

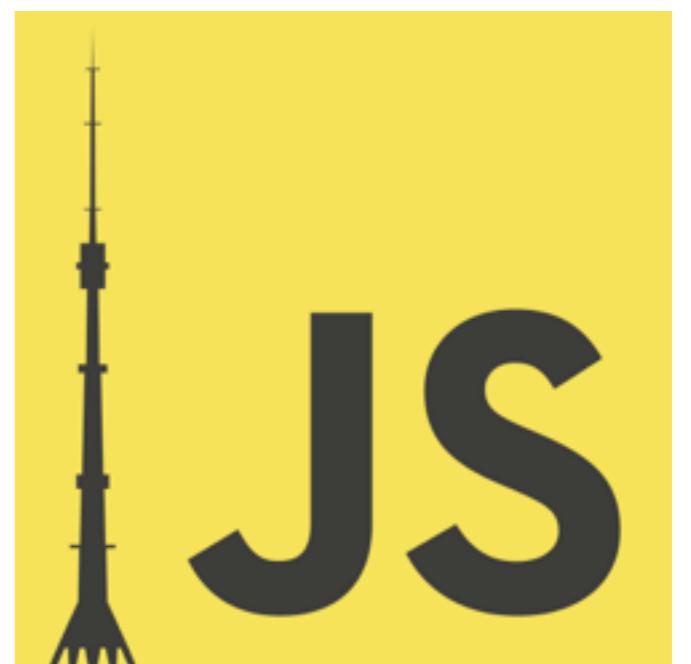
Яндекс

Яндекс

Карты

Promise me an Image...

Антон Корзунов, MoscowJS ^{#33}, 2016

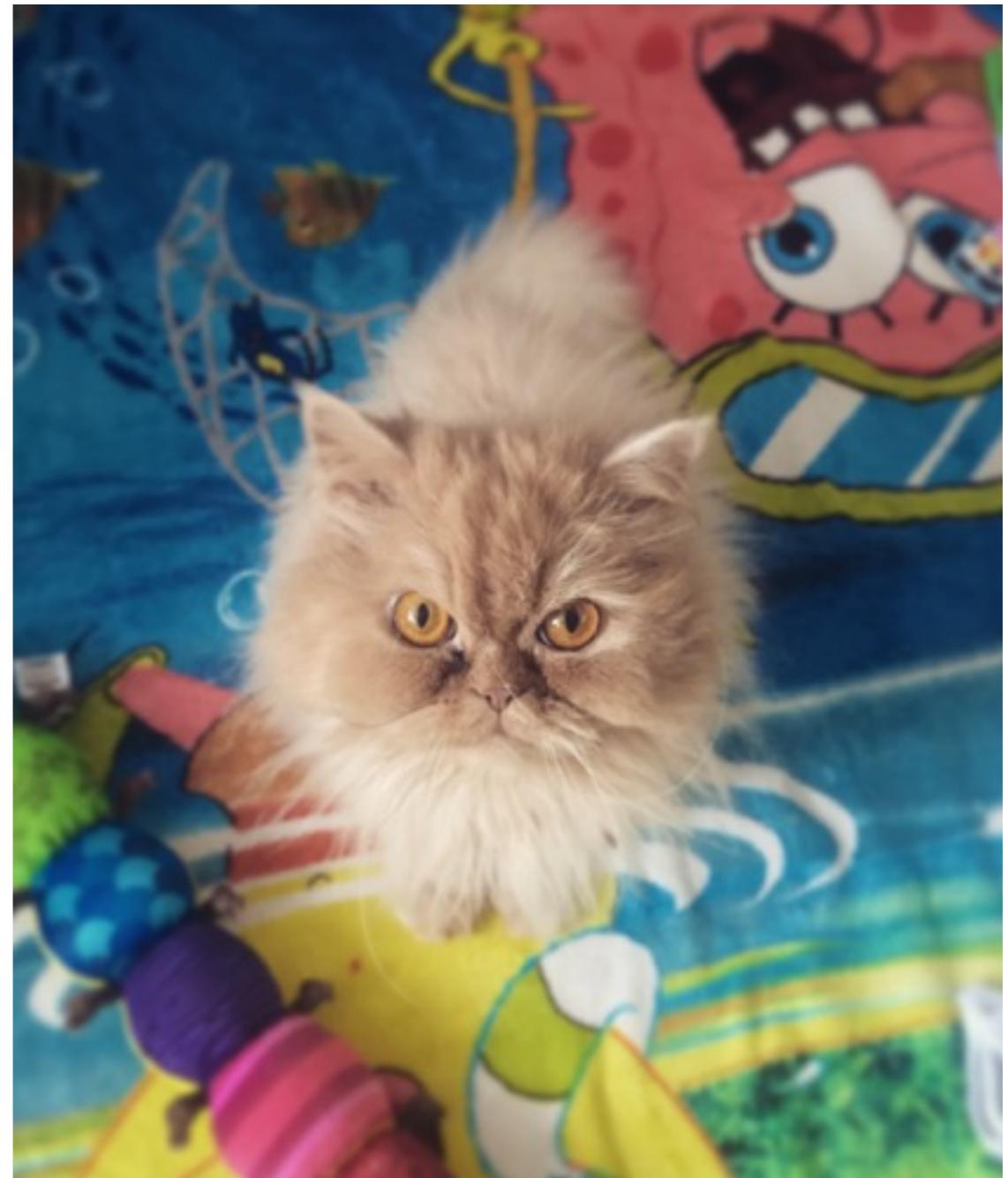


A close-up photograph of a woman's face and upper body. She is wearing a light-colored, ribbed straw hat with a dark, curly feather tucked behind her ear. Her hair is dark and pulled back. She is looking directly at the camera with a neutral expression. The background is dark and out of focus.

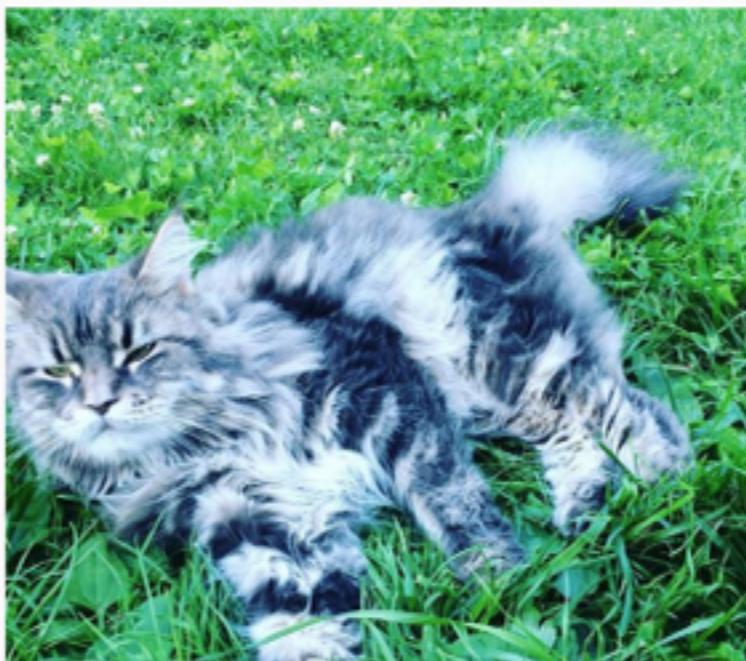
An Image...

Что мы знаем о картинках?

- › На них котики!
- › Все любят котиков. И картинки.
- › Люди - визуалы!
- › Больше котиков - лучше!



We need moooarr!!



Но за все надо платить

- › Img - один из самых тяжелых элементом страницы.
- › Много больших, много маленьких.
- › Fullscreen «Retina-ready» Image – 5-20mb.

	DSC_2479.JPG	Вчера	12,4 МБ
	DSC_2475.JPG	Вчера	12,4 МБ
	DSC_2462.JPG	Вчера	13,5 МБ
	DSC_2460.JPG	Вчера	9,4 МБ
	DSC_2452.JPG	Вчера	13,5 МБ
	DSC_2443.JPG	Вчера	10,9 МБ
	DSC_2442.JPG	Вчера	10,6 МБ
	DSC_2437.JPG	Вчера	9,3 МБ
	DSC_2432.JPG	Вчера	10,9 МБ
	DSC_2424.JPG	Вчера	13,1 МБ
	DSC_2421.JPG	Вчера	11 МБ
	DSC_2418.JPG	Вчера	12,6 МБ
	DSC_2417.JPG	Вчера	10,8 МБ
	DSC_2410.JPG	Вчера	11,1 МБ
	DSC_2407.JPG	Вчера	10,2 МБ
	DSC_2404.JPG	Вчера	10,9 МБ
	DSC_2393.JPG	Вчера	10,5 МБ
	DSC_2392.JPG	Вчера	7,8 МБ
	DSC_2391.JPG	Вчера	9,5 МБ
	DSC_2387.JPG	Вчера	11,9 МБ
	DSC_2383.JPG	Вчера	11,4 МБ

В общем самый жир...

- › Скорость соединения - невелика.
- › Потребность в картинках - велика.
- › Сиюминутный аппетит - ограничен.

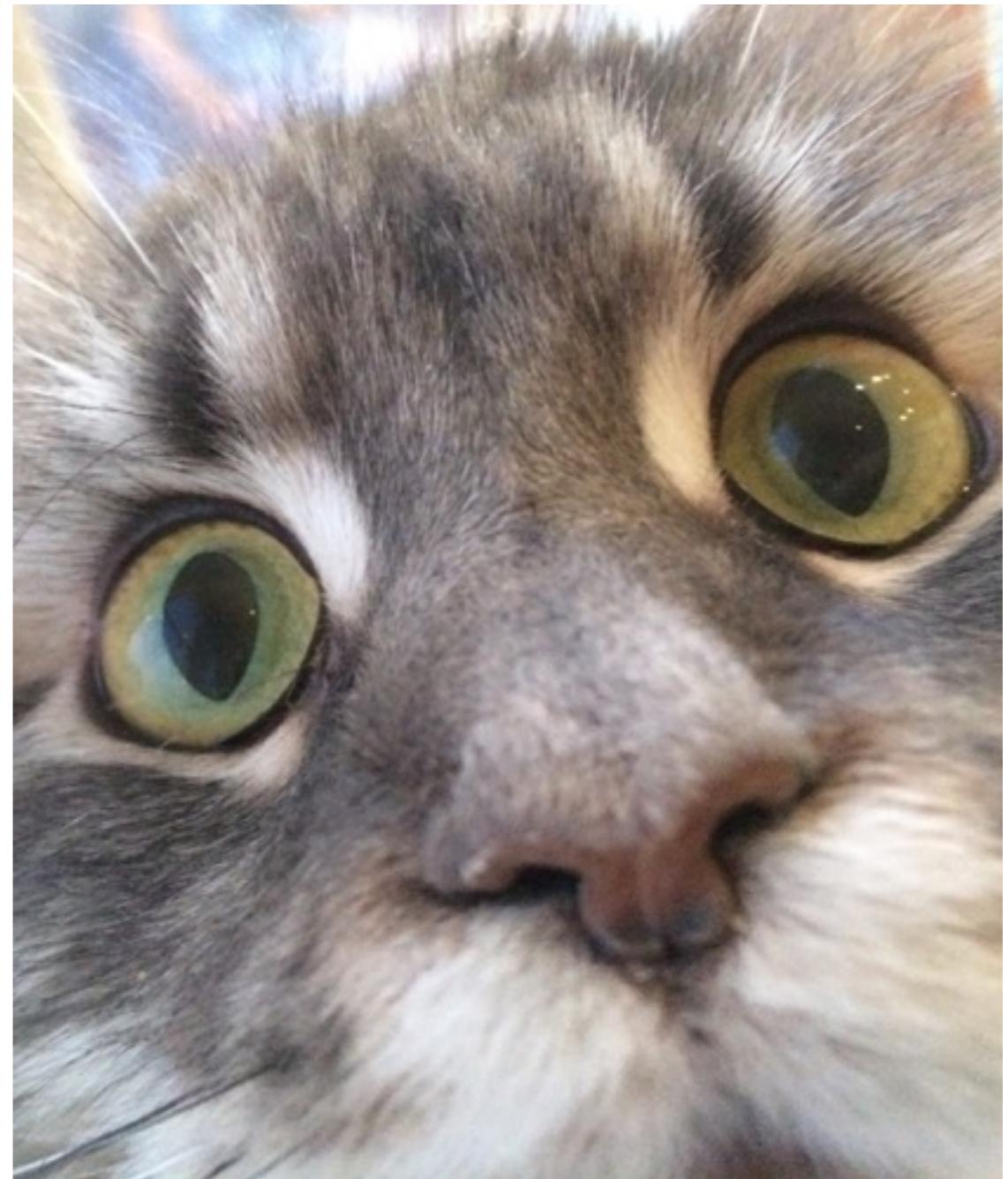


Картинки НЕ нужны!



Картинки не нужны,

- › Когда мы их не видим!
- › Или когда мы их **еще** не видим...



A close-up photograph of a woman's face and upper body. She is wearing a light-colored straw hat with a dark feather on the side. Her hair is dark and pulled back. She is looking directly at the camera with a neutral expression. The background is dark and out of focus.

Lazy Load

jQuery.lazyLoad

early 2010...

```
<img src='image1.jpg'>  
jQuery('img').lazyLoad();
```

```
<img data-lazy-src='image1.jpg'>  
jQuery('img').lazyLoad();
```

187 requests. Pending...

Orthodoxie und Celibat

The screenshot shows the Network tab in the Chrome DevTools developer console. The title bar indicates there are 187 requests pending. The main area displays a table of network requests with the following columns: Name, Method, Status, Type, Initiator, Size, and Time. The 'Img' tab is selected in the filter bar. The table lists several requests, mostly for images (webp format) from JoyReactor, with some requests initiated by the parser. The bottom status bar shows 91 / 187 requests and 12.7 KB / 6.7 MB transferred.

Name Path	Method	Status Text	Type	Initiator	Size Content	Time
/images/icon_en.png	GET	200	webp	Other	0 B	0 B
504058 <img0.joyreactor.c...< td=""><td>GET</td><td>200</td><td>jpeg</td><td>Other</td><td>0 B</td><td>0 B</td></img0.joyreactor.c...<>	GET	200	jpeg	Other	0 B	0 B
LoL-Ero-League-... <img1.joyreactor.c...< td=""><td>GET</td><td>200</td><td>webp</td><td>Other</td><td>0 B</td><td>0 B</td></img1.joyreactor.c...<>	GET	200	webp	Other	0 B	0 B
reactor_bg_middl... <umlhux.joyreacto...< td=""><td>GET</td><td>200</td><td>webp</td><td><u>(index):142</u> Parser</td><td>0 B</td><td>0 B</td></umlhux.joyreacto...<>	GET	200	webp	<u>(index):142</u> Parser	0 B	0 B
icon_home.png <umlhux.joyreacto...< td=""><td>GET</td><td>200</td><td>webp</td><td><u>(index):142</u> Parser</td><td>0 B</td><td>0 B</td></umlhux.joyreacto...<>	GET	200	webp	<u>(index):142</u> Parser	0 B	0 B
joyreactor_logo.p... <umlhux.joyreacto...< td=""><td>GET</td><td>200</td><td>webp</td><td><u>(index):142</u> Parser</td><td>0 B</td><td>0 B</td></umlhux.joyreacto...<>	GET	200	webp	<u>(index):142</u> Parser	0 B	0 B

91 / 187 requests | 12.7 KB / 6.7 MB transferred

Profit?

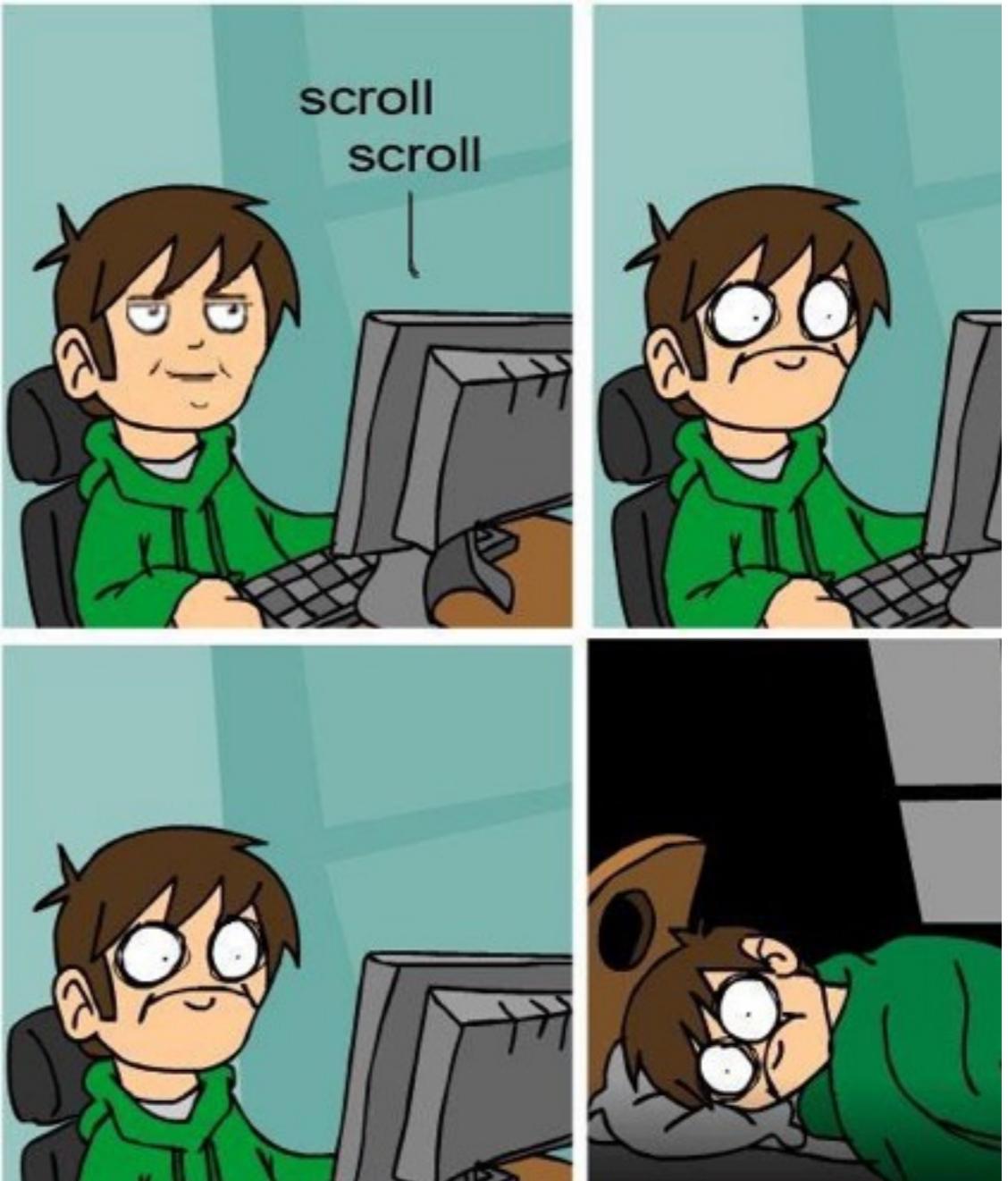
- › Экономия канала и трафика.
- › Более плавная загрузка как клиента, так и сервера.



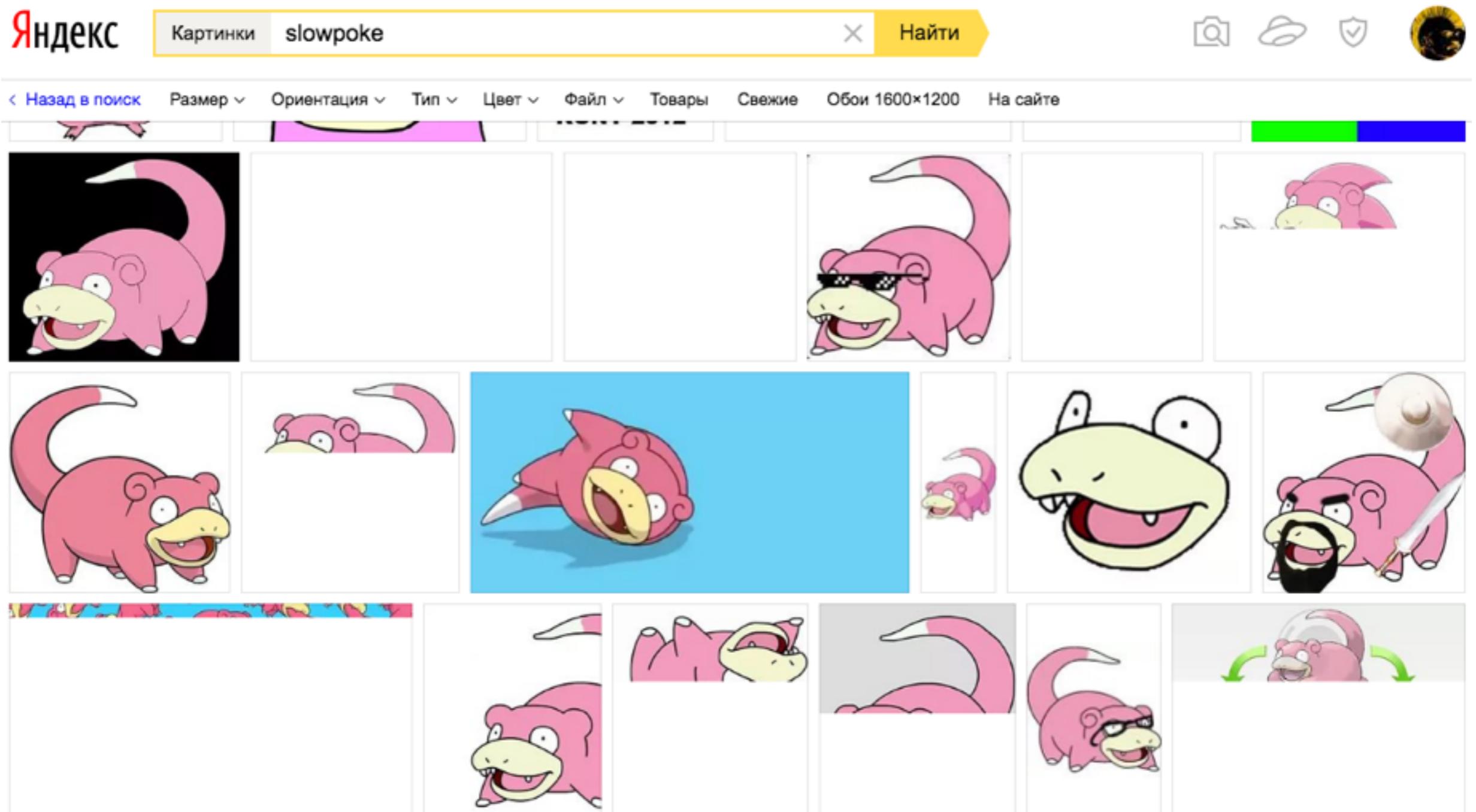
Problems?

- › Нельзя прервать загрузку изображения.
- › Не загружает «ненужное».

PS: кому ненужное?



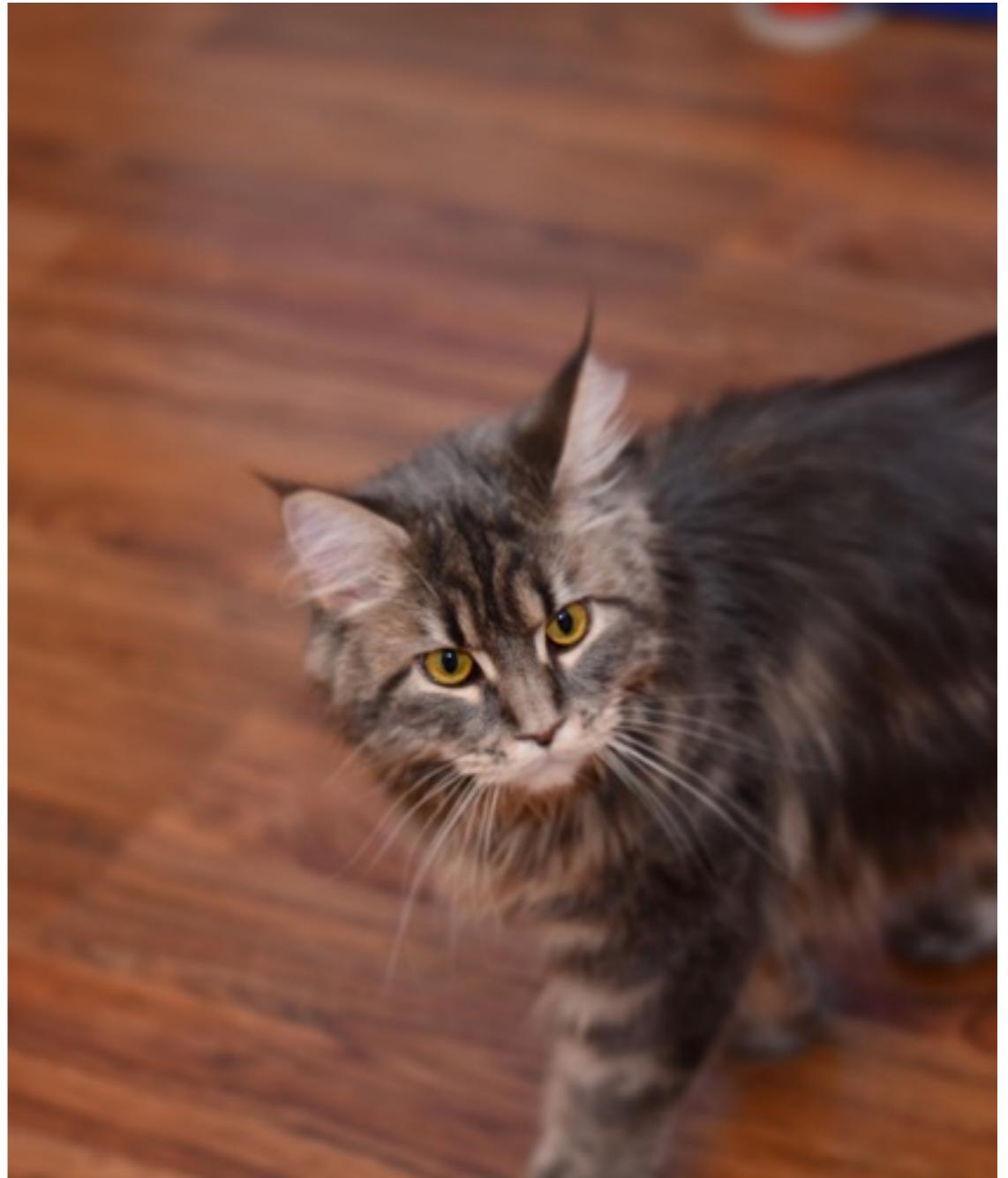
Лень загружать...

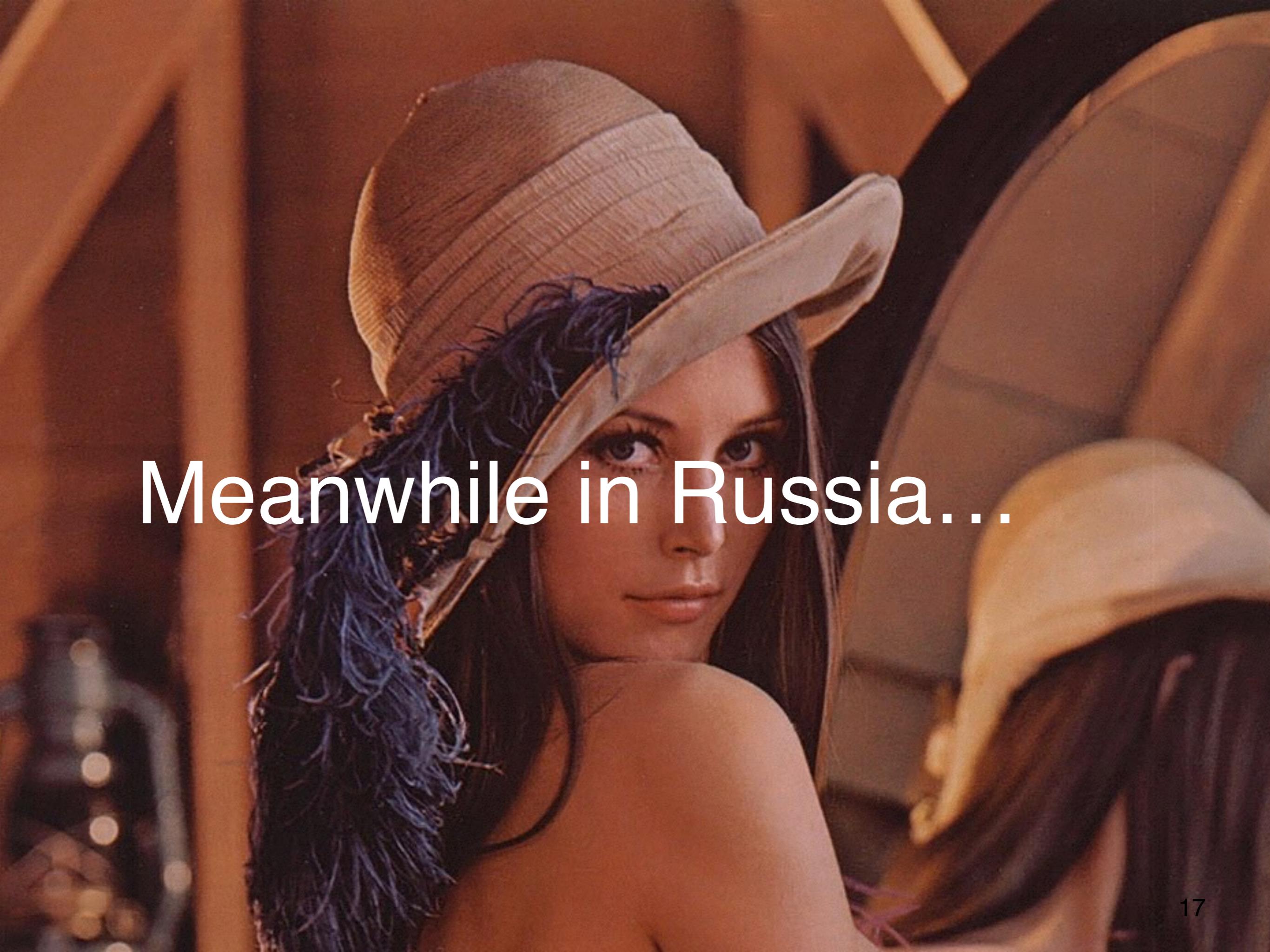


)

А что если решения нет?

- › Без lazyLoad всего много, долго и глупо.
- › С lazyLoad никакого префетча, опережающей загрузки.

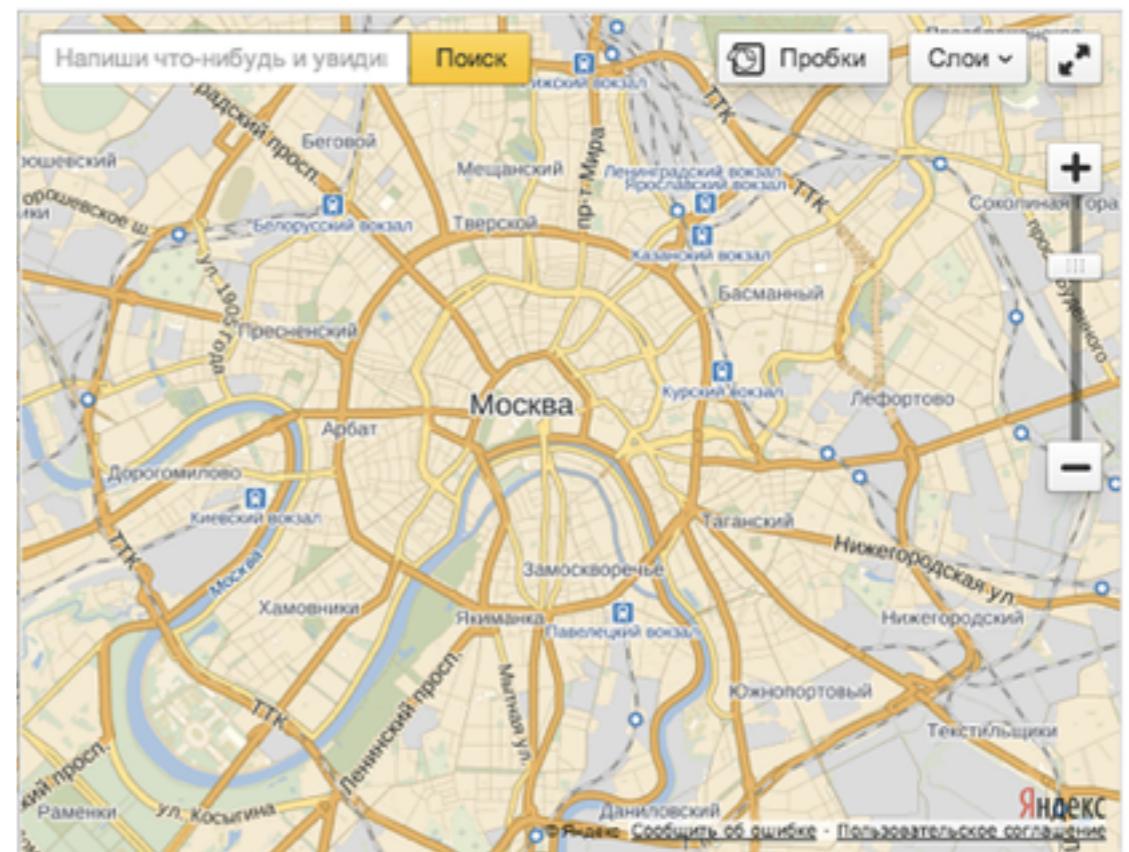


A close-up photograph of a woman's face and upper body. She is wearing a light-colored, ribbed straw hat with a dark, curly feather tucked behind her ear. Her gaze is directed towards the camera over her right shoulder. The lighting is warm and dramatic, casting shadows on one side of her face. The background is dark and out of focus.

Meanwhile in Russia...

Развитие API Карт

- › 2004 год – запуск Яндекс.Карт
- › API 1.0 – 13 ноября 2008
- › API 1.1 – 25 июня 2009
- › API 2.0 – 19 апреля 2012
- › API 2.1 – 4 октября 2013
- › Что же можно делать столько лет?



Яндекс.Карты

Карты - это очень много маленьких по размеру (но не по объему) картинок.



Карта - она как пирог

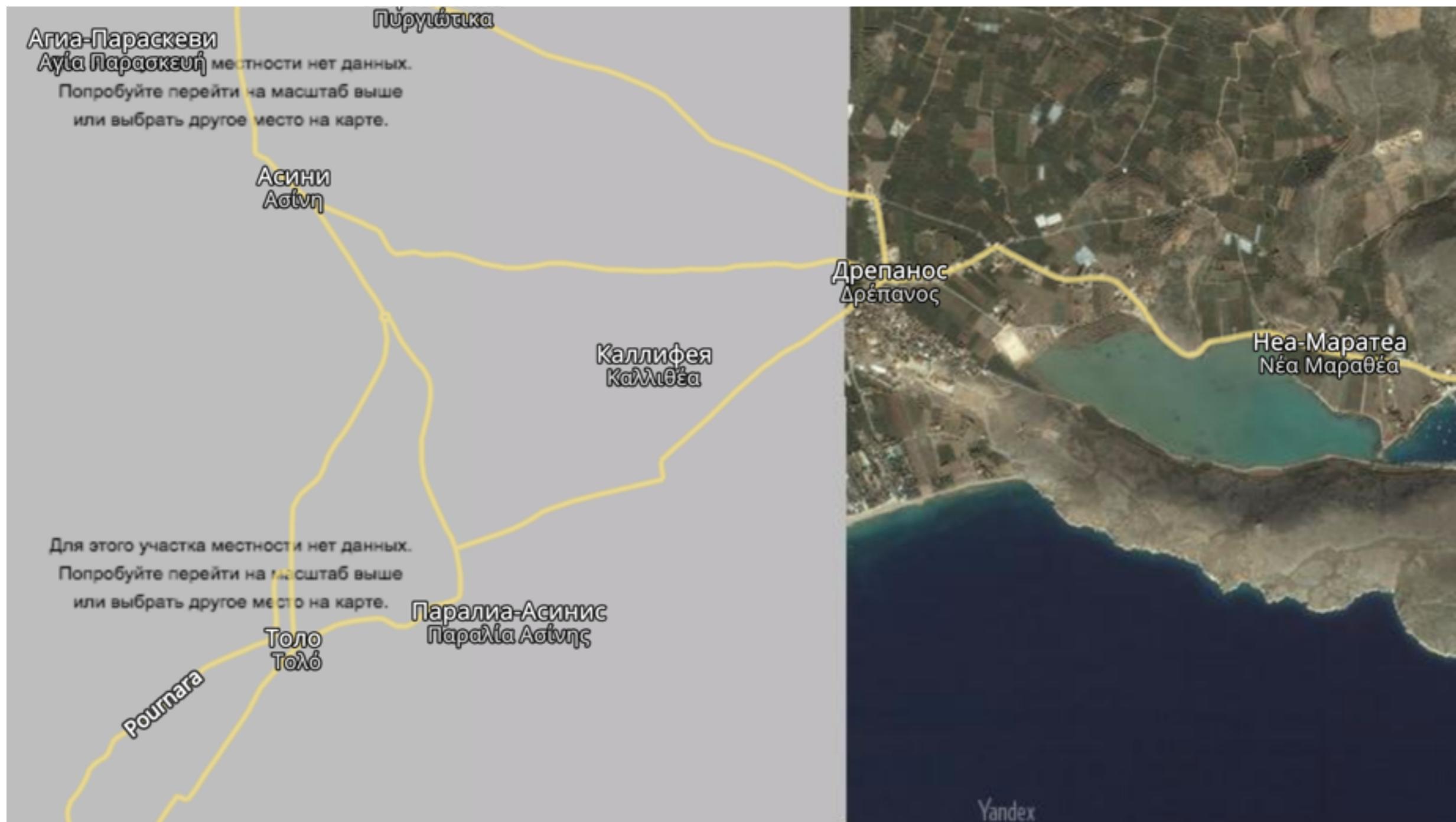


util.imageLoader

One ring to rule them all, one ring to find them, One ring
to bring them all and in the darkness bind them...

```
/**  
 * @param {String} url Image to load  
 * @param {Function} callback  
 */  
imageLoader.load(url, function (image) {  
});
```

Важно знать, что картинки нет



util.imageLoader

Явно сообщает о проблемах загрузки

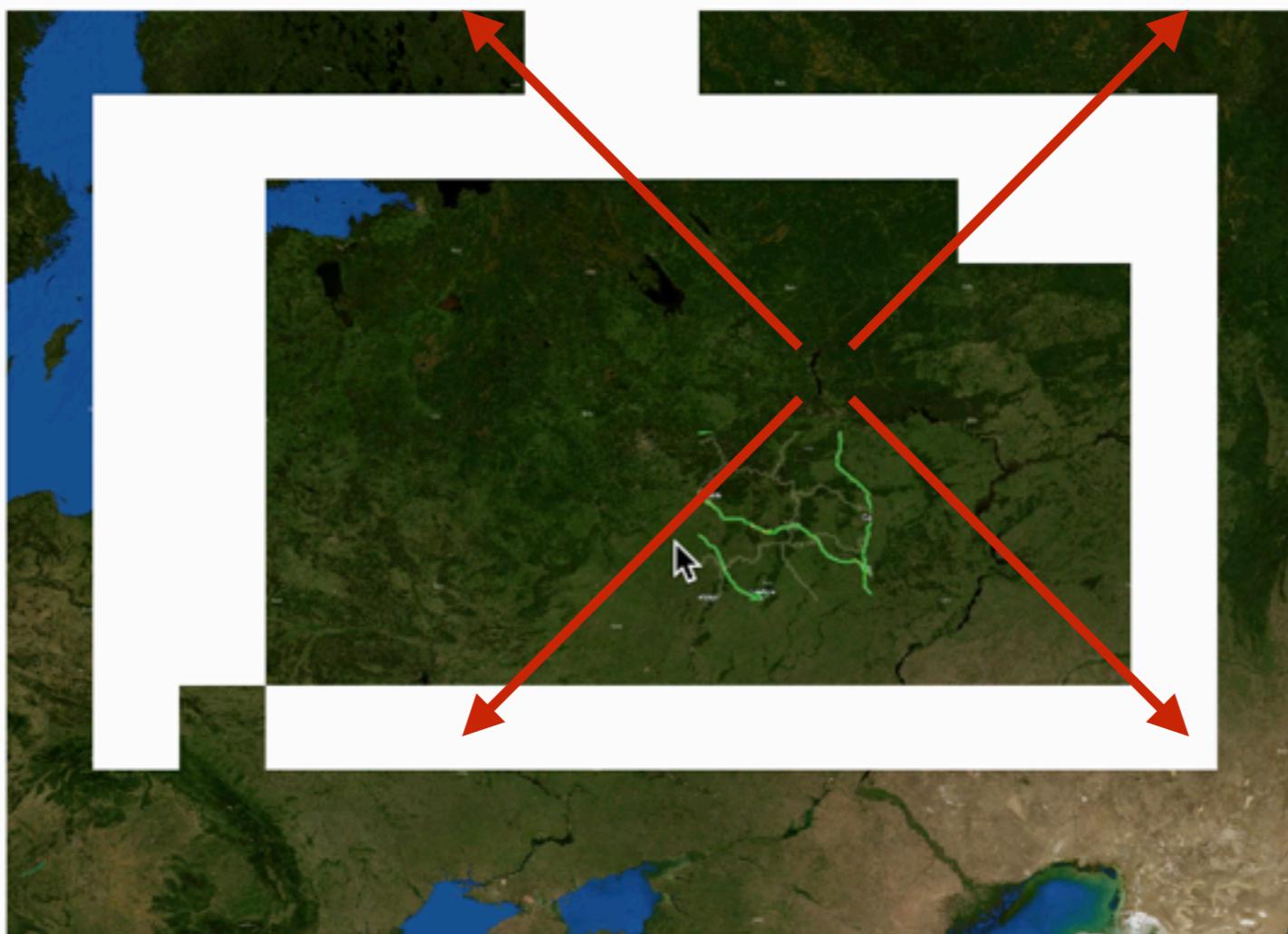
```
/**  
 * @param {String} url Image to load  
 * @param {Function} callback  
 */  
imageLoader.load(url, function (image, state) {  
});
```

util.imageLoader

Позволяет управлять порядком загрузки. Зачем?

```
/**  
 * @param {String} url Адрес изображение  
 * @param {Integer} priority Приоритет загрузки  
 * @param {Function} callback  
 */  
imageLoader.load(url, priority, function (image, state) {  
});  
// priority = 1.0 / tileDist;
```

Не все тайлы одинаково полезны



Speed Limits!

- › Количество одновременных запросов.
- › Таймауты.
- › Допустимые временные кванты.
- › Скорость обработки очереди.



Метки? Они тоже картинки!

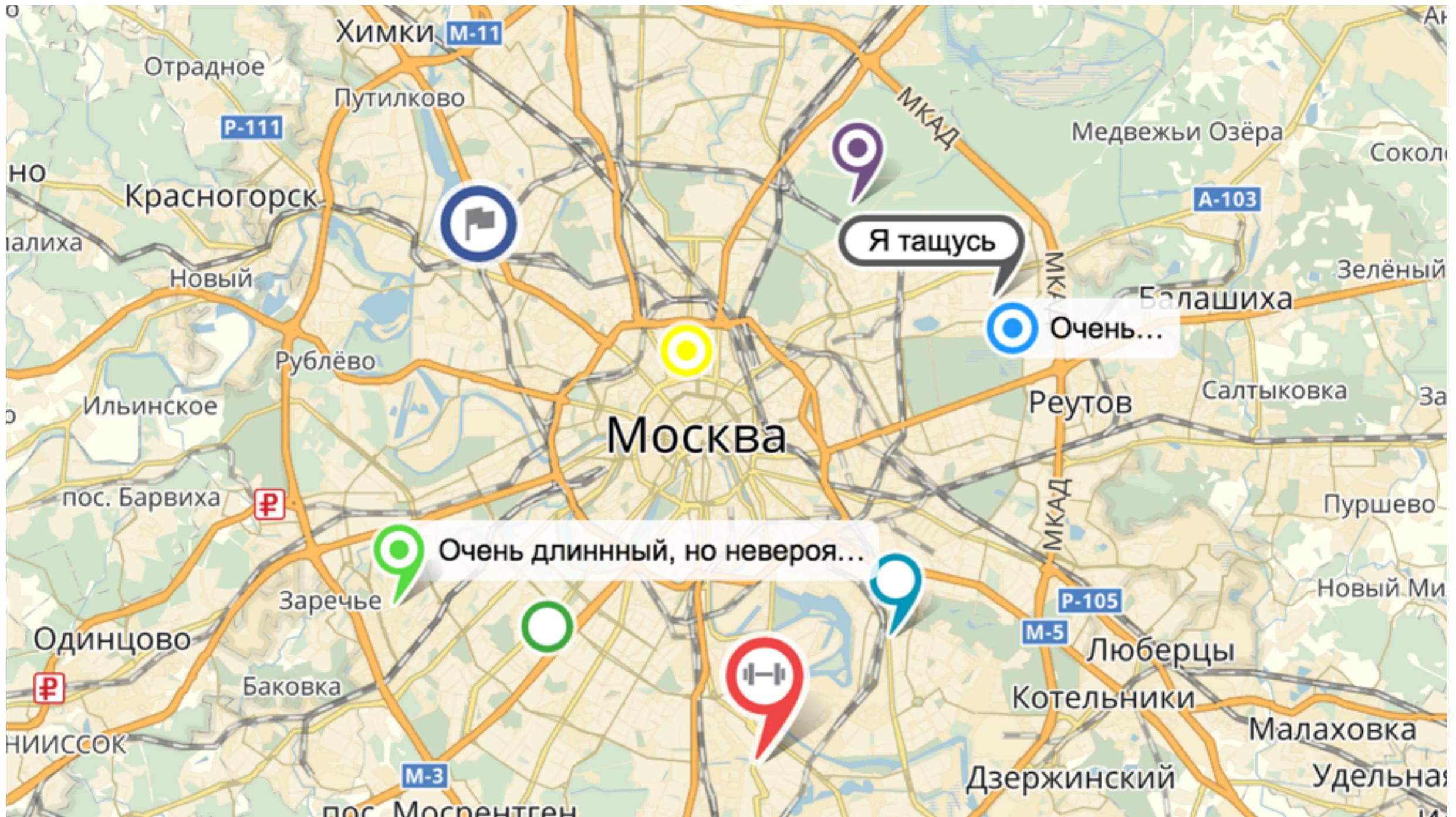


util.imageLoader

Мухи. Отдельно. Котлеты. Отдельно.

```
/**  
 * @param {String|Object} coordinates  
 * @param {Function} callback  
**/  
imageLoader.load({  
  url: url,  
  priority: priority,  
  channel: channel // tiles, placemarks, dataURI, functional  
, function (image, state) {  
});
```

Картички бывают разными



Measure twice..

- › DataURI - не самый быстрый, и не самый компактный.
- › SVG - компактный, но ваще не быстрый.
- › Canvas - удобный, но мало применимый.
- › BlobURL - клевый, но неудобный!



SVG -> PNG

The screenshot shows a browser developer tools element inspector with two icons selected. The top icon is a blue circle with a white center, and the bottom icon is a stylized number 9. Both icons have their styles and base64 image URLs displayed in the right panel.

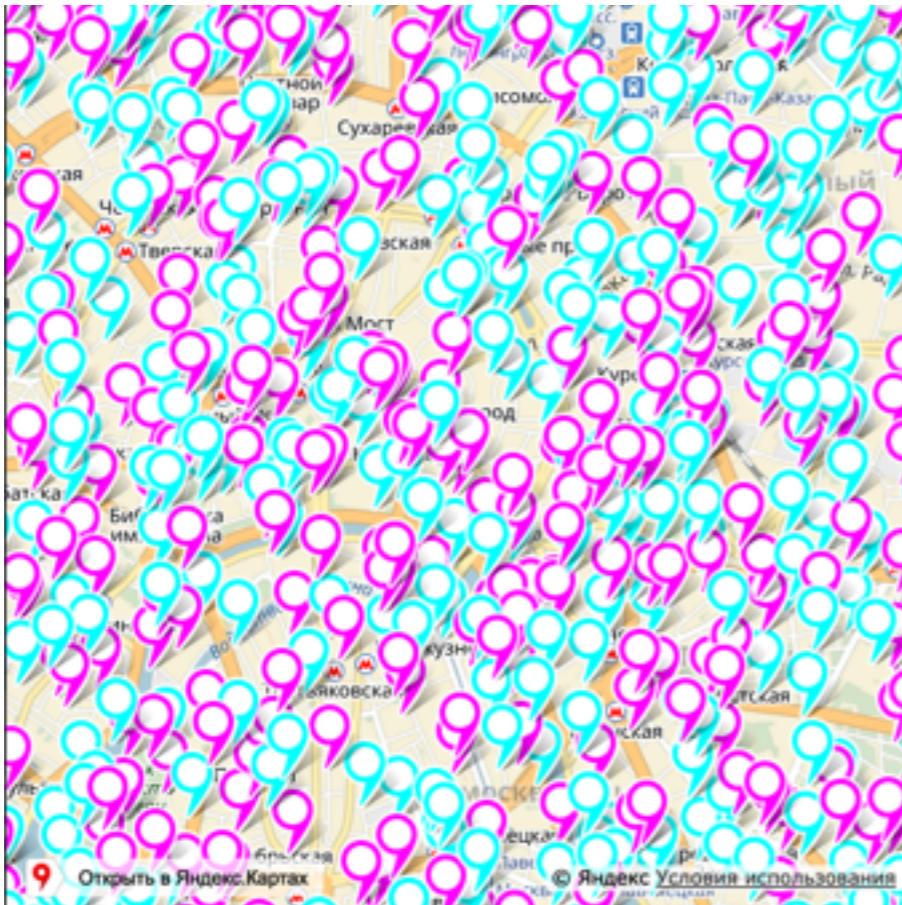
Top Icon (Blue Circle):

```
><ymaps>
  <ymaps class="ymaps-2-1-42-placemark-overlay ymaps-2-1-42-user-selection-none" unselectable="on" style="z-index: 650; height: 0px; width: 0px; position: absolute; left: 211px; top: 327px;">
    ...
  </ymaps>
  <ymaps class="ymaps-2-1-42-placemark-overlay ymaps-2-1-42-user-selection-none" unselectable="on" style="z-index: 650; height: 0px; width: 0px; position: absolute; left: 88px; top: 89px;">
    ...
  <ymaps>
    <ymaps id="id_147063689893154057326">
      <ymaps class="ymaps-2-1-42-svg-icon ymaps_https__api_maps_yandex_ru_2_1_42_604203414303islands_circleIcon__3b5998_40x40_1470643963500" style="position: absolute; width: 40px; height: 40px; left: -20px; top: -20px;"></ymaps> == $0
    </ymaps>
    <ymaps class="ymaps-2-1-42-pictogramCircleIconContent ymaps-2-1-42->
```

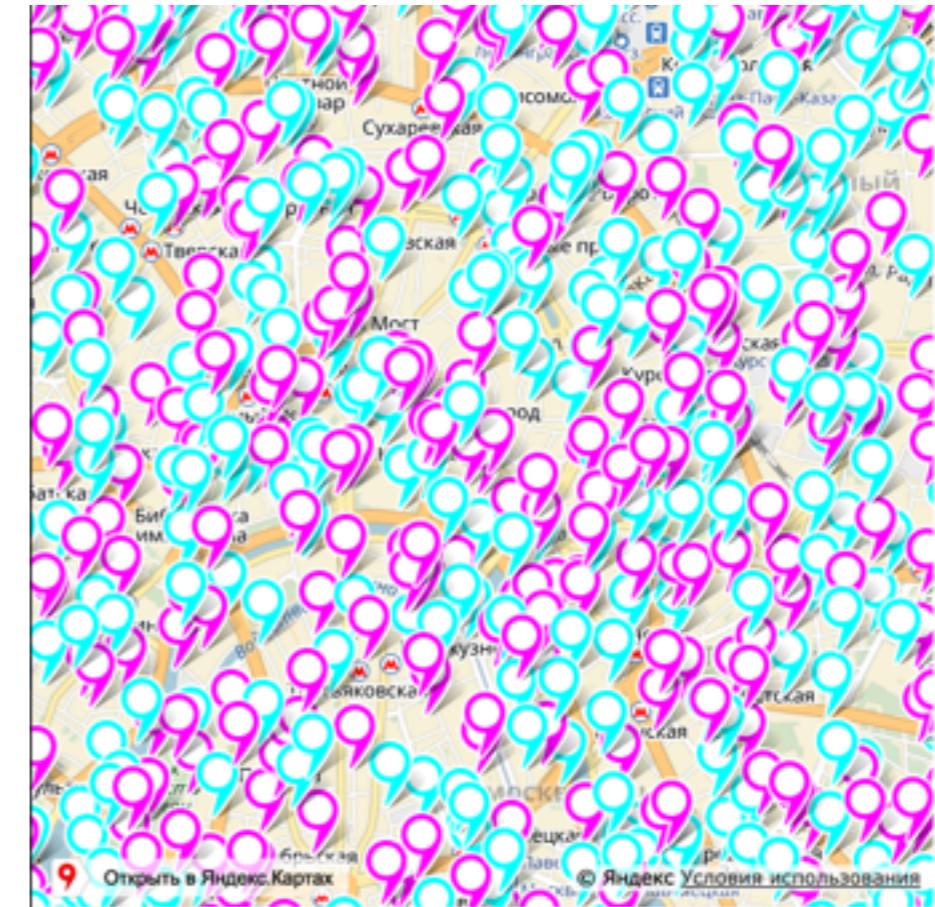
Bottom Icon (Stylized Number 9):

```
<s class="ymaps-2-1-43-controls-pane" style="width: 100%; top: 0px; left: 0px; position: absolute; transform: translate3d(0px, 0px, 0px) scale(1, 1); z-index: 5403;">_</ymaps>
<s class="ymaps-2-1-43-places-pane" style="top: 0px; left: 0px; position: absolute; transform: translate3d(0px, 0px, 0px) scale(1, 1); transition-duration: 0ms; z-index: 2404;">
<ymaps class="ymaps-2-1-43-placemark-overlay ymaps-2-1-43-user-selection-none" unselectable="on" style="z-index: 650; height: 0px; width: 0px; position: absolute; left: 623px; top: 282px;">_</ymaps>
<ymaps class="ymaps-2-1-43-placemark-overlay ymaps-2-1-43-user-selection-none" unselectable="on" style="z-index: 650; height: 0px; width: 0px; position: absolute; left: 480px; top: 315px;">
<ymaps class="ymaps-2-1-43-svg-icon ymaps_http__node_jsapi_kashey_fuji_dev_cloud_maps_yandex_net_1462633444358islands_icon__1E98FF_34x41_1470646950002" style="position: absolute; width: 34px; height: 41px; left: -11px; top: -38px;"> == $0
<ymaps class="ymaps-2-1-43-svg-icon-content" style="font: 13px Arial, sans-serif; position: absolute; text-align: center; left: 5px; top: 6px; width: 16px; height: 16px;">
<ymaps>1</ymaps>
```

SVG хороший. Но у нас много точек.

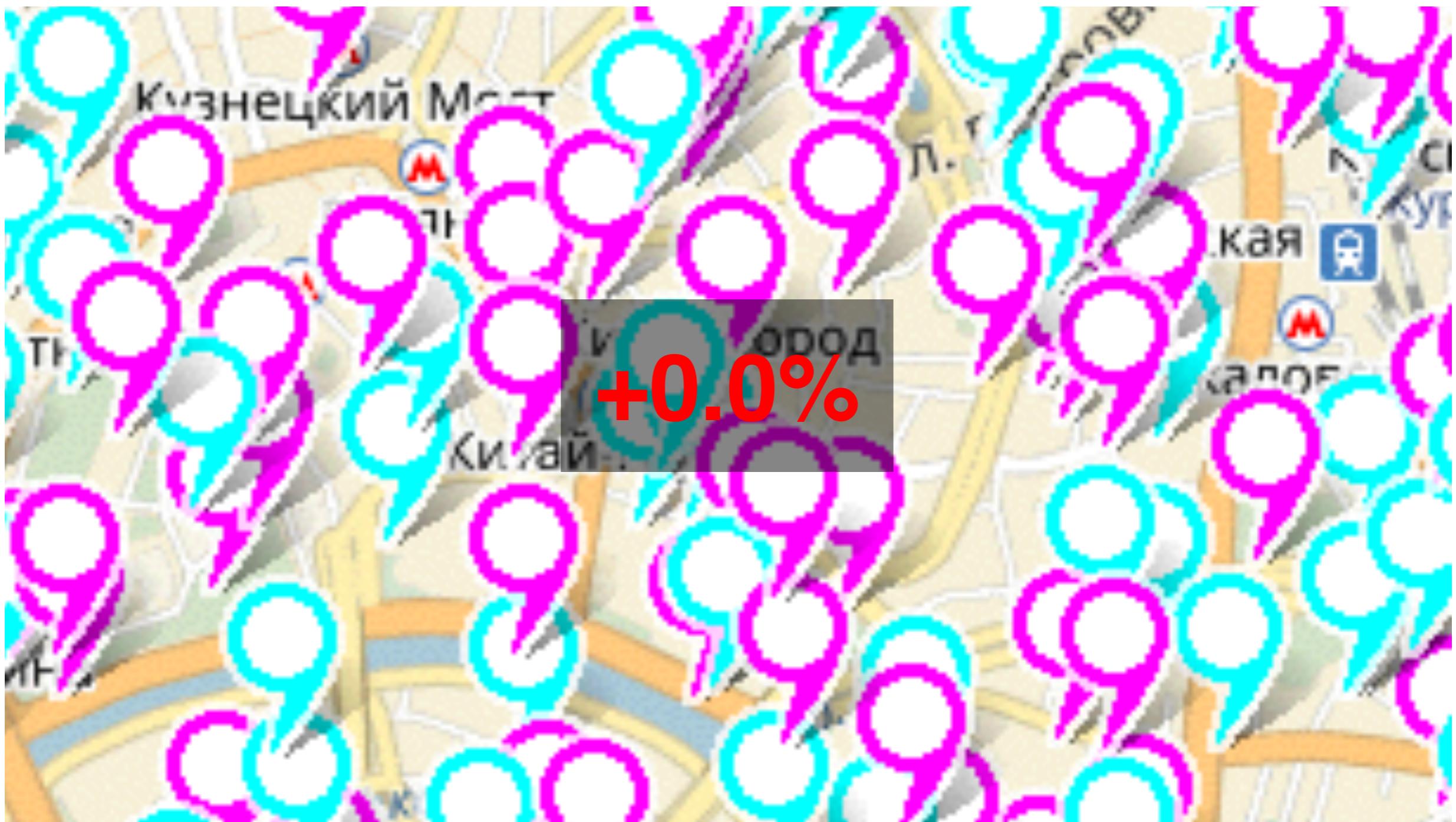


SVG – 10-30 FPS



PNG – 40-60

ShapeRendering: optimizeSpeed.



util.imageLoader

Тайлам – URL

Графике – Image

WebGL – CORS!

```
imageLoader.load({
  url: url,
  priority: priority,
  channel: 'tiles',
  returnAs: 'image', // url, bloburl, texture
, function (image, state) {
}) ;
```



MOAR!

util.imageLoader.proxy

Перехватить и обработать!

Только зачем?

```
imageLoader.proxy.add({  
  matchUrl: function (someUrl) {  
    },  
    /**  
     * @param {String} url  
     * @return {Promise}  
     */  
    load: function (url) {  
    }  
}) ;
```

A close-up photograph of a woman with long dark hair, wearing a wide-brimmed straw hat with a feather. She is looking over her shoulder at the camera with a neutral expression. The background is blurred, showing what appears to be a person in a suit.

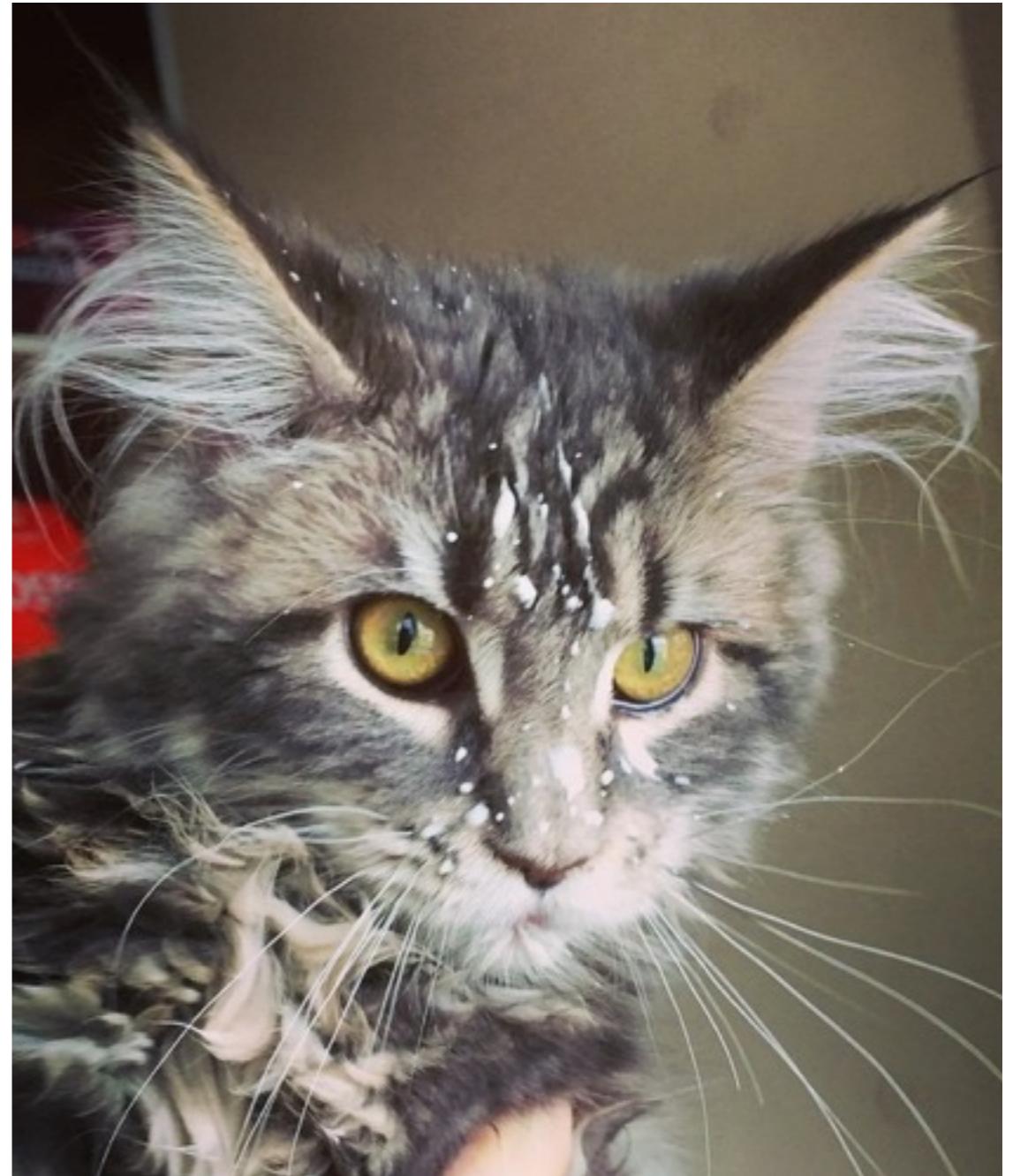
return a Promise/A+

Зачем мы туда полезли?

Важно понимать, что мы добавили роху исключительно потому, что:

МОЖЕМ

А можем, потому что imageLoader.



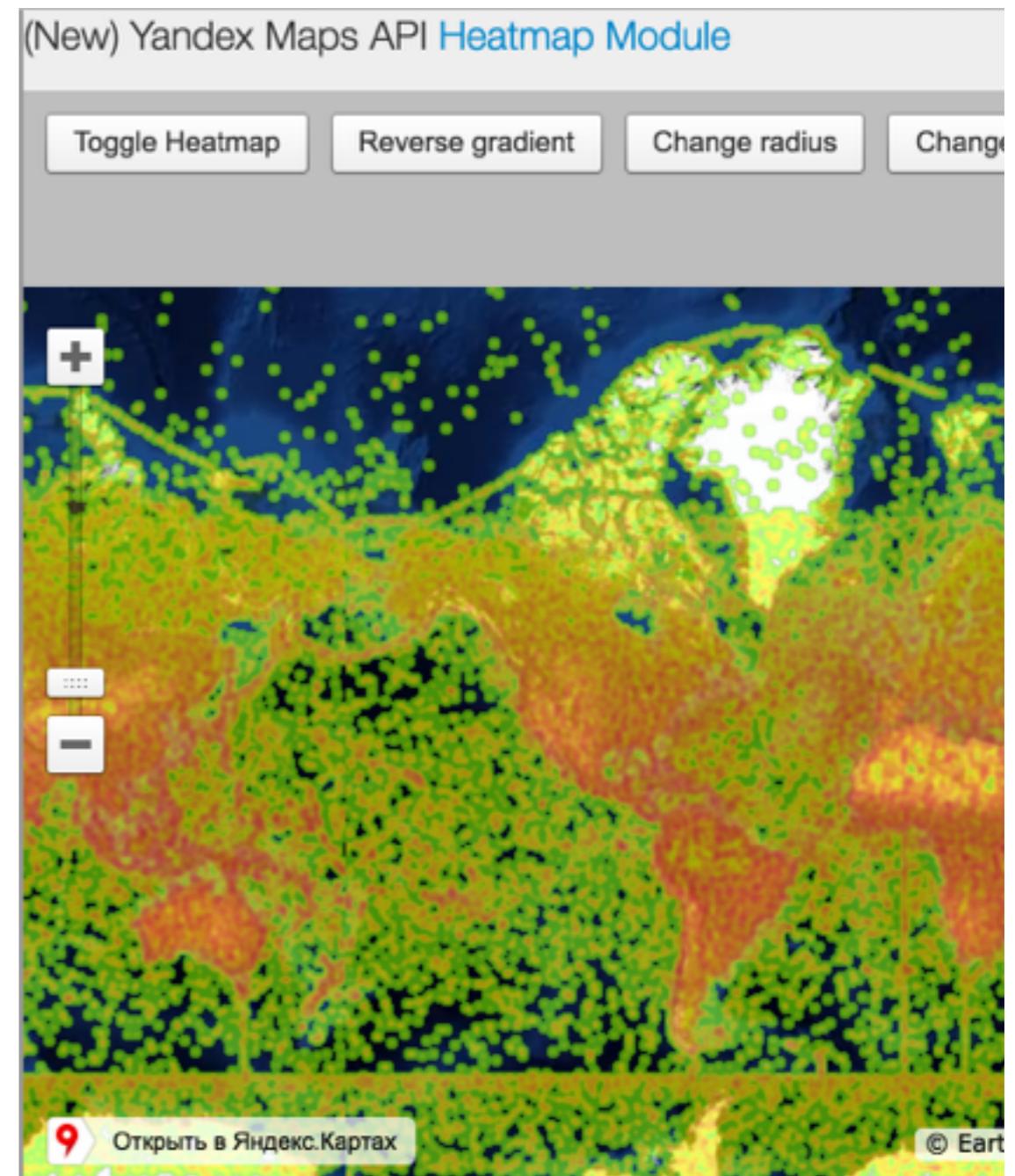
HeatMap

- › Удобный инструмент.
- › Который молча страдает.
- › От нашего API.



Проблемы первого мира...

- › HeatMap – процедурно-генерируемые тайлы.
- › Отдают dataURI вместо URL
- › Синхронно и одновременно.



util.imageLoader.proxy

Шаг 1 – отдаём виртуальный URL.

```
TileUrlsGenerator.prototype.getTileUrl = function (tileNumber, zoom)
{
    return `ym-heatmap://mid=${this.heatMapID}/x=${tileNumber[0]}/
y=${tileNumber[1]}/z=${zoom}`;
};
```

util.imageLoader.proxy

Шаг 2 – прокси забирает этот адрес себе.

```
imageLoader proxy.add({
  matchUrl: function (url) {
    return url && url.indexOf('ym-heatmap') === 0 ? 1 : 0;
  },
  load: function (url, request) {
    ...
  }
});
```

util.imageLoader.proxy

Шаг 3 – дергает «реальную» функцию генерации тайла.

```
load: function (url, request) {
  if (mapId && lookupStorage[mapId]) {
    return lookupStorage[mapId].getTileImage([+x, +y], +z)
  } else {
    return vow.reject();
  }
}
```

util.imageLoader.proxy

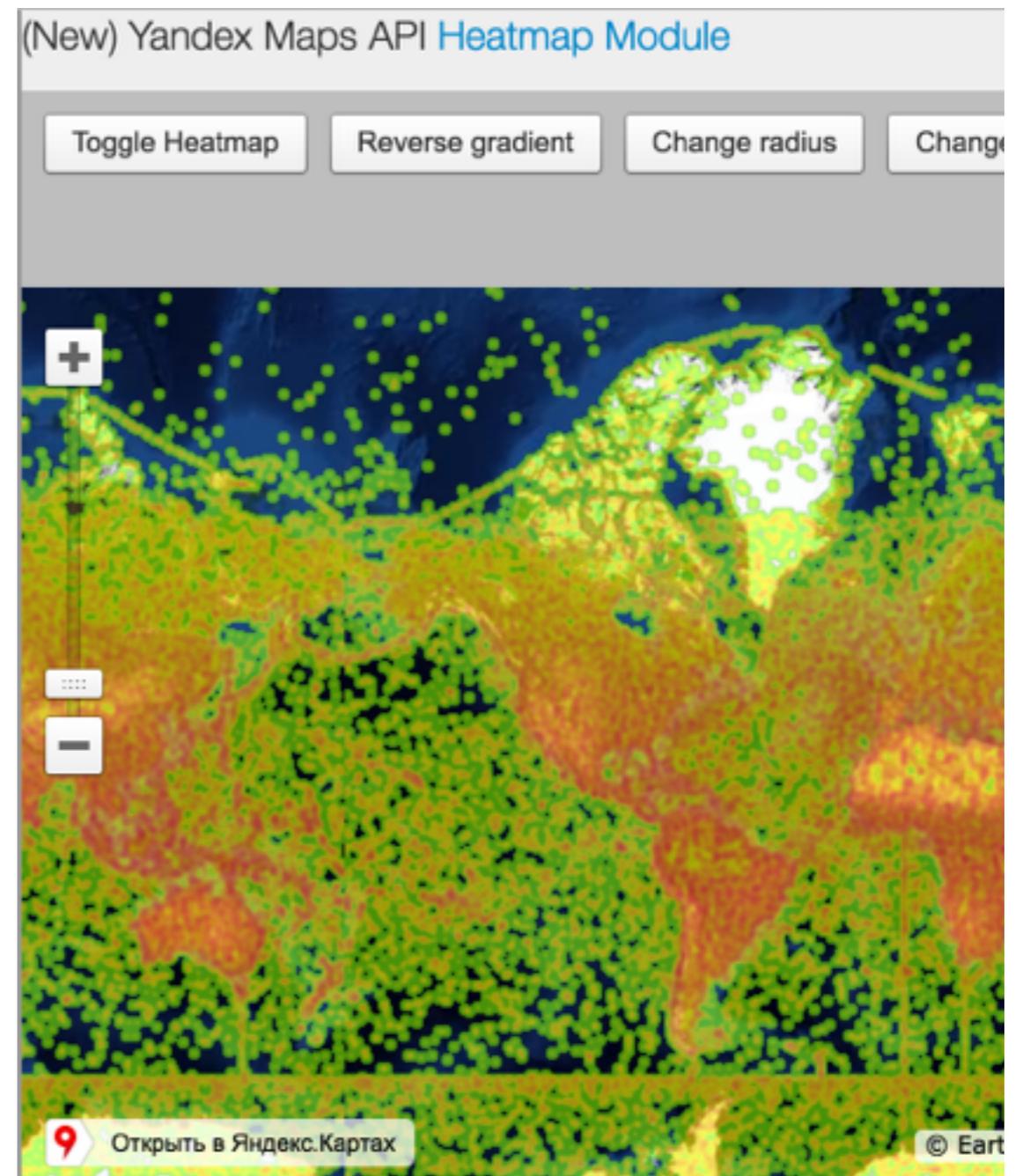
```
TileUrlsGenerator.prototype.getTileUrl = function (tileNumber, zoom) {
    return `ym-heatmap://mid=${this.heatMapID}/x=${tileNumber[0]}/y=${tileNumber[1]}/
z=${zoom}`;
};
//.....
imageLoader proxy.add({
    matchUrl: function (url) {
        return url && url.indexOf('ym-heatmap') === 0 ? 1 : 0; ----->
    },
    load: function (url, request) {
        //...
        if (params.mid && lookupStorage[params.mid]) {
            return vow
                .resolve(
                    heatmapLookupStorage[params.mid]
                        .getTileImage([+params.x, +params.y], +params.z)
                );
        } else {
            return vow.reject();
        }
    }
})
```

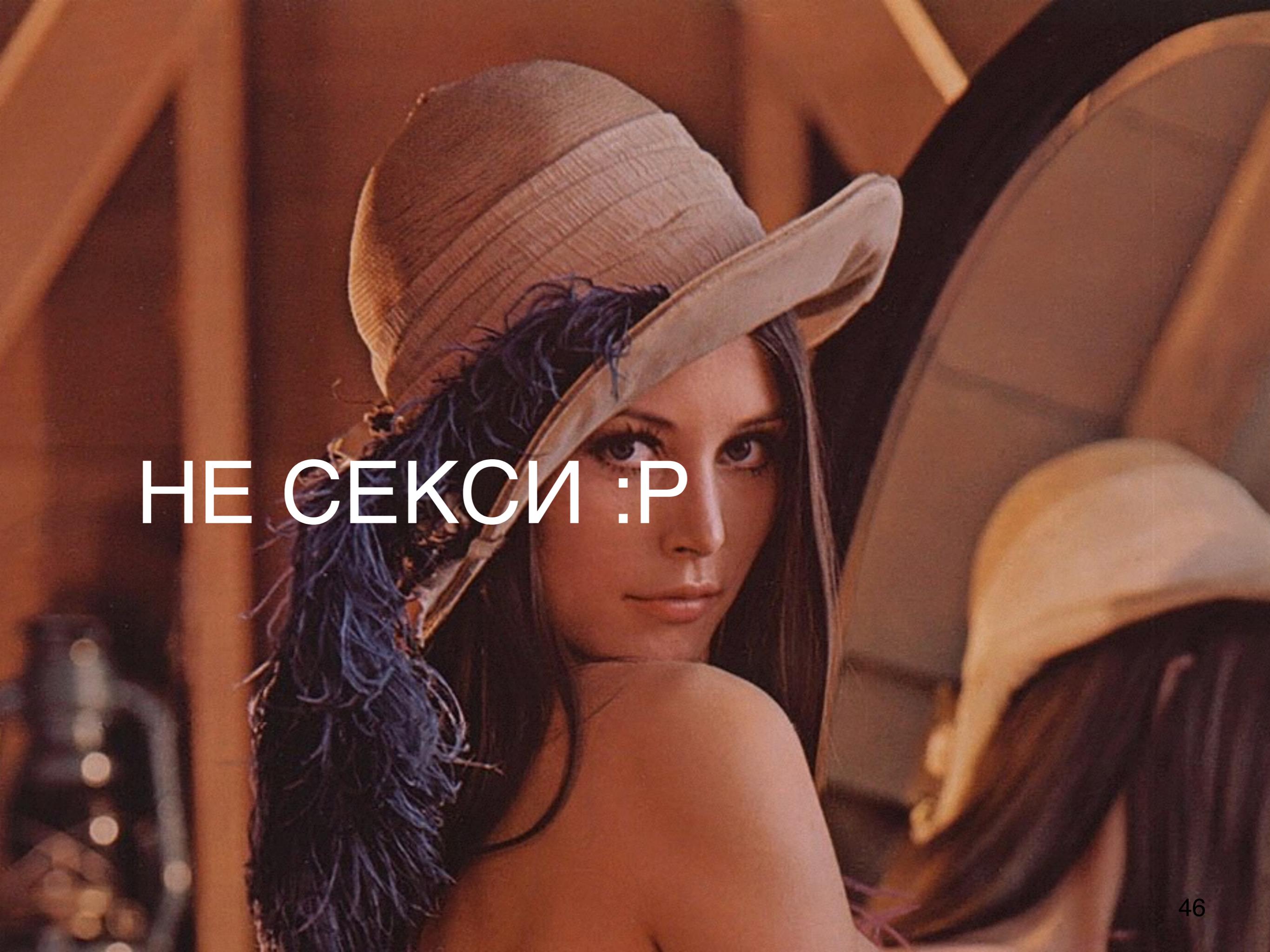
Measure twice..

В настоящий момент

- › асинхронные
- › и «нетормозящие»
- › кешируемые

Heatmaps работают как-то...



A photograph of a woman wearing a straw hat with a feather, looking over her shoulder at the camera. She is wearing a dark, patterned dress. To her right, a man in a dark tuxedo jacket and white shirt is visible, holding a glass. The background is dark and out of focus.

НЕ СЕКСИ :Р

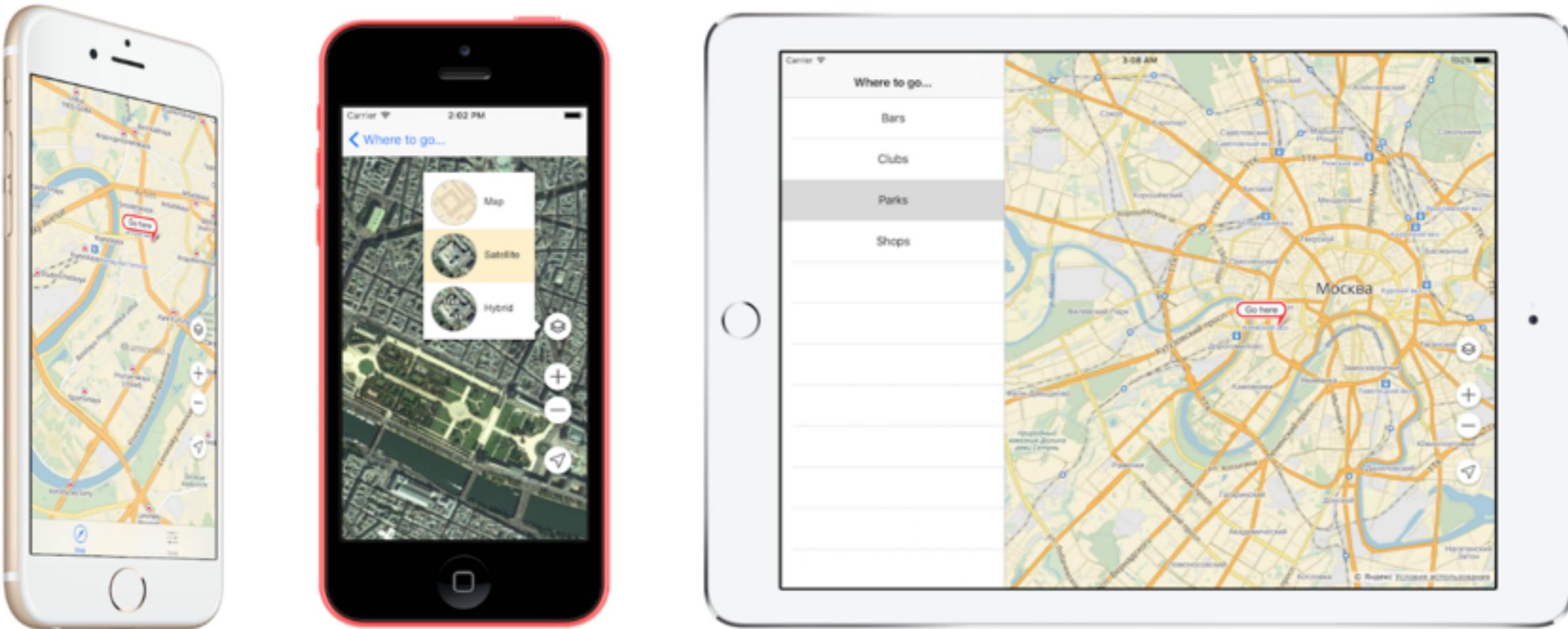
util.imageLoader.proxy

```
var RE_TILE_HOST = /vec\d+\.\maps\.\yandex\.net\/\/;
imageLoader.proxy.add({
  matchUrl: function (url) {
    return !!url.match(RE_TILE_HOST);
  },
  load: function (url) {
    return fetchLocalUrl(url, 'image/jpeg', getPath(url))
      .fail(function () { return url; });
  }
})
```

Offline cache!

MapsAPI

Allows to easily add Yandex.Maps to your existing iOS project using [Yandex.Maps JavaScript API](#).



util.imageLoader.load(function)

```
/**  
 * @param {String|Object|Function} coordinates  
 * @return {Promise}  
 */  
imageLoader.load(function() {  
    return madeSomeAsyncMagics(); // return a Promise  
});
```

non-ES6 Promises?

- 14?! Это нелепо! Нам нужно разработать один универсальный стандарт, чтобы все им пользовались.
- Да, точно!



Программируем, как умеем!

promise.progressNotifyTick()

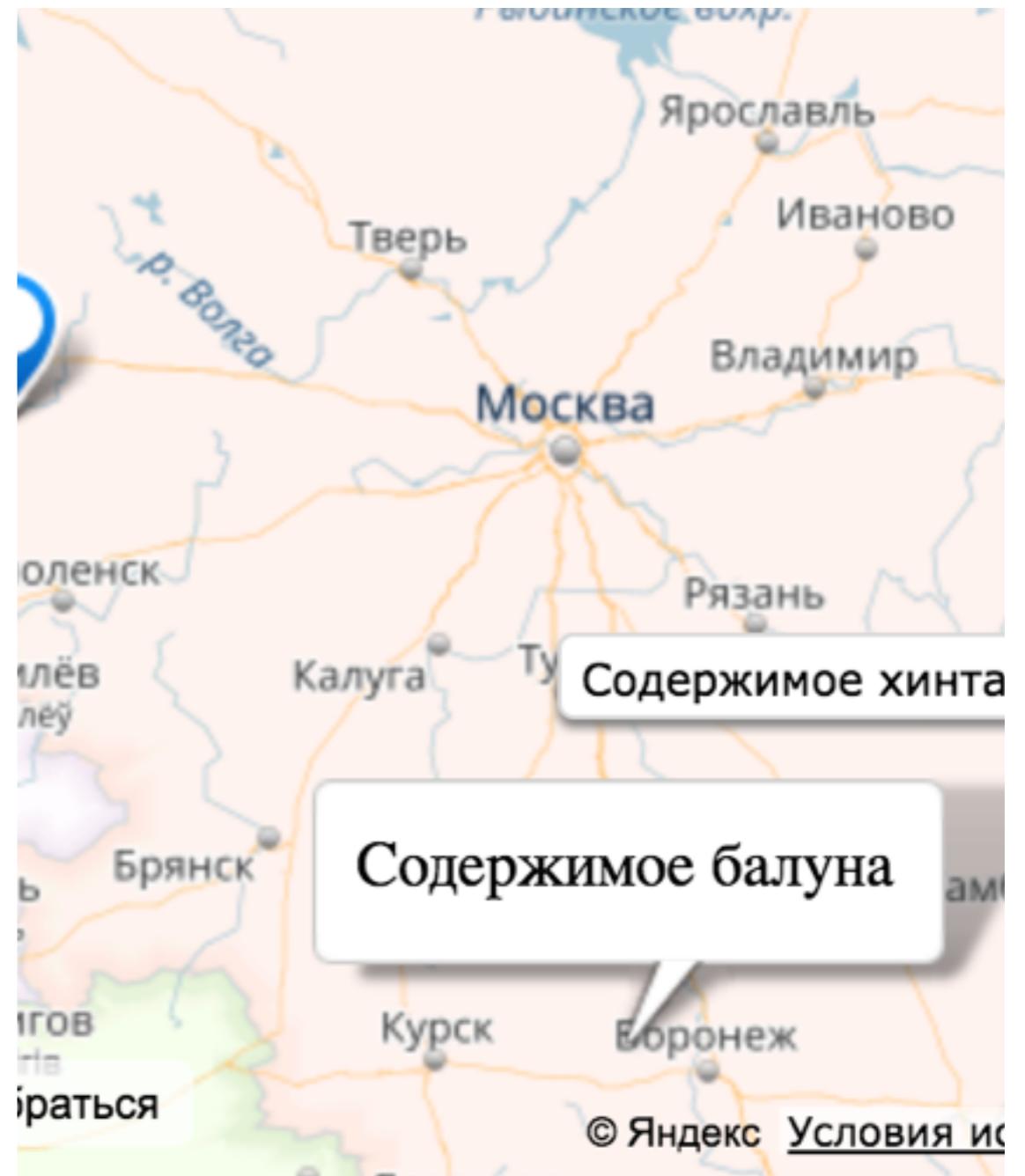
Был такой метод. Да всплыл.

```
imageLoader  
    .load(url)  
    .progress(fitToBounds)  
    .then(finish);
```

Зачем?

- › Progressive JPEG
- › Мобильные сети
- › Пользовательская верстка
- › Динамические размеры балунов.

В общем, «готовность» изображения раньше времени.



Остановите Землю! Тормозит!

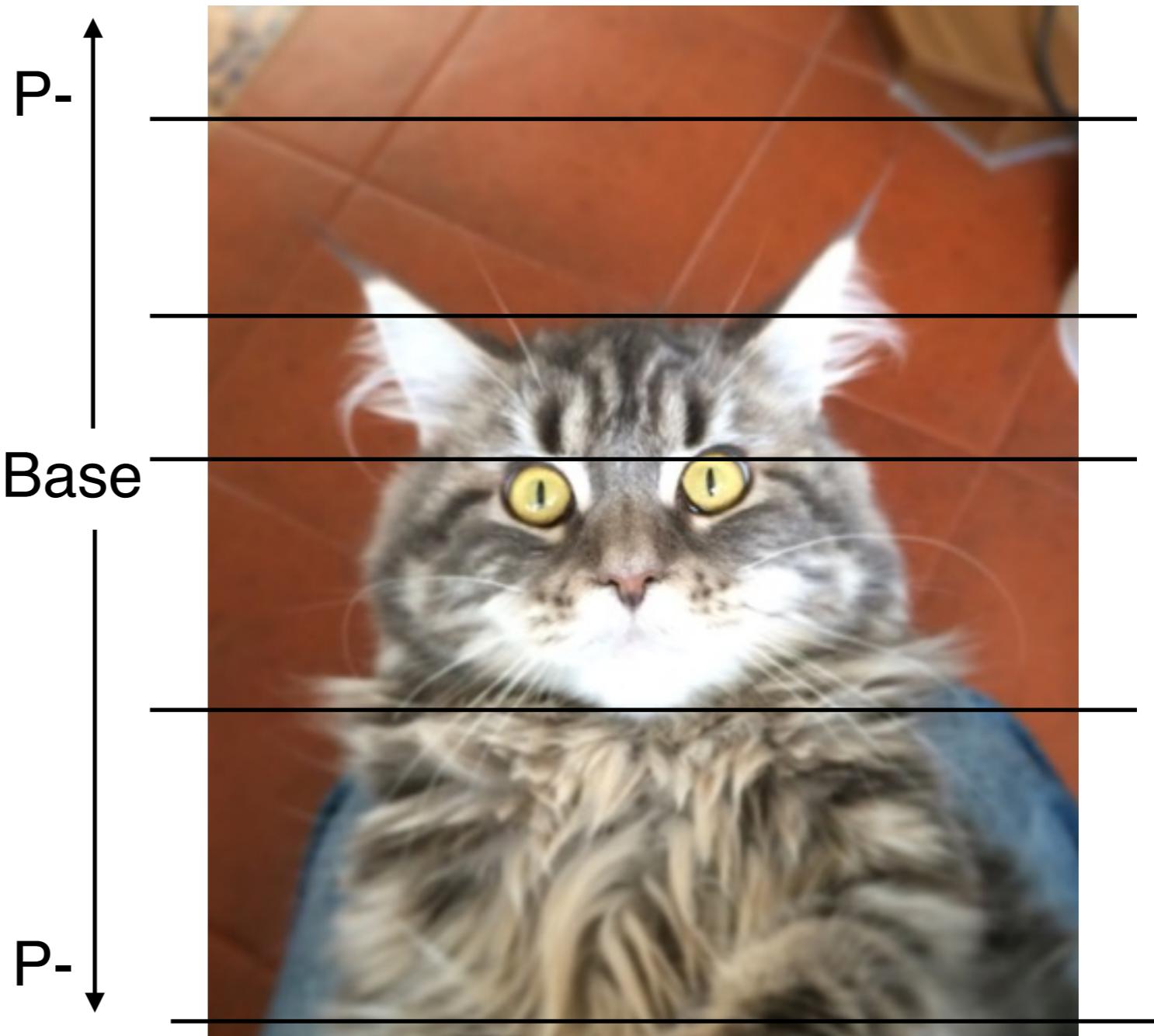
У нас еще есть
централизованный scheduler.

- › И world.stop, world.smooth.
- › Остановка мира блочит все.
- › Плавный режим - меняет
лимиты imageLoader.



СТОЙ! Так что с lazyLoad?

- › Да нормально все с ним.
- › Только smooth, а не lazy.
- › И да, есть нюансы.



Живите как кошка с собакой!



Разработчик

Пользователь

A close-up photograph of a woman with long dark hair, wearing a light-colored straw hat with a feather and a bikini top. She is looking over her shoulder with a neutral expression. The background is blurred.

! СЕКСИ !

Контакты

Антон Корзунов

Разработчик интерфейсов АПИ карт.



kashey@yandex-team.ru



thekashey