


20
GB
MX9


ARRIVAL

Постановка задачи

- Мониторинг грузовиков
- Оперативный мониторинг грузовиков
- Чтобы само всё рассказывало

Решение

- Телеграм-бот
  - (Сервис, пока ещё не запрещенный на территории Российской Федерации)

Инструменты

- GeoJSON
- Headless Chrome
- Puppeteer
- Docker

DEV - ARRIVAL Techical Bot

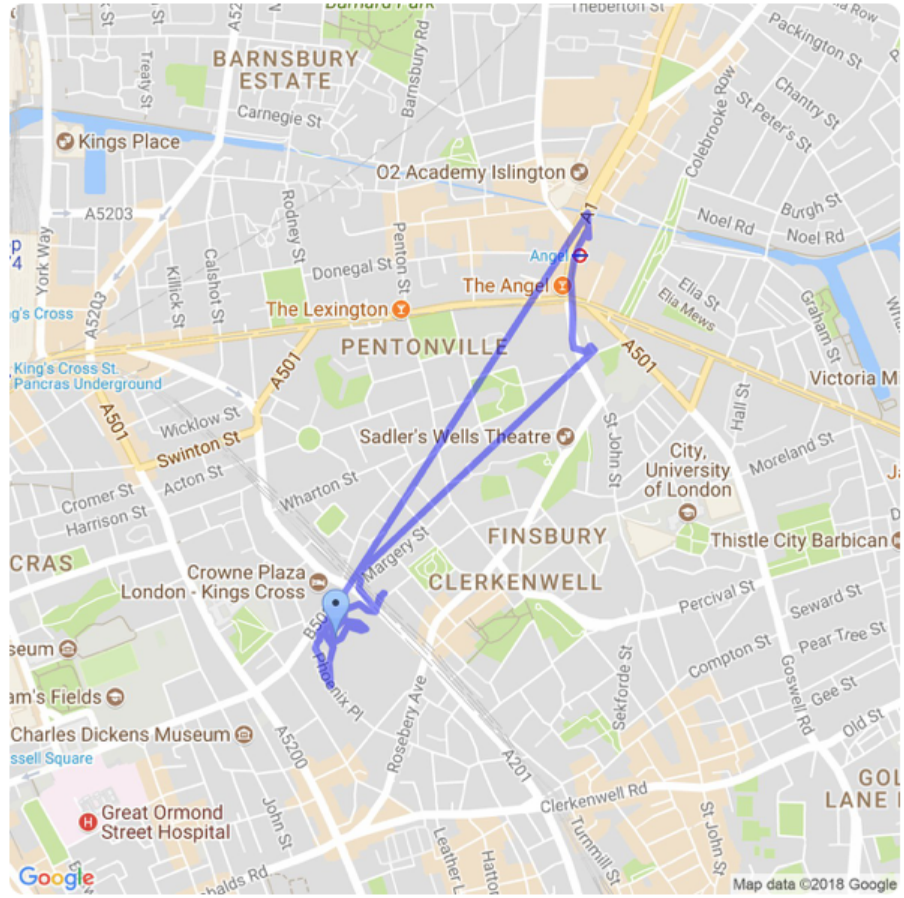
Daily Report 09 Apr 2018

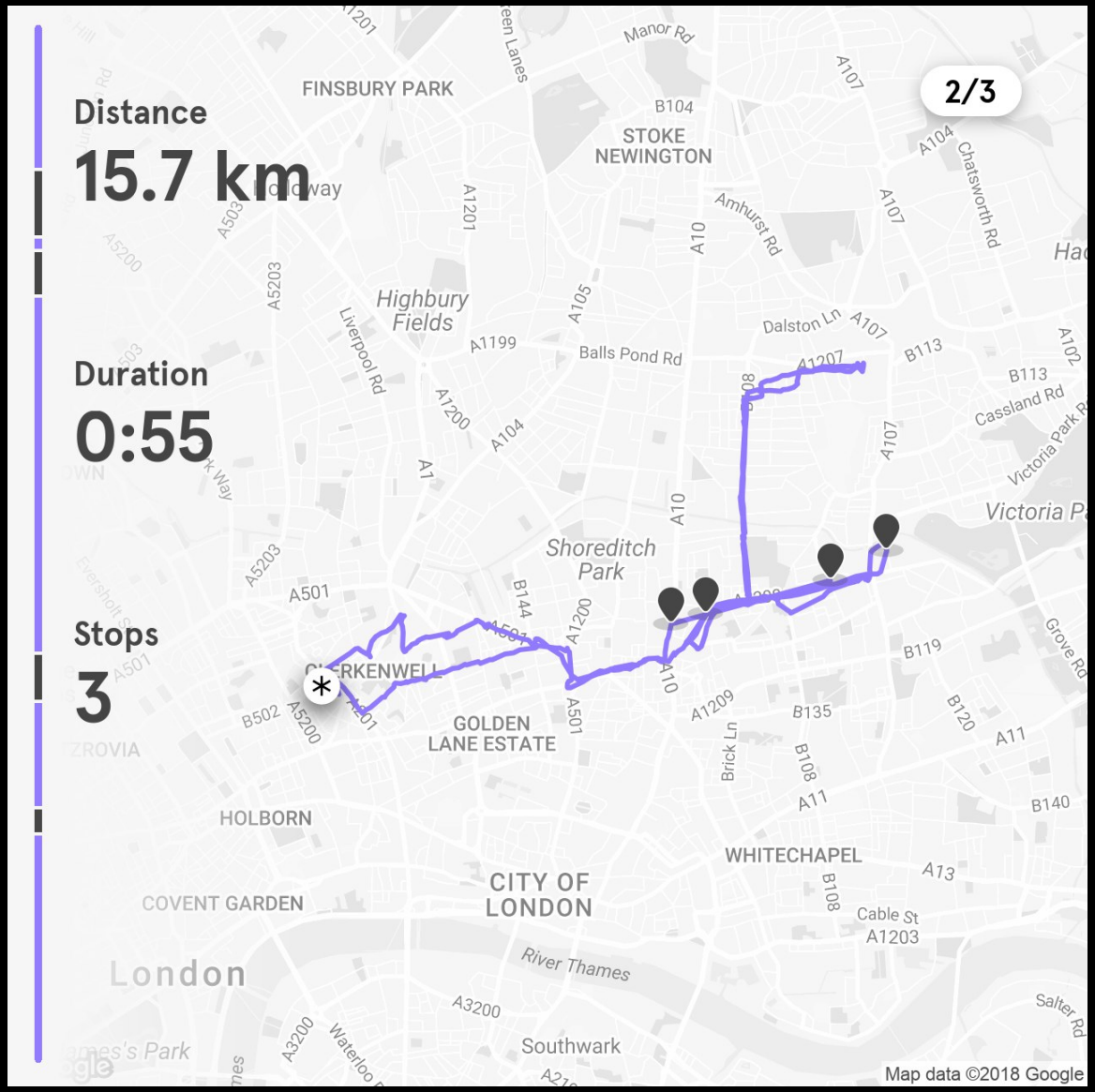
Vehicle: D2.2 (Royal Mail)
Number of Trips: 1
Distance Driven: 5.83 km
Driving Duration: 00:18
Driving Cons: 708 Wh/km
Total Energy: 9.5 kWh
Total Cost: 1.15 £

Trip: 1
Number of Stops: 0
Distance Driven: 5.83 km
Driving Duration: 00:18
Driving Cons: 708 Wh/km
Total Energy: 5.7 kWh
Total Cost: 0.69 £

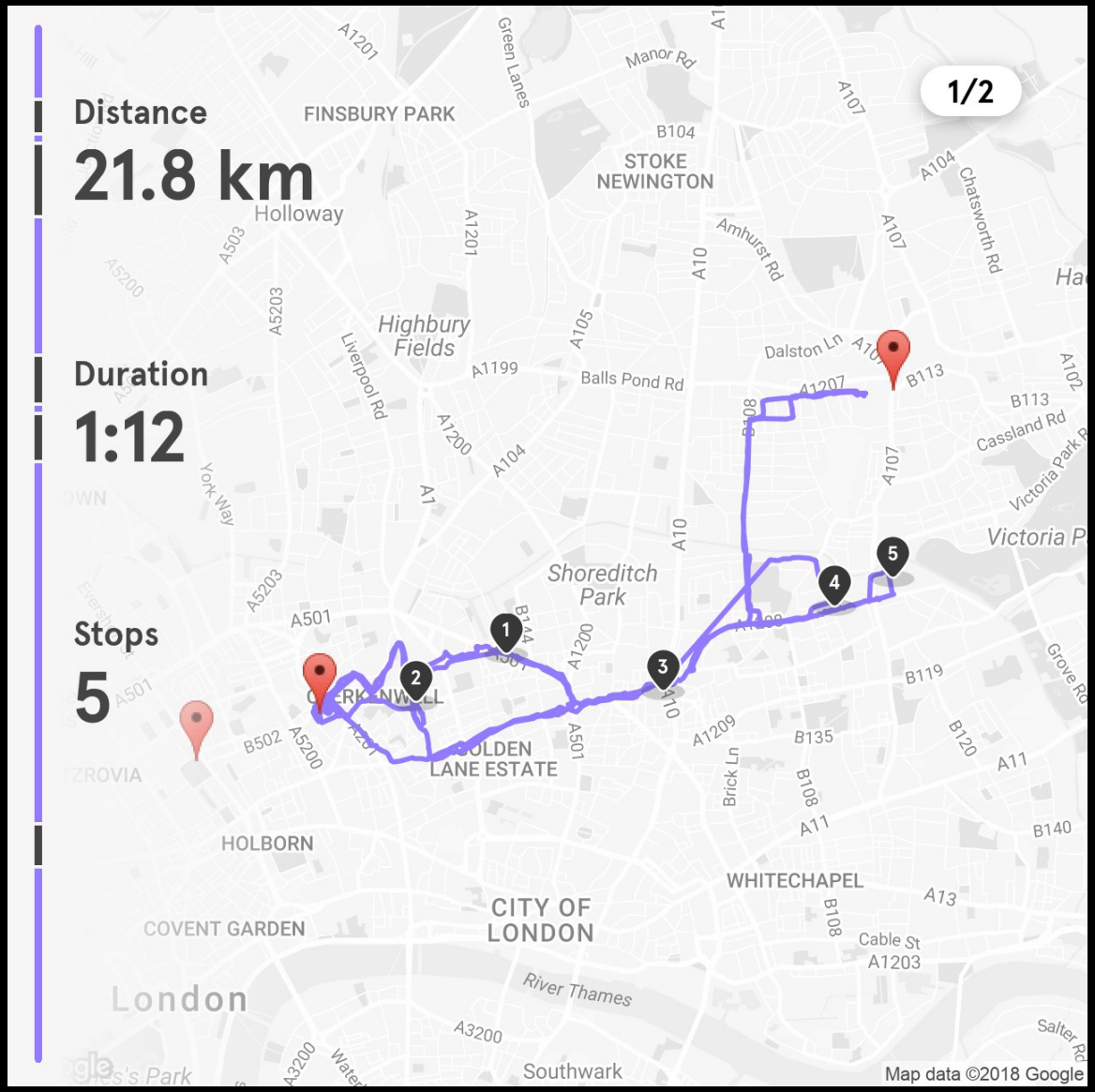
Errors:
- BMS error (x15)

Details in Cloud:
- Daily Tracks
- Charts & Errors




Distance
15.7 km
Duration
0:55
Stops
3
2/3
FINSBURY PARK
STOKE NEWINGTON
Highbury Fields
Shoreditch Park
CLERKENWELL
GOLDEN LANE ESTATE
HOLBORN
COVENT GARDEN
CITY OF LONDON
WHITECHAPEL
London
Southwark
River Thames
Balls Pond Rd
Dalston Ln
Cassland Rd
Victoria Park Rd
Amhurst Rd
Chatsworth Rd
Liverpool Rd
Brick Ln
Cable St
Map data ©2018 Google


Distance
16.5 km
Duration
0:54
Delivery Stops
6
1/1
FINSBURY PARK
STOKE NEWINGTON
Highbury Fields
Shoreditch Park
GOLDEN LANE ESTATE
HOLBORN
COVENT GARDEN
CITY OF LONDON
WHITECHAPEL
London
River Thames
Southwark
Victoria P
Map data ©2018 Google
Terms of Use
Report a map error


Distance
21.8 km
Duration
1:12
Stops
5
1/2
FINSBURY PARK
STOKE NEWINGTON
Holloway
Highbury Fields
Balls Pond Rd
Dalston Ln
Cassland Rd
Victoria Park
Shoreditch Park
Victoria P
CLERKENWELL
GOLDEN LANE ESTATE
HOLBORN
COVENT GARDEN
CITY OF LONDON
WHITECHAPEL
London
River Thames
Southwark
Map data ©2018 Google

Custom Icons

Rather than use Google's marker icons, you are free to use your own custom icons instead. Custom icons are specified using the `icon` descriptor in the `markers` parameter. For example:

`markers=icon:`*`URLofIcon`*`|`*`markerLocation`*

Specify the `icon` using a URL (which should be URL-encoded). You can use URLs created by URL-shortening services such as `https://goo.gl`. Most URL-shortening services have the advantage of automatically encoding URLs.

You may specify an anchor point for the custom icon. The anchor point sets how the icon is placed in relation to the specified `markers` locations. By default, the anchor point of a custom icon is the bottom center of the icon image. You can specify a different anchor point using the `anchor` descriptor in conjunction with your `icon`. Set the `anchor` as an x,y point of the icon (such as `10,5`), or as a predefined alignment using one of the following values: `top`, `bottom`, `left`, `right`, `center`, `topleft`, `topright`, `bottomleft`, or `bottomright`. For example:

`markers=anchor:bottomright|icon:`*`URLofIcon`*`|`*`markerLocation1`*`|`*`markerLocation2`*

You can use up to five unique custom icons per request. This limitation does not mean that you are limited to only 5 marked locations on your map. Each unique icon may be used with more than one `markers` location on your map.

Icon format:

- Icon images may be in PNG, JPEG or GIF formats, though PNG is recommended.
- Icons may be up to 4096 pixels maximum size (64x64 for square images).

Custom Icon Examples

Example 1 creates custom icons and positions the icons using anchors.

```
http://maps.googleapis.com/maps/api/staticmap?&size=600x400&style=visibility:on
```

`markers=icon:URLofIcon|markerLocation`

Specify the `icon` using a URL (which should be URL-encoded). You can use URLs created by URL-shortening services such as `https://goo.gl`. Most URL-shortening services have the advantage of automatically encoding URLs.

You may specify an anchor point for the custom icon. The anchor point sets how the icon is placed in relation to the specified `markers` locations. By default, the anchor point of a custom icon is the bottom center of the icon image. You can specify a different anchor point using the `anchor` descriptor in conjunction with your `icon`. Set the `anchor` as an x,y point of the icon (such as `10,5`), or as a predefined alignment using one of the following values: `top`, `bottom`, `left`, `right`, `center`, `topleft`, `topright`, `bottomleft`, or `bottomright`. For example:

`markers=anchor:bottomright|icon:URLofIcon|markerLocation1|markerLocation2`

You can use up to five unique custom icons per request. This limitation does not mean that you are limited to only 5 marked locations on your map. Each unique icon may be used with more than one `markers` location on your map.

Icon format:

- Icon images may be in PNG, JPEG or GIF formats, though PNG is recommended.
- Icons may be up to 4096 pixels maximum size (64x64 for square images).

Custom Icon Examples

Example 1 creates custom icons and positions the icons using anchors.

You can use up to five unique custom icons per request. This limitation does not mean that you are limited to only 5 marked locations on your map. Each unique icon may be used with more than one `markers` location on your map.

Алгоритм

- Отправляем geojson
- Рисуем карту через Google Maps JS API
- Делаем скриншот
- Отправялем байты обратно

GeoJSON

- Стандартный формат представления геоданных
- RFC 7946 (2016)
- Легко стилизуеся в GoogleMaps API

```
{
  "type": "FeatureCollection",
  "features": [ {
    "type": "Feature",
    "geometry": {
      "type": "Point", "coordinates": [ 0.0719, 51.5347 ] },
    "properties": {
      "type": "vehicleMarker",
      "url": "https://goo.gl/W194zv",
      "size": [ 128, 128 ]
} ] }
```

```
01. function setFeatureStyle(feature) {
02.   const type = feature.getProperty('type');
03.   switch (type) {
04.     case 'vehicleMarker':
05.       const image = {
06.         url: feature.getProperty('url'),
07.         size: new google.maps.Size(...feature.getProperty('siz
08.       };
09.       return {icon: image,optimized: false};
10.   }
11. }
```

```
01. function initMap(root) {
02.   const map = new google.maps.Map(root);
03.   fetch(`http://localhost:8080/map_data?id=${resourceId}`)
04.     .then((response) => response.json())
05.     .then((data) => {
06.       const features = map.data.addGeoJson(data);
07.       map.data.setStyle(setFeatureStyle);
08.     );
09. }
```

Headless Chrome

```
alias chrome="/Applications/Google\
Chrome.app/Contents/MacOS/Google\ Chrome"
```

Headless Chrome

```
chrome \
  --headless \
  --disable-gpu \
  --window-size=1200,800 \
  --hide-scrollbars \
  --screenshot \
  https://piterjs.org/
```

JS

Подать завку на доклад

Headless Chrome

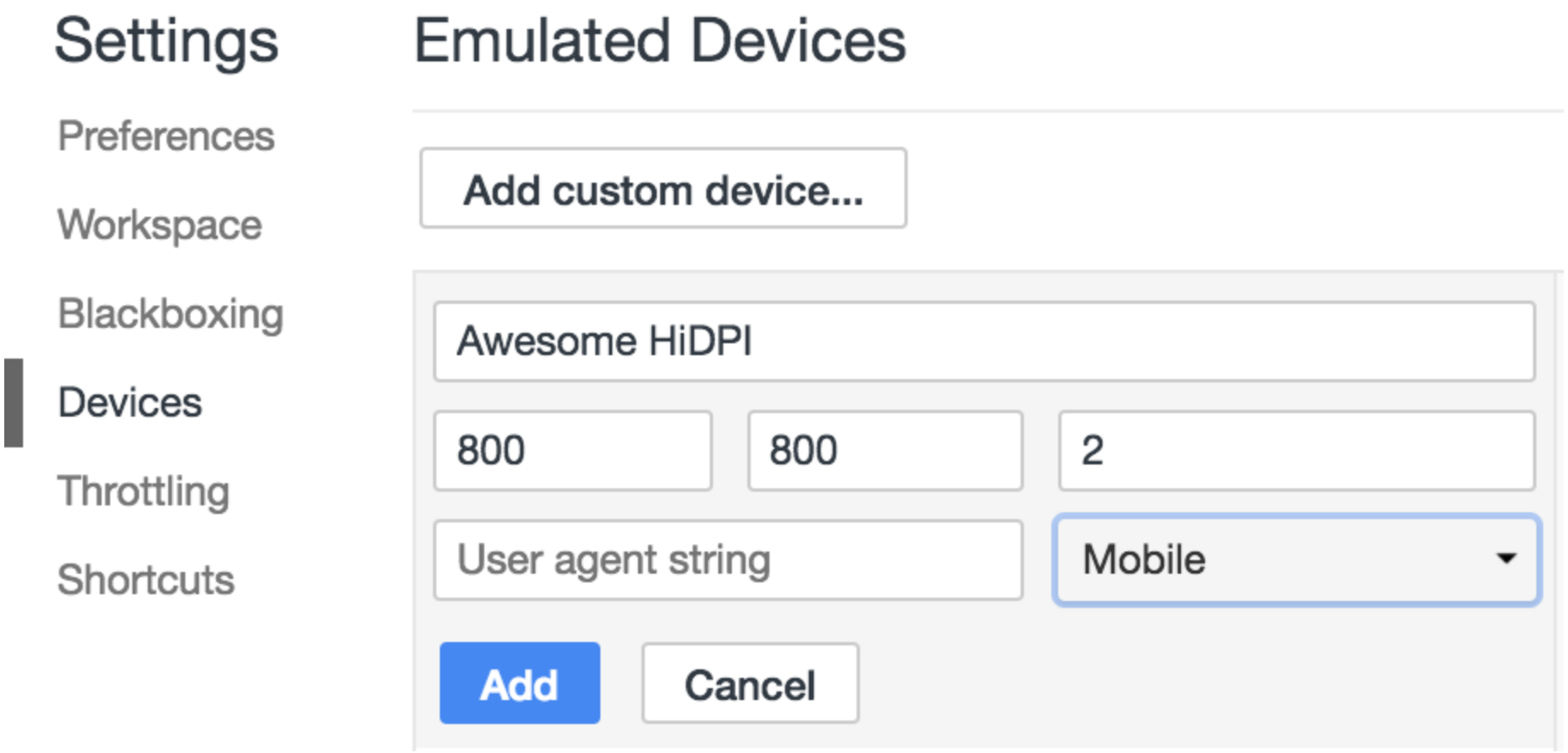
- Быстрый старт
- «Документация» к Api

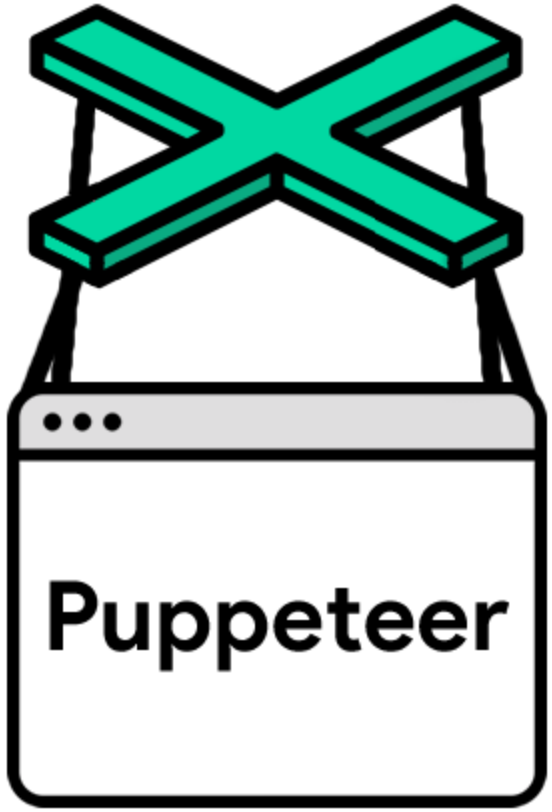
List of Chromium Command Line Switches

There are lots of command lines which can be used with the Google Chrome browser. Some change behavior of features, others are for debugging or experimenting. This page lists the available switches including their conditions and descriptions. Last automated update occurred on **2018-04-09**.

Condition	Explanation
--	Report pseudo allocation traces. Pseudo traces are derived from currently active trace events.
--/prefetch:1[1]	/prefetch:# arguments to use when launching various process types. It has been observed that when file reads are consistent for 3 process launches with the same /prefetch:# argument, the Windows prefetcher starts issuing reads in batch at process launch. Because reads depend on the process type, the prefetcher wouldn't be able to observe consistent reads if no /prefetch:# arguments were used. Note that the browser process has no /prefetch:# argument; as such all other processes must have one in order to avoid polluting its profile. Note: # must always be in [1, 8]; otherwise it is ignored by the Windows prefetcher.
--/prefetch:2[1]	*No description*
--/prefetch:3[1]	*No description*
--/prefetch:4[1]	*No description*

Retina


Settings
Preferences
Workspace
Blackboxing
Devices
Throttling
Shortcuts
Emulated Devices
Add custom device...
Awesome HiDPI
800
800
2
User agent string
Mobile
Add
Cancel

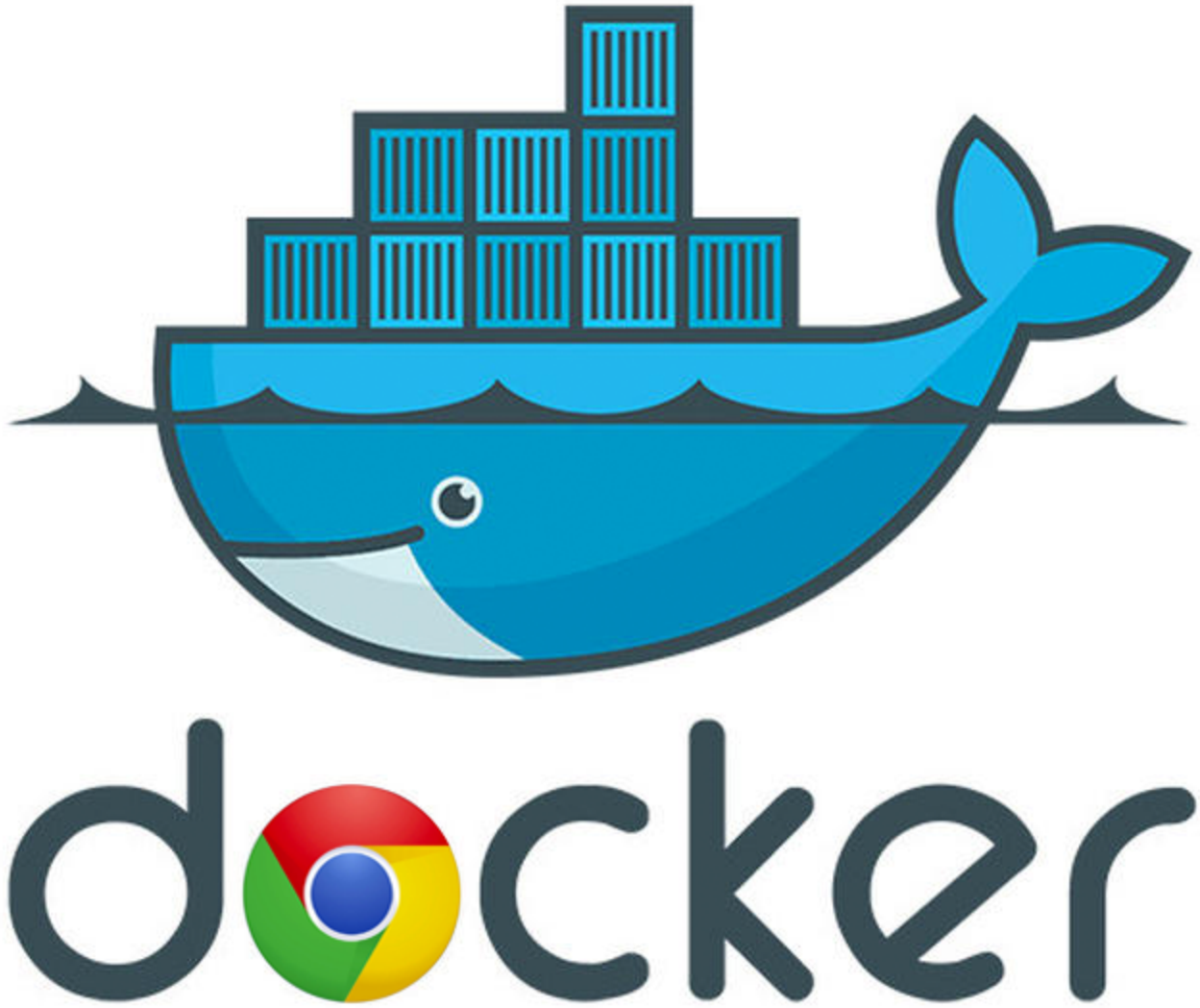

Puppeteer

Puppeeter

- `npm i puppeteer`
- Developed by the Chrome team
- pyppeteer

```
page = await browser.newPage()
await page.setViewport({
    "width": width,
    "height": height,
    "deviceScaleFactor": 2 if hidpi else 1,
})
```

```
01. await page.goto(url, {
02.    "waitUntil": "networkidle0",
03. })
04. with tempfile.TemporaryDirectory() as tmpdirname:
05.     image = await page.screenshot({
06.       "path": f'{tmpdirname}/image.png'
07.     })
08. await page.close()
```


docker

Зачем вообще Docker

- Одинаковое окружение у разработчков, тестировщиков и на проде
- Работает на всех платформах
- Минимальный оверхед
  - Linux — стандартые пакеты
  - Mac — xhyve
  - Win — Hyper V / VirtualBox
- Изоляция процессов

Контейнер

- Изолированный процесс
- Умирает после остановки контейнера
- Можно примонтировать каталоги с хоста
- Можно открыть порты
- ~~Можно грабить корованы~~

Образ

- Код приложения
- Зависимости приложения
- Операционная система
- Контейнер запускается из образа

Официальные образы всего

Dockerfile

- Описание, как собрать образ
- Базовый образ
- Как ставить зависимости
- Откуда брать код

Документация

Docker Compose

- Запуск нескольких связанных контейнеров
- Контейнеры используют общую сеть
- Контейнеры получают имена

```
01. FROM python:alpine3.7
02. ADD requirements /opt/requirements
03. COPY apk/chromium-64.0.3282.168-r0.apk /
04. COPY src /opt/app
05. RUN apk add --no-cache chromium && \
06.     apk add --allow-untrusted  /chromium-64.0.3282.168-r0.apk
07.     rm /chromium-64.0.3282.168-r0.apk && \
08.     pip install --no-cache-dir -r /opt/requirements/base.txt
09. CMD ["python3", "/opt/app/main.py"]
```

Сборка образа

```
01. docker build -t maps-generator .
02. docker images
03. docker push maps-generator
```

Запуск контейнера

```
01. docker run --rm -p 8080:8080 maps-generator
02. docker run -d -p 8080:8080 maps-generator
03. docker run -v $(pwd)/src:/opt/app maps-generator
04. docker run --name my-maps maps-generator
05. docker ps
06. docker ps -a
07. docker exec -it my-maps bash
08. docker kill my-maps
```

Итого

- Число объектов на карте ⟶ ∞
- Больший контроль за результатом
- Сделали всё за два дня

Демка

- Сайтшот-бот
- Репозиторий с кодом

Спасибо!

Миша Баранов

- mkhl.brnv.ru
- mkhl@brnv.ru
- github.com/h4