

Яндекс

карты



Простая обработка групп геообъектов на карте

Сергей Константинов

Руководитель группы разработки API Карт
Я.Субботник, Екатеринбург, 06.07.2013

API Яндекс.Карт – это
инструмент для размещения
карт на сайтах



TAXI.tm

[Онлайн заказ такси](#) [О сервисе](#)

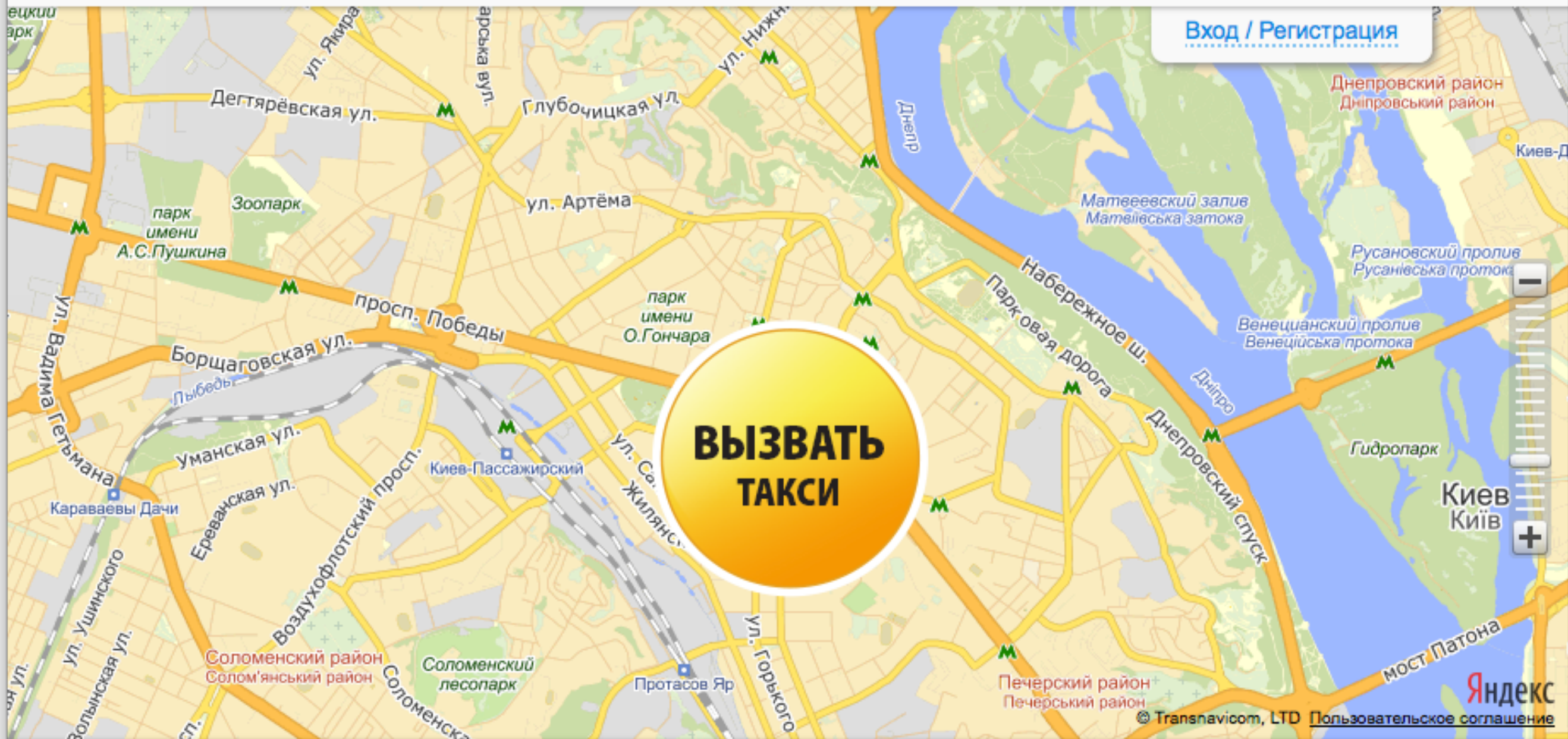
[Загрузить на
мобильный](#)

[Новости](#)



поддержка

[Вход / Регистрация](#)



Яндекс

© Transnavicom, LTD [Пользовательское соглашение](#)

[Контакты](#)



063 980-45-13



067 409-64-35



050 353-79-79



[Информация о
сервисе](#)

ТТК

Контакты | ТТК-Байкал

+

Яндекс

baikal-ttk.ru/contact/



Выберите город: Иркутск



+7 (3952) 79-88-88



[Личный кабинет](#) / [Подключиться](#)

Поиск

ТТК

Взгляни на мир
под другим углом

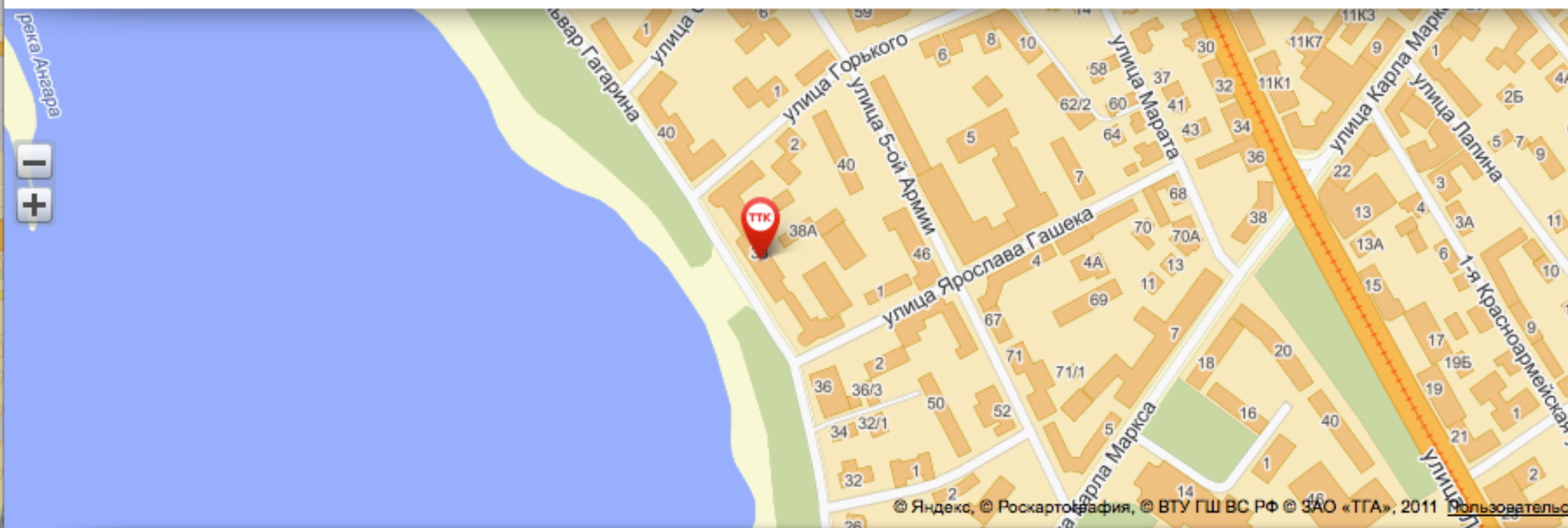
[Частным клиентам](#)

[Корпоративным клиентам](#)

[Новости](#)

[О компании](#)

К



Контактная информация

Город: Иркутск

Адрес: 664025, г. Иркутск, бульвар Гагарина, дом 38.

Реквизиты

Юридическое название:

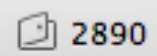
Б

Банкоматор



Яндекс

bankomator.ru/#lat=55.816895&lng=37.567844&z=11&



Банкоматор

[Яндекс.Карты](#)

[Google Maps](#)

Терминалы

Банкоматы



QIWI



Элекснет

Поиск на карте

Найти

Схема



© 2009-2012 Проект студии [Future Colors](#).

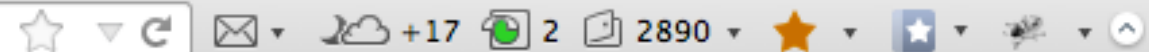


Яндекс

© ЗАО «ТГА», 2006. © ЗАО «Резидент», © Роскартография, © Яндекс — Условия использования

Свадьба. Свадебные платья, п...

Яндекс www.tili-testo.ru

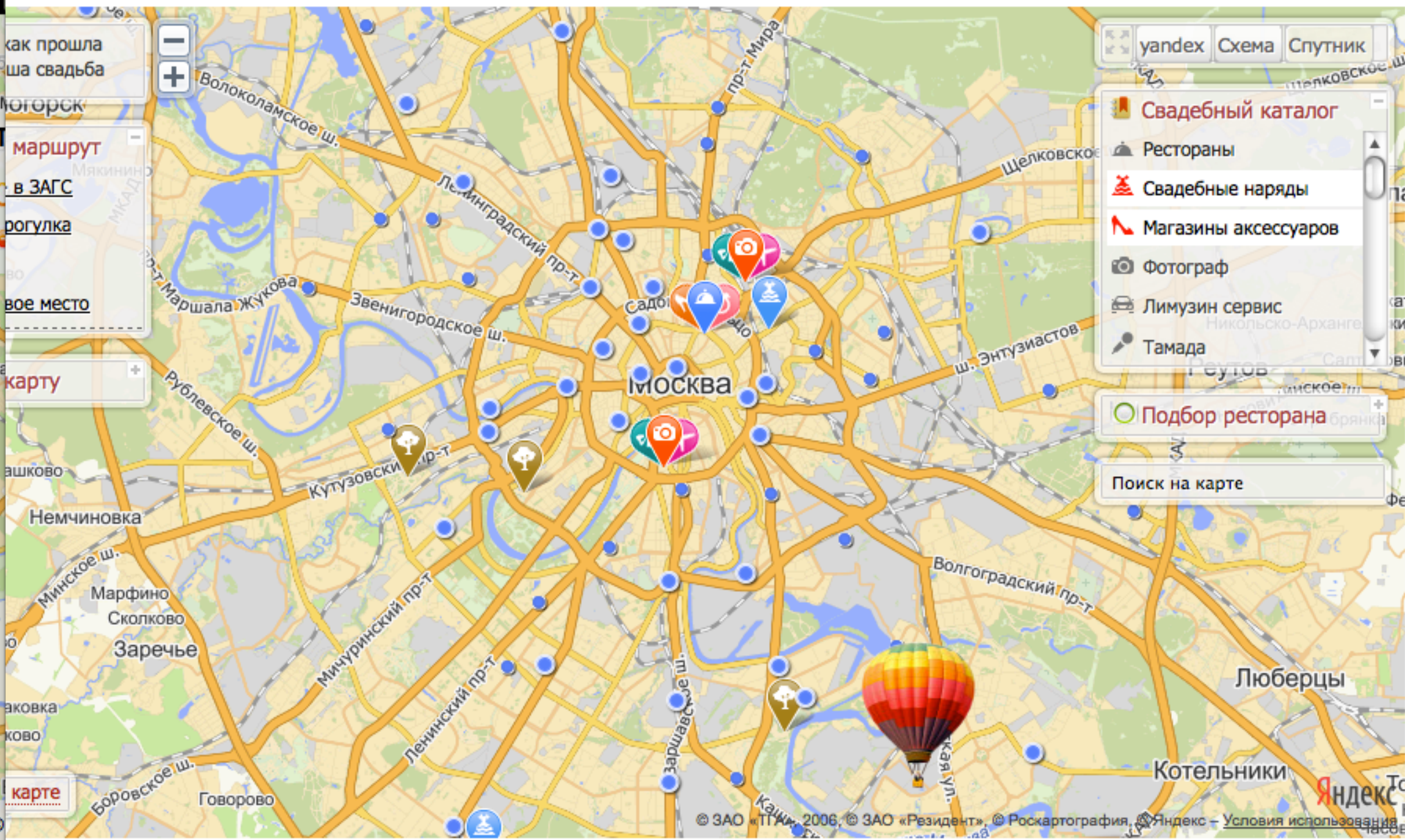


yandex Схема Спутник

- Свадебный каталог
- Рестораны
- Свадебные наряды
- Магазины аксессуаров
- Фотограф
- Лимузин сервис
- Тамада

Подбор ресторана

Поиск на карте



Скрыть свадебную карту

Яндекс

© ЗАО «ТТА», 2006. © ЗАО «Резидент». © Роскартография. © Яндекс — Условия использования

Карта Керчи – организации, п...

Яндекс

gorodkerch.com/map/#z7/c-3787.6155;11980.8656

Город
Керчь

Новости

Карта

Афиши

Желтые страницы

Эфир

Вход/Регистрация



Городской Шпион



Добавить

Типичные операции

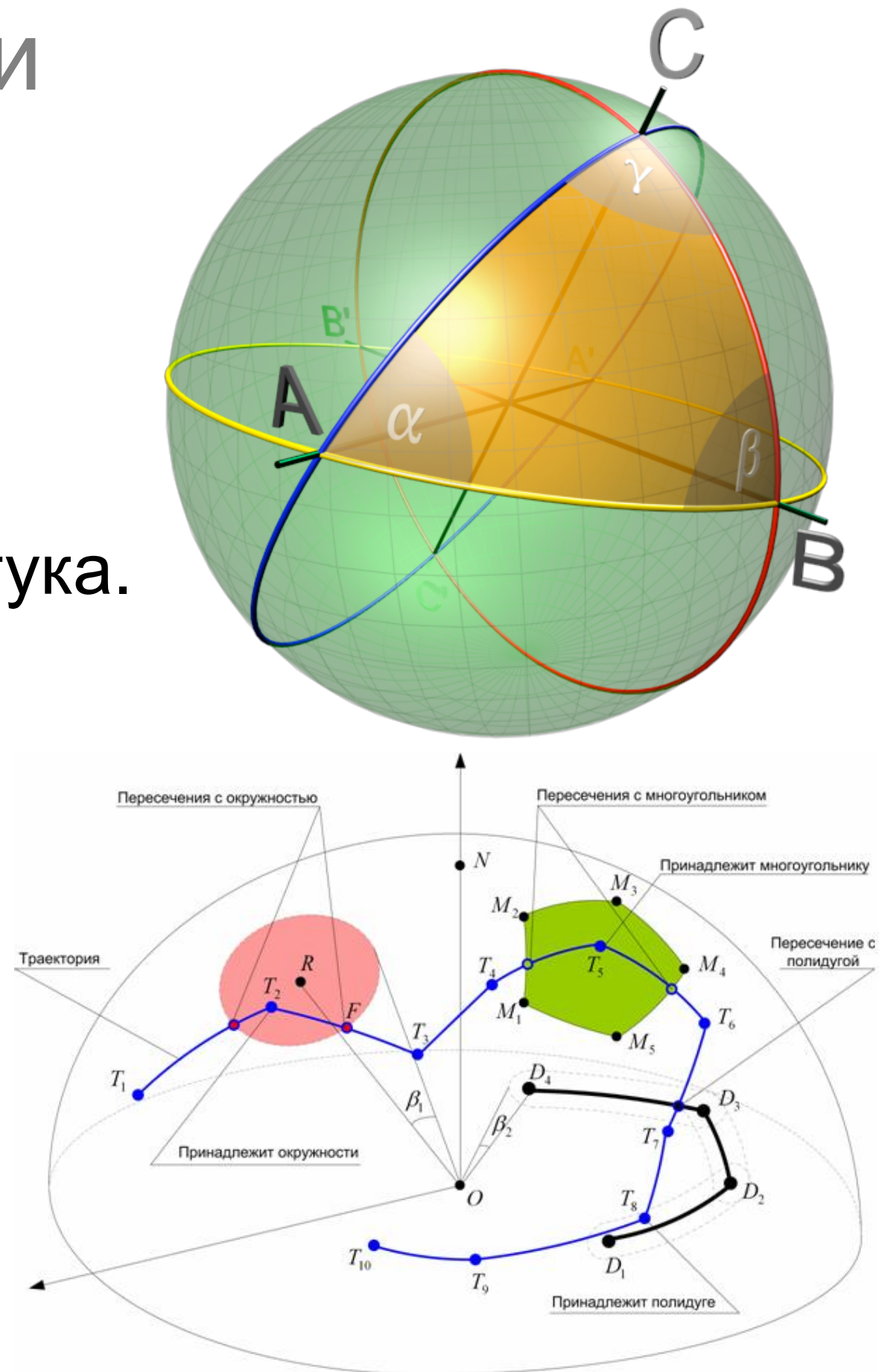
Многие сайты реализуют одни и те же действия:

- Показать/скрыть группу объектов
- Выбрать все объекты с определённым свойством
- Отсортировать объекты по критерию
- Найти ближайшие объекты
- Найти все объекты, попадающие в область (прямоугольник, круг, границы региона и т.д.)

Типичные операции

Однако операции с географическими сущностями – непростая штука.

- Как посчитать расстояния на сфере?
- Как посчитать попадание точки в сферический треугольник? А в многоугольник?



Типичные операции

В итоге мы получаем:

- кучу одинакового кода,
- который всё равно работает неправильно.

Что делать?

Что делать?



Хватит это терпеть

Исследования

Мы перебрали самые популярные вопросы, касающиеся работы с географическими объектами:

- Наш собственный клуб
<http://clubs.ya.ru/mapsapi>
- Stack Overflow
- Форумы MSDN
- Experts Exchange

Исследования

А также руками просмотрели несколько сотен крупных и средних сайтов с нашим API



Use-Cases

В итоге мы получили набор из 2 групп кейсов

- Геометрические операции
- Манипуляции над данными

Геометрические операции

- Вычисление границ и масштабов
- Пересечения геометрий
- Выборки: по типу геометрии, по ограничивающей области
- Сортировка по удалённости
- Центры и крайние точки геометрий
- Поиск ближайших к данной геометрии объектов

Манипуляции над данными

- Поиск по полям данных
- Фильтрация
- Сортировка
- Перебор
- Кластеризация
- Подписка на события
- Смена опций

ymaps.geoQuery

Выборки

Сортировки

Манипуляции

Геометрические операции

Инициализация

```
ymaps . geoQuery (что-то) ;
```

Конструируем выборку из:

- геообъектов карты
- JSON
- произвольной геометрии
- результата геокодирования или xml-файла (YMapsML, KML, GPX)

Выборка

```
var geoQueryResult =  
    ymaps.geoQuery(что-то) ;
```

Результат – объект типа `ymaps.GeoQueryResult`

- Набор геообъектов
- Можно итерироваться (`.each`), выбирать объекты (`.get`), добавлять и удалять объекты (`.add`, `.remove`)
- Можно добавлять объекты на карту и удалять их с карты
- Можно делать всякие крутые штуки!

Выборка

```
var geoQueryResult =  
    ymaps.geoQuery(что-то) ;
```

И ещё несколько интересных особенностей:

- Набор объектов неизменяем
- Операции асинхронны
- Операции чайнятся

Immutable

```
var geoQueryResult1 = ymaps.geoQuery(что-то) ,  
    geoQueryResult2 =  
        geoQueryResult1.filter(что-то) ,  
    geoQueryResult3 =  
        geoQueryResult2.remove(что-то) ;  
// в итоге geoQueryResult1 не изменился
```

- Ни одна операция не изменяет состав исходного `geoQueryResult`
- Каждая операция, изменяющая набор результатов, возвращает новый экземпляр `ymaps.GeoQueryResult`

Async

```
var geoQueryResult =  
    ymaps.geoQuery(ymaps.geoXml.load(<url>))  
    // Фильтр сработает после загрузки  
    .filter(<expression>)  
    // Сработает после выполнения  
    // всех предыдущих операций  
    .then(function (res) { soSomething() });
```

- Некоторые операции асинхронны всегда (геокодирование, загрузка xml)
- Некоторые операции синхронны, но не стоит на это полагаться
- Для выполнения кода после завершения всех действий используйте .then

Promises

```
ymaps.geoXml.load(<url>)
  .then(
    // обработчик положительного ответа
    function (res) { doSomething(); },
    // обработчик отрицательного ответа
    function (err) { doSomething(); }
  );
```

- Promises – замена обычным callback-ам. Они удобнее и функциональнее
- Мы имплементируем стандарт Promises/A в нашем ymaps.util.Promise
- ymaps.GeoQueryResult.then вернёт promise, который будет подтвержден после выполнения всех операций

Chaining

```
var geoQueryResult1 =  
    ymaps.geoQuery(map)  
        .filter(...)  
        .setOptions(...)  
        .addToMap(...)  
// Продолжать до полного удовлетворения
```

- Все операции, которые имеет смысл чайнить – чайнятся
- Чайнинг работает даже для асинхронных операций

Лучше один раз увидеть

```
var result = ymaps.geoQuery({
  type: 'FeatureCollection',
  features: [{
    type: 'Feature',
    geometry: {
      type: 'Circle',
      coordinates: [15, 15],
      radius: 100
    }
  }, {
    type: 'Feature',
    geometry: {
      type: 'LineString',
      coordinates: [[15, 16], [66, 23], [10, 12]]
    }
  }
]
}).addToMap(map);
```

Лучше второй раз увидеть

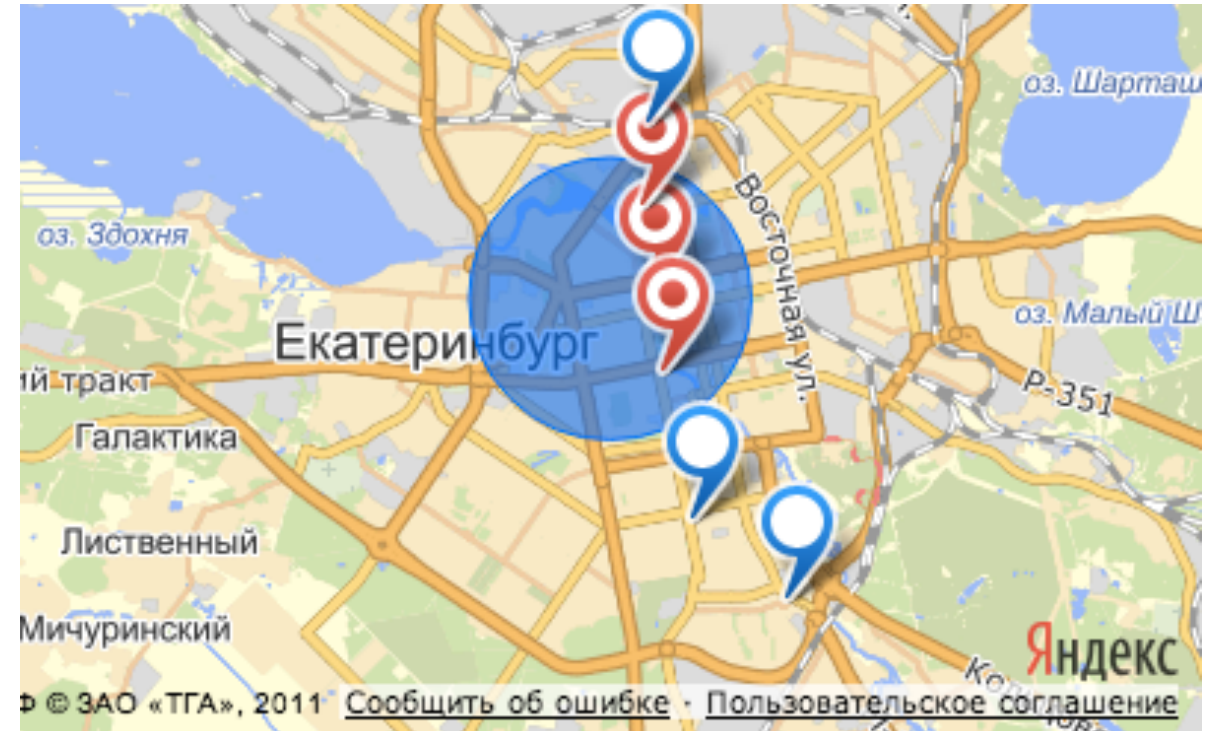
```
// Найдём Москву и Екатеринбург
// Приятный бонус – теперь можно чайнить
// асинхронные операции
var result = ymaps.geoQuery(
    ymaps.geocode( 'Москва' , { results: 1 } )
).add(
    ymaps.geocode( 'Екатеринбург' , { results: 1 } )
).then(function () {
    var moscow = result.get(0) ,
        ekat = result.get(1) ;
    // сделаем что-нибудь с полученным знанием
});
```

Вкусняшки



Что умеет geoQuery?

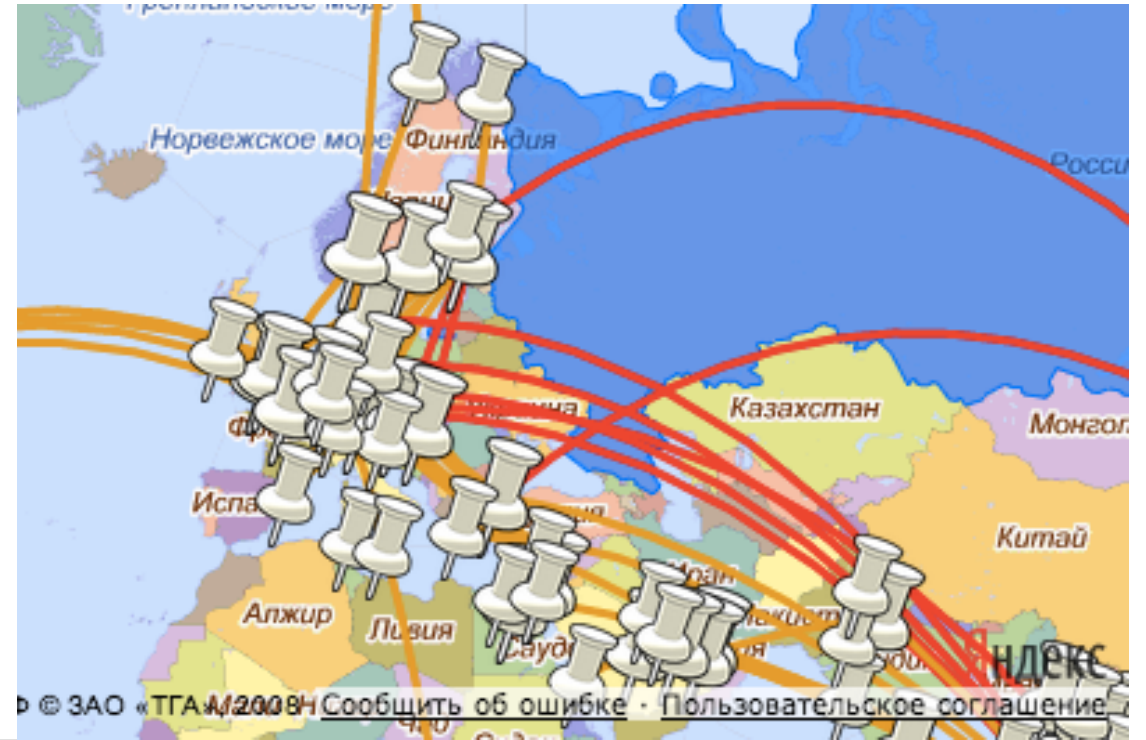
- Выбирать объекты, попадающие в указанную область на карте



```
// Например, загрузим список точек
// и подсветим те, которые попадают в круг
var circle = new ymaps.Circle([ center, 2000 ]);
myMap.geoObjects.add(circle);
ymaps.geoQuery(ymaps.geocode([56.8, 60.6], { kind: 'metro' }))
    .addToMap(myMap)
    .searchInside(circle)
    .setOptions({ preset: 'twirl#redDotIcon' });
```

Что умеет geoQuery?

- Выбирать объекты, пересекающиеся с другими объектами



```
// Например, загрузим траектории летящих самолётов
// и подсветим те, которые пересекают территорию России
ymaps.geoQuery(ymaps.geoXml.load(
    "http://openflights.org/demo/openflights-sample.kml"
))
.addToMap(myMap)
.searchIntersect(RUGeometry)
.setOptions({ strokeColor: "FF0000" });
```

Intermezzo: ymaps.regions

- В версии 2.0.31 мы добавили сервис "регионы", позволяющий работать с границами регионов России, Украины, Беларуси и Казахстана
- Данные берутся из OSM

```
ymaps.geoQuery(ymaps.regions.load(  
    // Код страны  
    "RU", {  
        // Язык подписей  
        lang: "ru",  
        // Уровень подробности границ  
        quality: 2  
    })  
    .addToMap(map)  
    .then(function () { doSomething(); }));
```

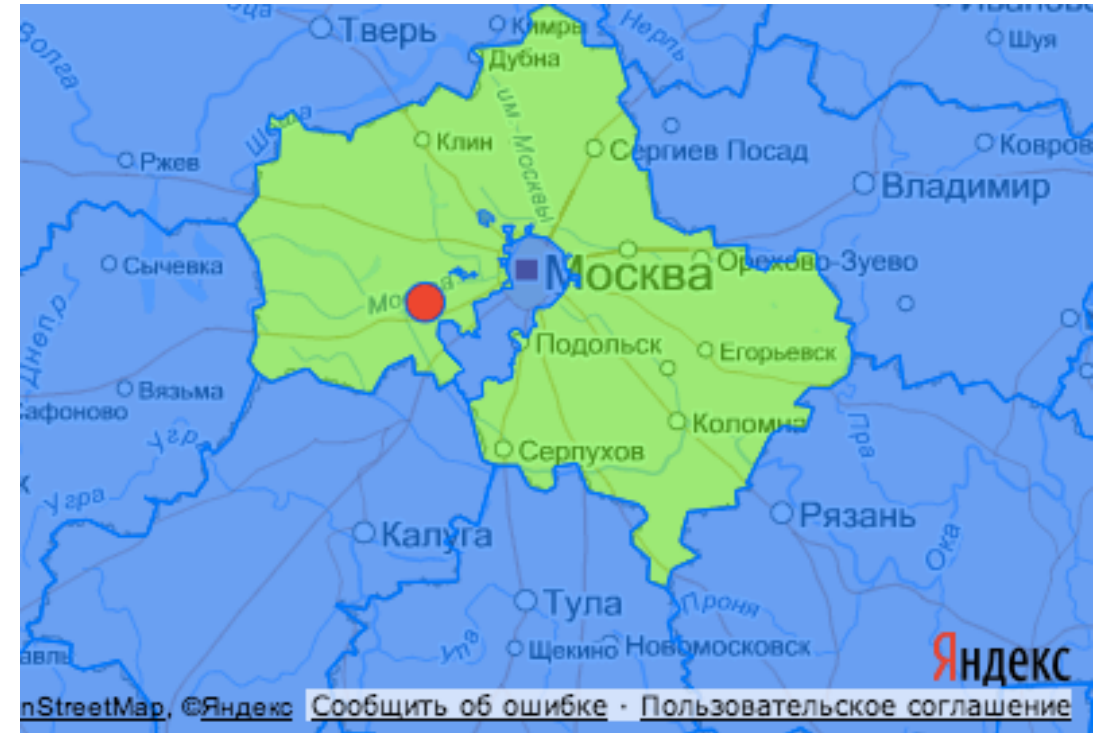
Intermezzo: ymaps.regions

Выглядит как-то так:



Что умеет geoQuery?

- Выбирать объекты, в которые попадает целиком другой объект



```
// Проверим, попадает ли интересующая нас область
// (например, круг) в какой-то из регионов целиком
var circle = new ymaps.Circle([[55.6, 36.7], 1e4], null, {
    fillColor: "FF0000", zIndex: 10000
});
myMap.geoObjects.add(circle);
ymaps.geoQuery(ymaps.regions.load("RU"))
    .addToMap(myMap)
    .searchContaining(circle)
    .setOptions({ fillColor: "00FF0080" });
```

Что умеет geoQuery?

- Сортировать объекты по удалённости от другого объекта



```
// Отсортируем регионы России по удалённости от Екатеринбурга
ymaps.geoQuery(ymaps.regions.load("RU"))
    .addToMap(myMap)
    .sortByDistance([60.6, 56.8])
    .each(function (region, index) {
        region.options.set("opacity", index/100);
    });
```

Что умеет geoQuery?

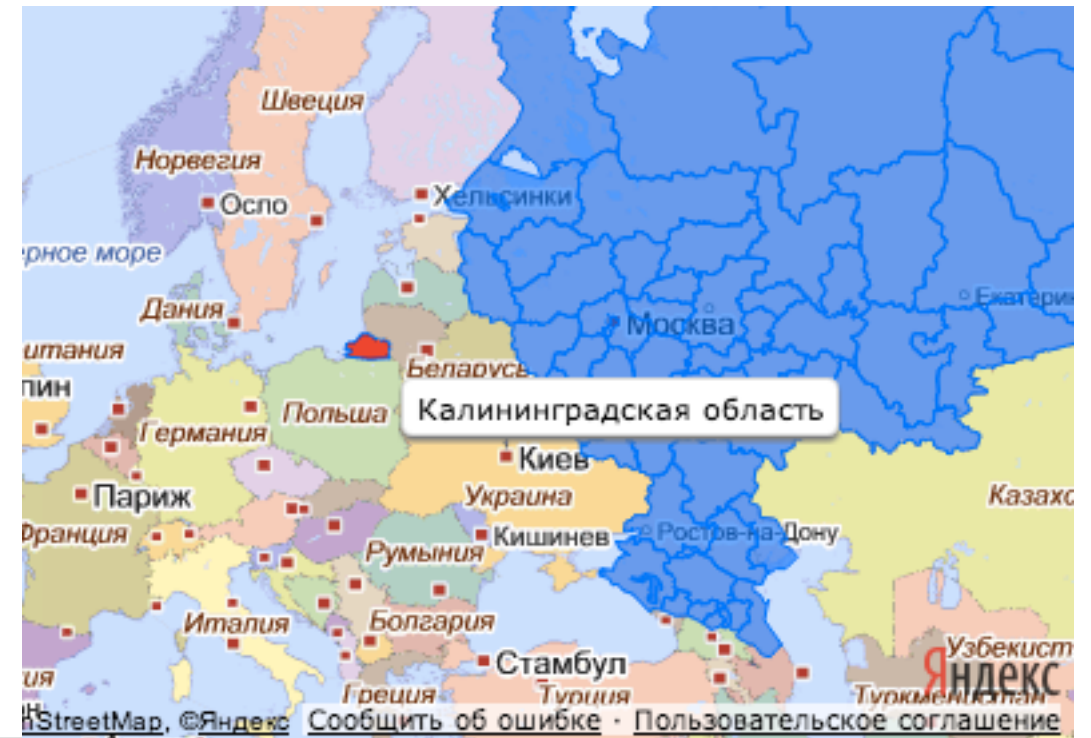
- Фильтровать объекты по критерию



```
// Покажем на карте регионы, в названии которых
// есть слово «область»
ymaps.geoQuery(ymaps.regions.load("RU"))
    // Можно фильтровать по: координатам, свойствам, опциям
    // И даже писать регулярки
    .search("properties.name RLIKE '(o|O)бласть'")
    .addToMap(myMap);
```

Что умеет geoQuery?

- Находить «центр» объекта и его крайние точки



```
// Например, подсветим самый западный регион России
var regions = ymaps.geoQuery(ymaps.regions.load("RU"))
    .addToMap(myMap).then(function () {
        westRegion = regions.getExtremeObject("left");
        westRegion.options.set({ fillColor: "FF0000" });
    });
```

Что умеет geoQuery?

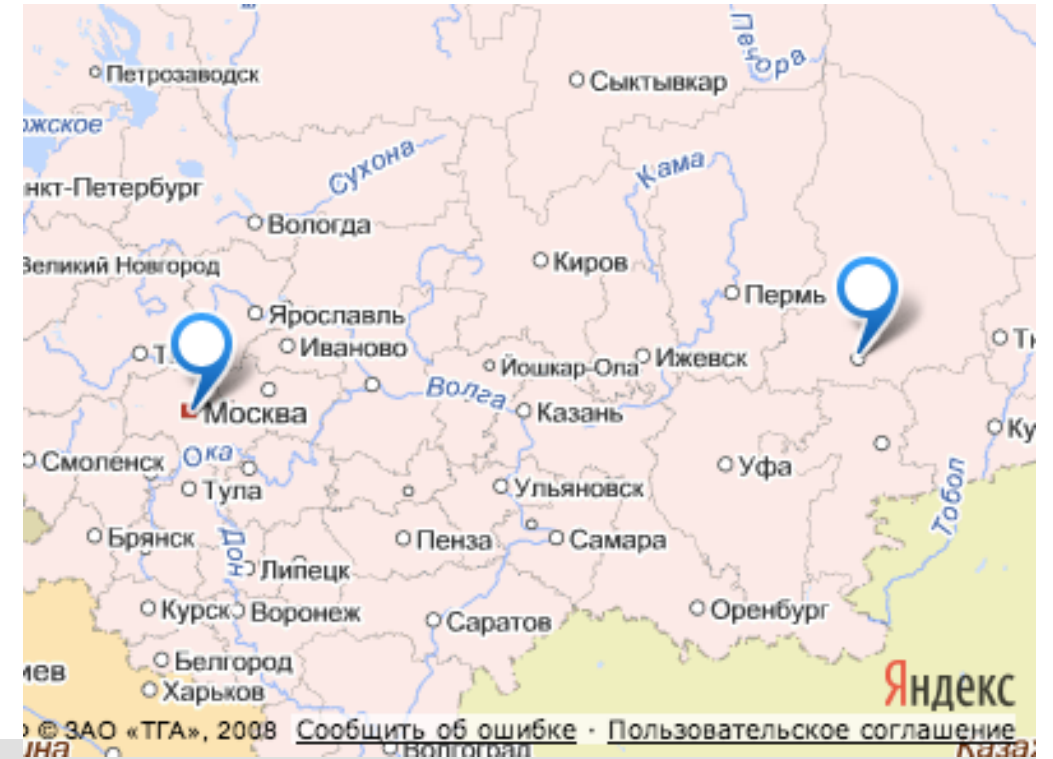
- Кластеризовать



```
// Например, загрузим траектории летящих самолётов,  
// возьмём все точки отправлений и прибытий и кластеризуем  
var clusters = ymaps.geoQuery(ymaps.geoXml.load(  
    "http://openflights.org/demo/openflights-sample.kml"  
))  
    .search("geometry.type == 'Point'")  
    .clusterize();  
map.geoObjects.add(clusters);
```

Что умеет geoQuery?

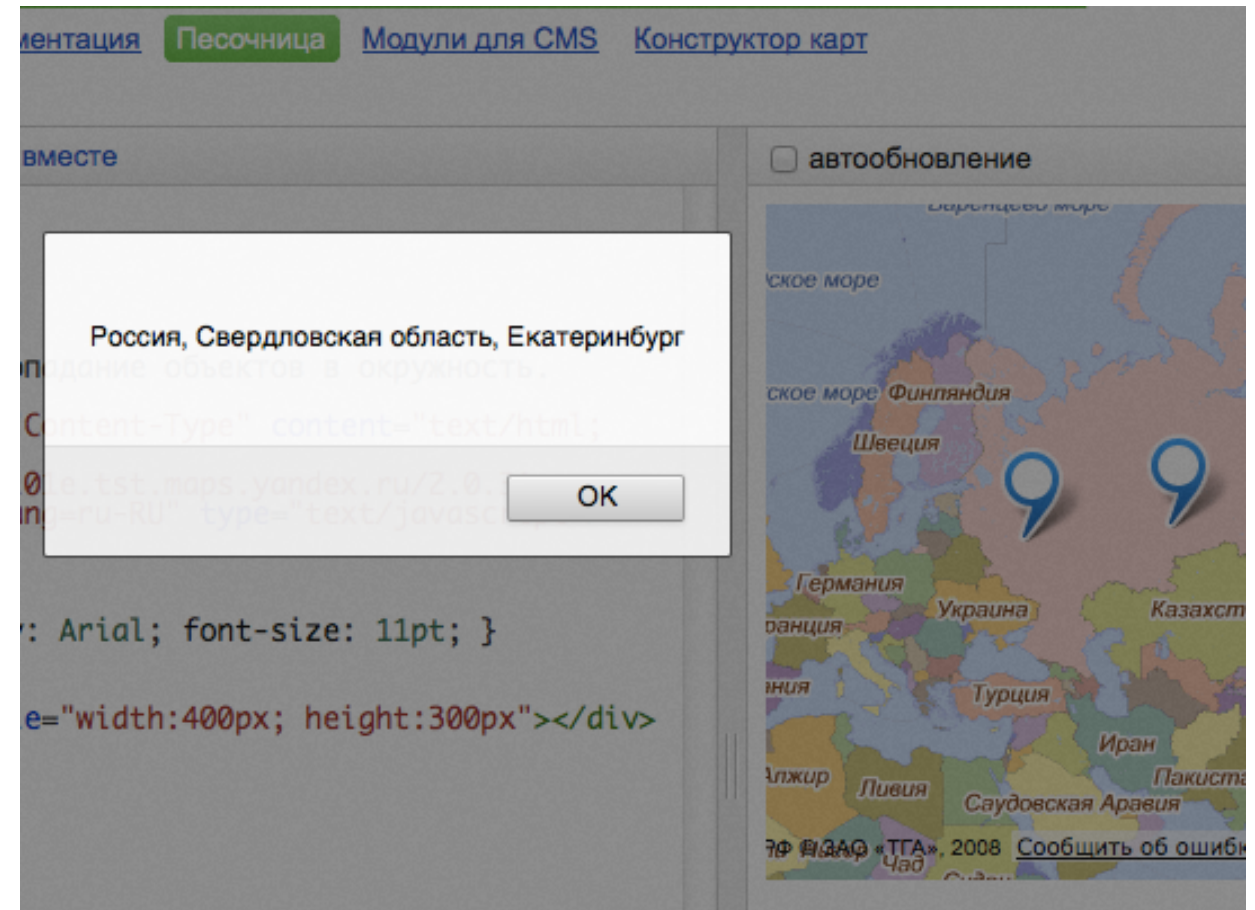
- Находить оптимальный центр и масштаб карты



```
// Например, найдём Москву и Екатеринбург и покажем их
// на карте
ymaps.geoQuery(
    ymaps.geocode("Москва", { results: 1 }))
    .add(ymaps.geocode("Екатеринбург", { results: 1 }))
    .addToMap(map)
    .applyBoundsToMap(map)
```

Что умеет geoQuery?

- Подписываться на события и отписываться от них



```
// Поставим метки «Москва» и «Екатеринбург»  
// и будем по клику выводить alert-ом полное описание  
ymaps.geoQuery(ymaps.geocode("Москва", { results: 1 })))  
  .add(ymaps.geocode("Екатеринбург", { results: 1 })))  
  .addToMap(map)  
  .addEvents("click", function (e) {  
    alert(e.get("target").properties.get("text"));  
  });
```

На правах рекламы: песочница

Заходите: <http://api.yandex.ru/maps/jsbox/>

[API Карт:](#) [Возможности](#) [Документация](#) [Песочница](#) [Модули для CMS](#) [Конструктор карт](#) [Поделиться...](#)

Карта

Геообъекты

GeoQuery

Попадание точек в окружность

Поиск ближайшего объекта

Добавление на карту объектов, попадающих в область видимости

Пересечение маршрута с многоугольником на примере МКАД

Фильтрация объектов на карте по различным признакам

Позиционирование карты для показа результатов геокодирования и их кластеризация

Поведения и события

Макеты и шаблоны

Поиск по карте

[Пользовательское соглашение](#)

[Обратная связь](#)

© 2012 «Яндекс»

JavaScript

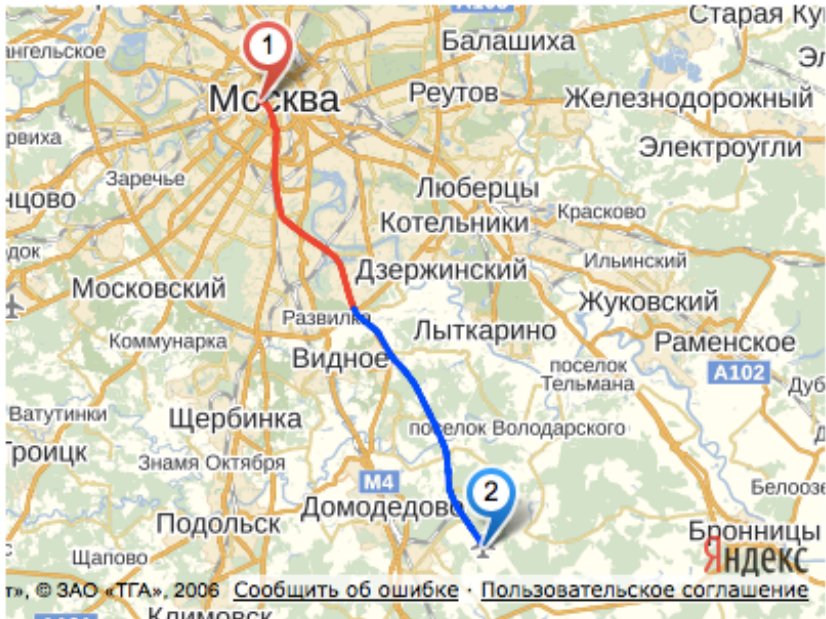
HTML

Всё вместе

```
1 ymaps.ready(init);
2
3 function init() {
4     var myMap = new ymaps.Map("map", {
5         center: [55.73, 37.75],
6         zoom: 9
7     });
8     moscowPolygon;
9
10    function onPolygonLoad (json) {
11        moscowPolygon = new ymaps.Polygon(json.coordinates);
12        // Если мы не хотим, чтобы контур был виден,
13        // зададим соответствующую опцию.
14        moscowPolygon.options.set('visible', false);
15        // Чтобы корректно осуществлялись геометрические
16        // операции
17        // над спроецированным многоугольником, его нужно
18        // добавить на карту.
19        myMap.geoObjects.add(moscowPolygon);
20
21        ymaps.route(["метро Арбатская", "аэропорт Домодедово"]).then(
22            function (res) {
23                // Объединим в выборку все сегменты
24                // маршрута.
25                var pathsObjects =
26                ymaps.geoQuery(res.getPaths()),
27                edges = [];
28
29                // Переберем все сегменты и разобьем их на
30                // отрезки.
31                pathsObjects.each(function (path) {
32                    var coordinates =
33                    path.geometry.getCoordinates();
34                    for (var i = 1, l =
35                    coordinates.length; i < l; i++) {
36                        edges.push({
37                            type: 'LineString',
38                            coordinates: [coordinates[i],
39                            coordinates[i - 1]]
40                        });
41                    }
42                });
43            }
44        );
45    }
46    ymaps.on(moscowPolygon, 'load', onPolygonLoad);
47 }
```

☐ автообновление

Маршрут внутри МКАД помечен красным, снаружи - синим



Используются [ready](#), [Map](#), [Polygon](#), [route](#), [geoQuery](#), [GeoQueryResult](#)

Часто возникает задача проложить маршрут между заданными точками, а затем проанализировать, в какие области или города попадают сегменты этого маршрута. В данном примере по задан...

Там ещё много клёвых штук!



© СОЮЗМУЛЬТФИЛЬМ

Будут вопросы — пишите!



api.yandex.ru/maps



clubs.ya.ru/mapsapi



facebook.com/ymapsapi



Сергей Константинов

Руководитель группы
разработки
API Яндекс.Карт

twirl@yandex-team.ru

api.yandex.ru/maps

clubs.ya.ru/mapsapi

facebook.com/ymapsapi

Спасибо!