Информационные технологии в корпоративных сетях

API

Application Programming Interface

Интерфейс программирования приложений
Интерфейс прикладного программирования

К.т.н., доцент
Феликс Васильевич Филиппов

Что такое API ?

API (интерфейс программирования приложений) — набор готовых классов, процедур, функций, структур и констант, предоставляемых сервисом (библиотекой) или операционной системой для использования во внешних программных продуктах.

Используется в качестве инструментального средства при написании всевозможных приложений.

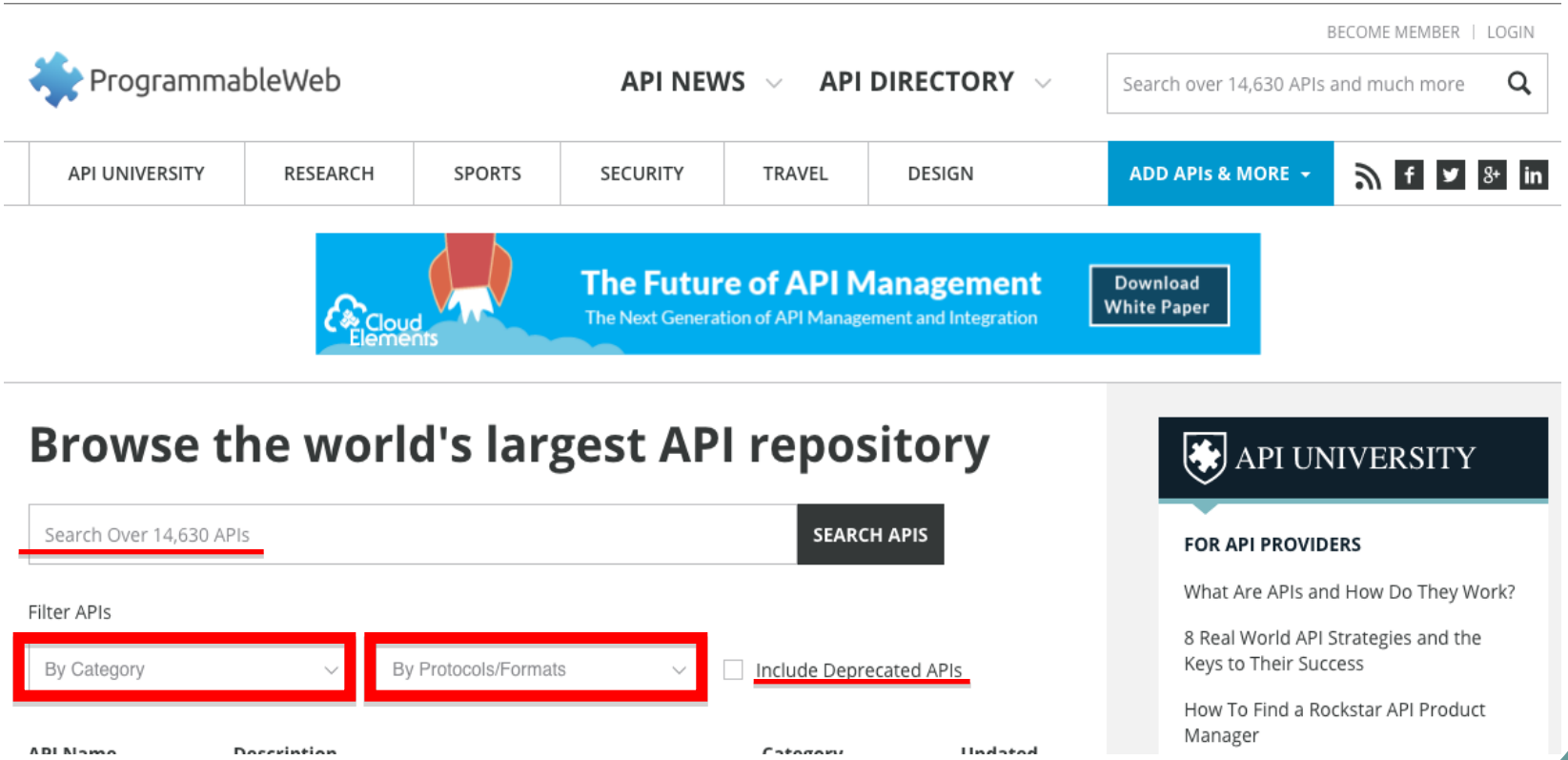
Как использовать API ?

Практика разработки API позволила веб-сообществам создать открытую архитектуру для совместного использования контента и данных. Тем самым содержимое, которое создается в одном месте, может динамически обновляться и размещаться во многих местах в вебе.

При использовании в контексте веб-разработки, API определяется как протокол передачи сообщений запроса (HTTP), наряду с определением структуры ответных сообщений, которые как правило, выдаются в формате XML или JSON.

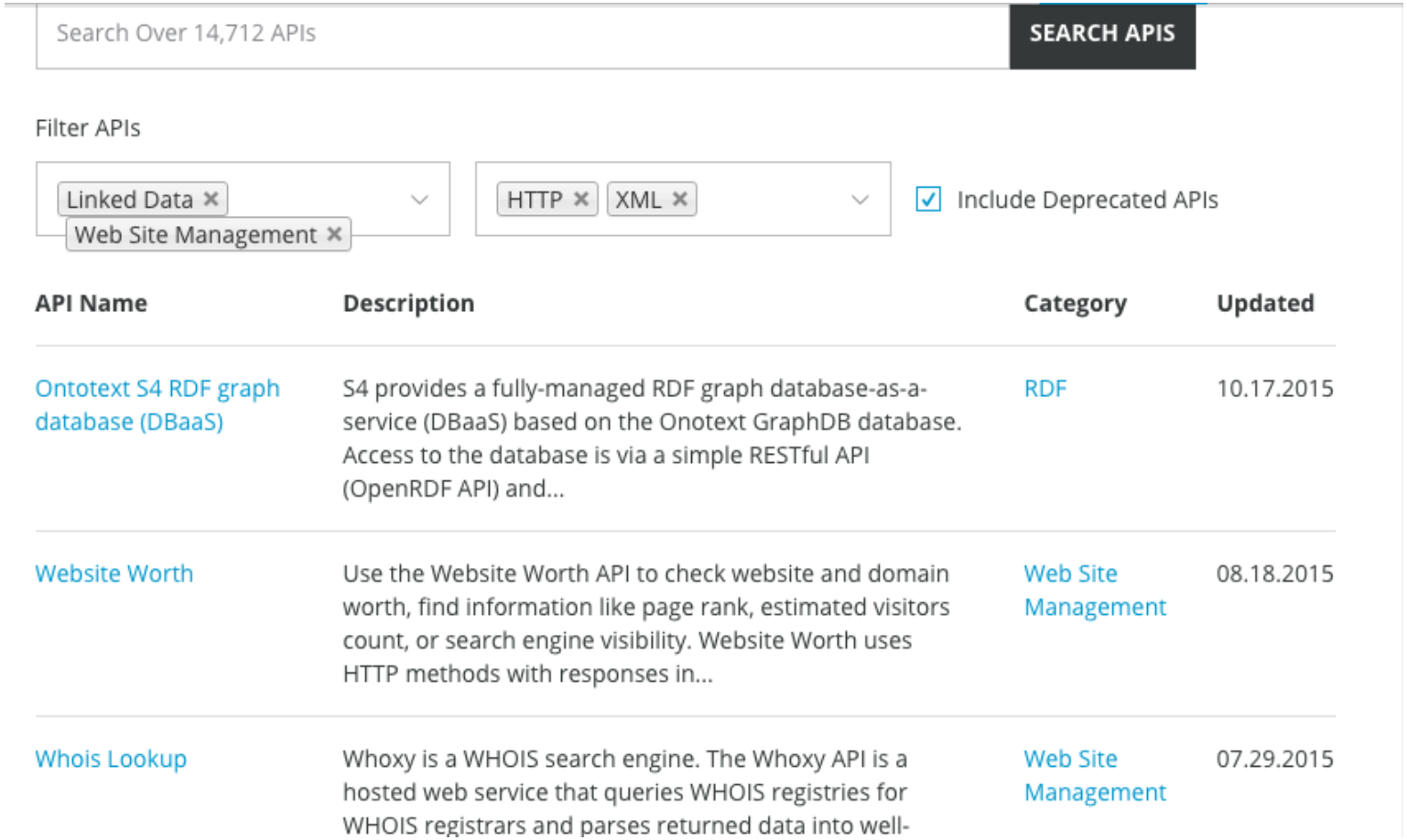
Поиск информации через API

Прежде всего находим нужный API:
http://www.programmableweb.com/apis/directory

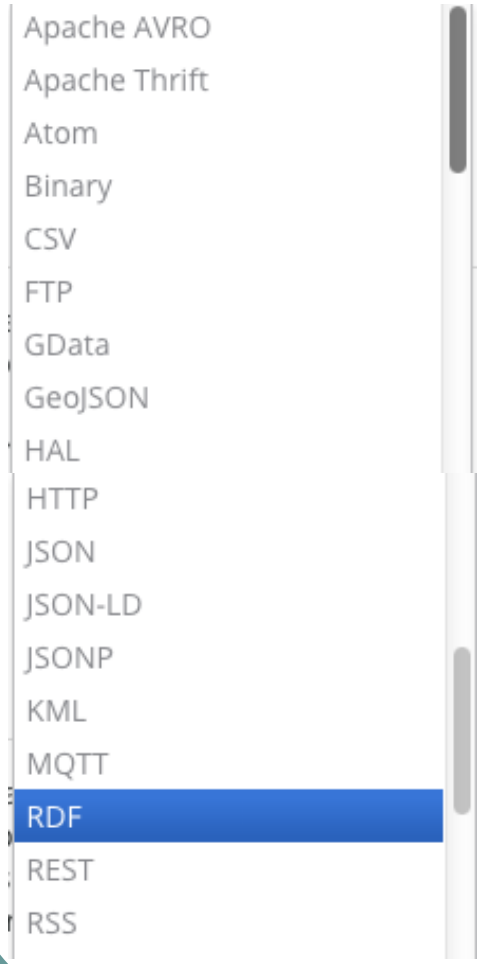


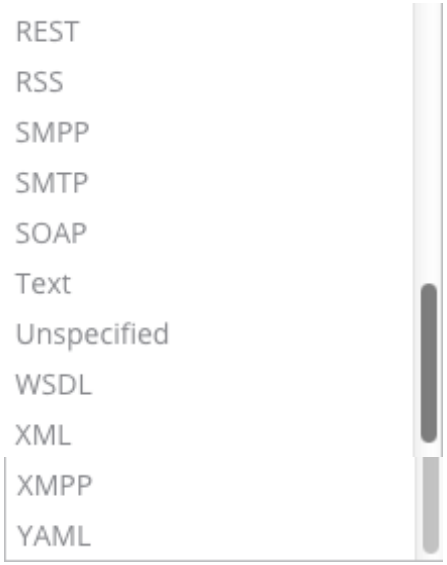
Google maps наиболее популярный.

API: категории, протоколы, форматы



API: протоколы, форматы





1 API географические карты

API карты Google

http://www.jose-gonzalez.org/using-google-maps-api-r/#.Ux2LEflgy3M

Два API, представляющие интерес:
- Географическое положение
- Статические карты

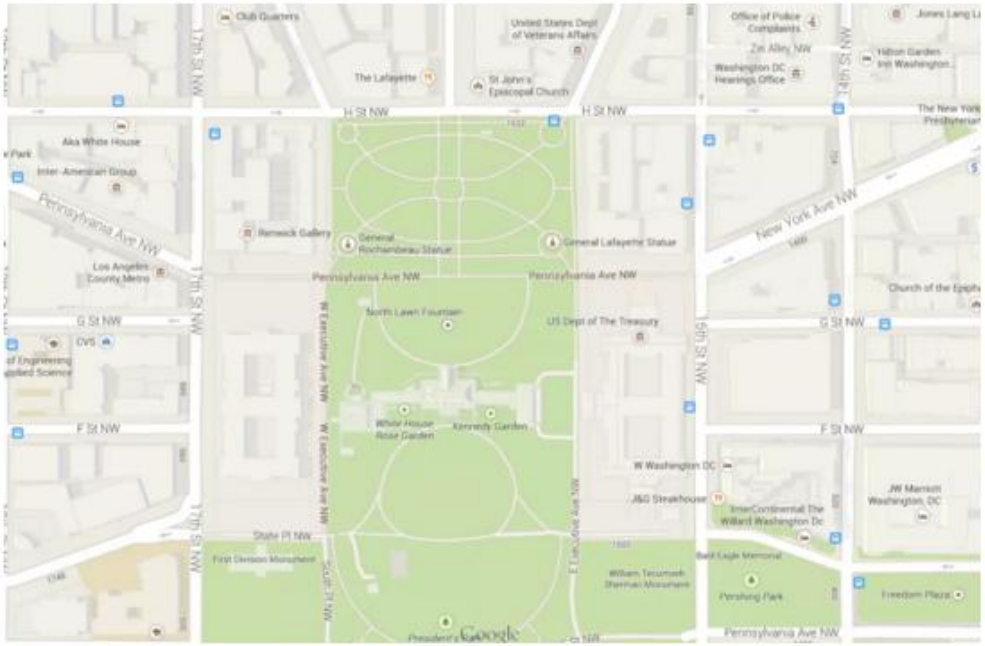
Что дают эти API?

https://developers.google.com/maps/documentation/geocoding/
https://developers.google.com/maps/documentation/staticmaps/

API карты Google в RStudio

http://www.jose-gonzalez.org/using-google-maps-api-r/#.Ux2LEflgy3M

Using Google maps API and R

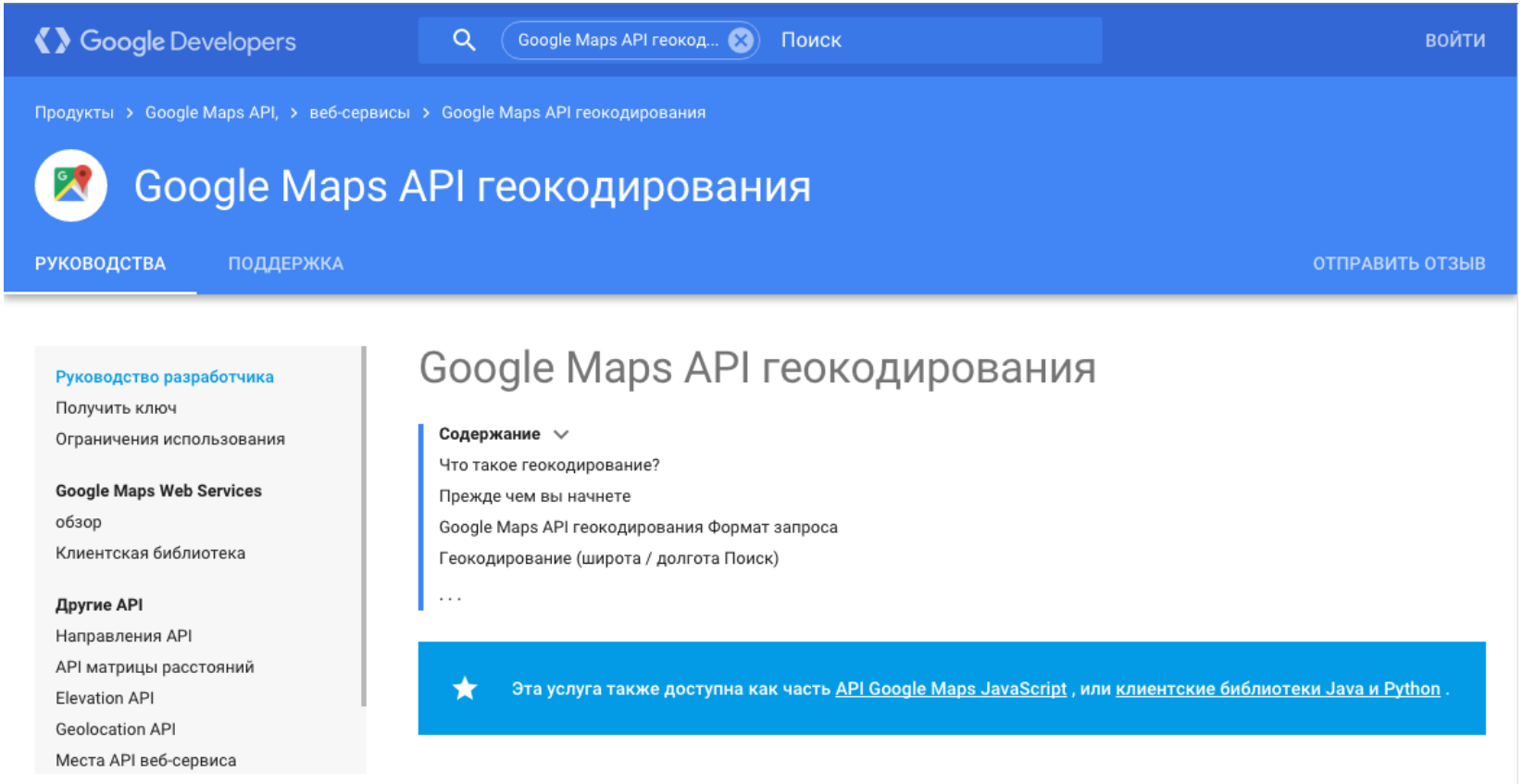


This post shows how to use Google Maps' API with R making some tweaks to this function. Combine the first part with sapply or Plyr and it becomes a very powerful tool in just a few lines of code. You can find a gist in RMarkdown with the code here or click below to continue reading.

```
#### This script uses RCurl and RJSONIO to download data from Google's API:
#### Latitude, longitude, location type (see explanation at the end), formatted address
```

Географическое положение

https://developers.google.com/maps/documentati



Необходимые параметры: address [place] и sensor [=false]

Пример запроса геокоординат

Запрос:

- https://maps.googleapis.com/maps/api/geocode/json?
- address=Kremlin,Moscow
- sensor=false

Параметры разделяются '&'

https://maps.googleapis.com/maps/api/geocode/json?address=Kremlin,Moscow&sensor=false

Какое из полей JSON нас интересует?

Результат запроса:

https://maps.googleapis.com/maps/api/geocode/json?address=Kremlin,Moscow&sensor=false


```
{
 "results" : [
   {
      "address_components" : [
         {
            "long_name" : "Московский Кремль",
            "short_name" : "Московский Кремль",
            "types" : [ "point_of_interest", "establishment" ]
         },
         {
            "long_name" : "Центральный административный округ",
            "short_name" : "Центральный административный округ",
            "types" : [ "sublocality_level_1", "sublocality", "political" ]
         },
         {
            "long_name" : "Москва",
            "short_name" : "Москва",
            "types" : [ "locality", "political" ]
         },
         {
            "long_name" : "город Москва",
            "short_name" : "г. Москва",
            "types" : [ "administrative_area_level_2", "political" ]
         },
         {
            "long_name" : "город Москва",
            "short_name" : "г. Москва",
            "types" : [ "administrative_area_level_1", "political" ]
         },
         {
            "long_name" : "Россия",
            "short_name" : "RU",
            "types" : [ "country", "political" ]
         },
         {
            "long_name" : "103073",
            "short_name" : "103073",
            "types" : [ "postal_code" ]
         }
      ],
      "formatted_address" : "Московский Кремль, Москва, Россия, 103073",
      "geometry" : {
         "location" : {
            "lat" : 55.7520233,
            "lng" : 37.6174994
         },
         "location_type" : "APPROXIMATE",
         "viewport" : {
            "northeast" : {
               "lat" : 55.75337228029149,
               "lng" : 37.6188483802915
            },
            "southwest" : {
               "lat" : 55.75067431970849,
               "lng" : 37.6161504197085
            }
         }
      },
      "place_id" : "ChIJc-UVs1BKtUYRaC6bPVq_hqg",
      "types" : [ "museum", "point_of_interest", "establishment" ]
   }
 ],
 "status" : "OK"
}
```

Результат запроса:

https://maps.googleapis.com/maps/api/geocode/json?address=Kremlin,Moscow&sensor=false

results[[1]]$geometry$location["lat"]

results[[1]]$geometry$location["lng"]

Запрос геокоординат в RStudio

Напишем функцию getUrl():

```
getUrl <- function(address,sensor = "false") {
 root <- "http://maps.google.com/maps/api/geocode/json?"
 u <- paste0(root,"address=", address, "&sensor=false")
 return(URLencode(u))
}
getUrl("Kremlin, Moscow")
```

Используем функцию getUrl():

```
require(RJSONIO)
target <- getUrl("Kremlin, Moscow")
dat <- fromJSON(target)
latitude <- dat$results[[1]]$geometry$location["lat"]
longitude <- dat$results[[1]]$geometry$location["lng”]
```

Получение карты

https://developers.google.com/maps/documentation/staticmaps/

Формирование URL:

base=“http://maps.googleapis.com/maps/api/staticmap?”“
center= “latitude (55.75), longitude (37.62)”
zoom (1 - 18)
maptype="hybrid”
suffix = “&size=800x800&sensor=false&format=png”“

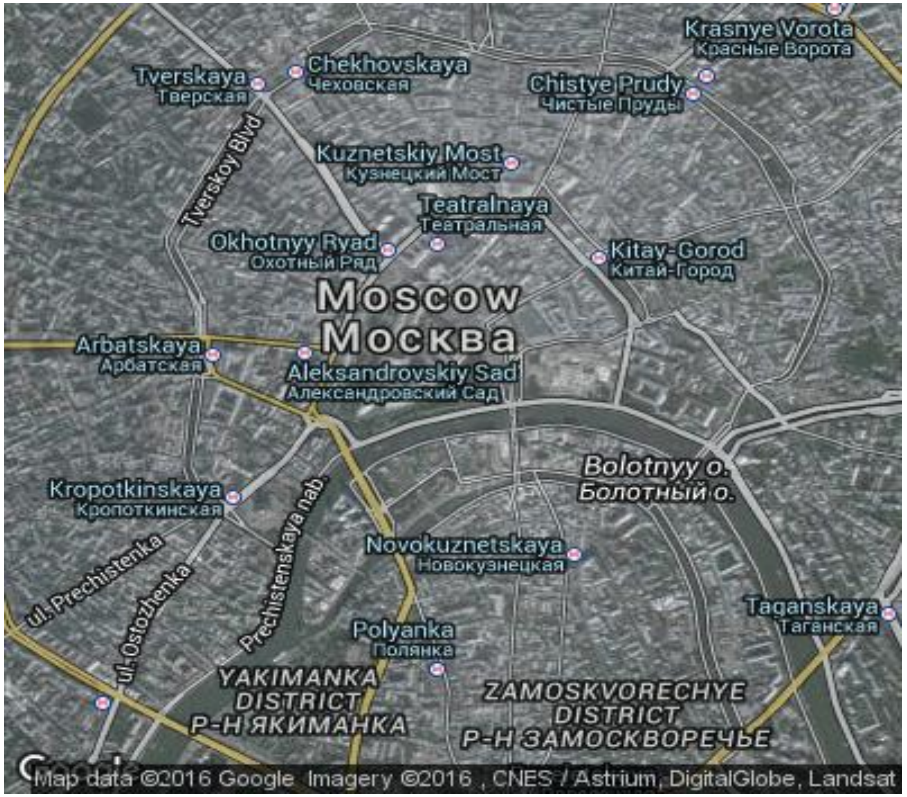
Формирование URL карты в RStudio

```
base="http://maps.googleapis.com/maps/api/staticmap?center="
latitude=55.75
longitude=37.62
zoom=13
maptype="hybrid"
suffix ="&size=400x400&sensor=false&format=png"

target <- paste0(base,latitude,",",longitude,
                 "&zoom=",zoom,"&maptype=",maptype,suffix)
```

Загрузка карты lat=55,75 lon=37,62

download.file(target,"test13.png", mode = "wb")



zoom=13

Загрузка карты lat=55,75 lon=37,62

download.file(target,"test15.png", mode = "wb")

zoom=15

Использование карт с API Yandex

Yandex карты имеют очень похожий синтаксис с картами Google.

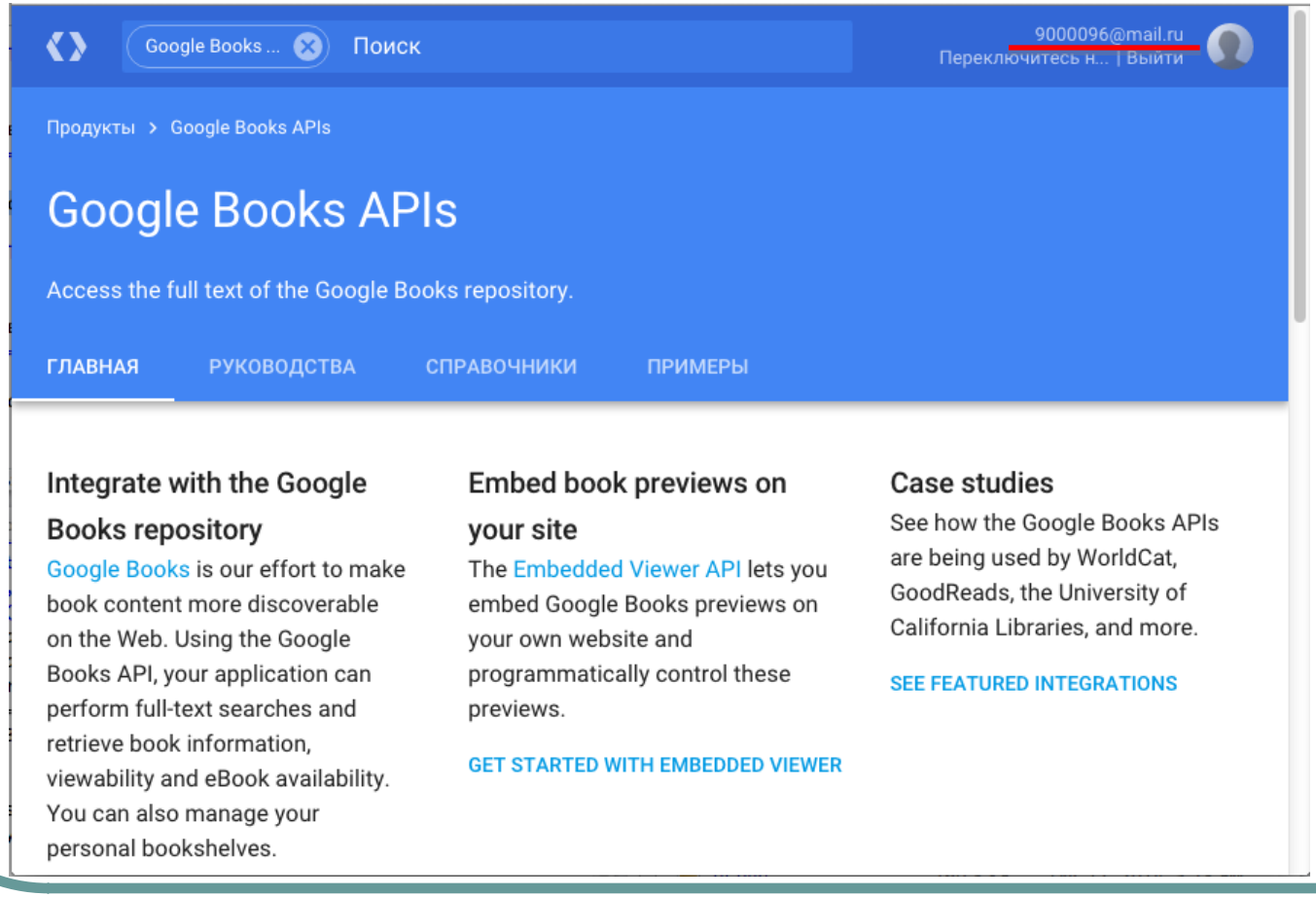
Чтобы построить сценарии с API для карт Yandex, нужно использовать информацию из:

http://api.yandex.com/maps/doc/staticapi/1.x/dg/concepts/input_params.xml

2 API
Google книги

Использование API Google книги

https://developers.google.com/books/docs/v1/getting



Пример использования API Google книги

https://www.googleapis.com/books/v1/volumes?q=harry+potter&callback=handleResponse

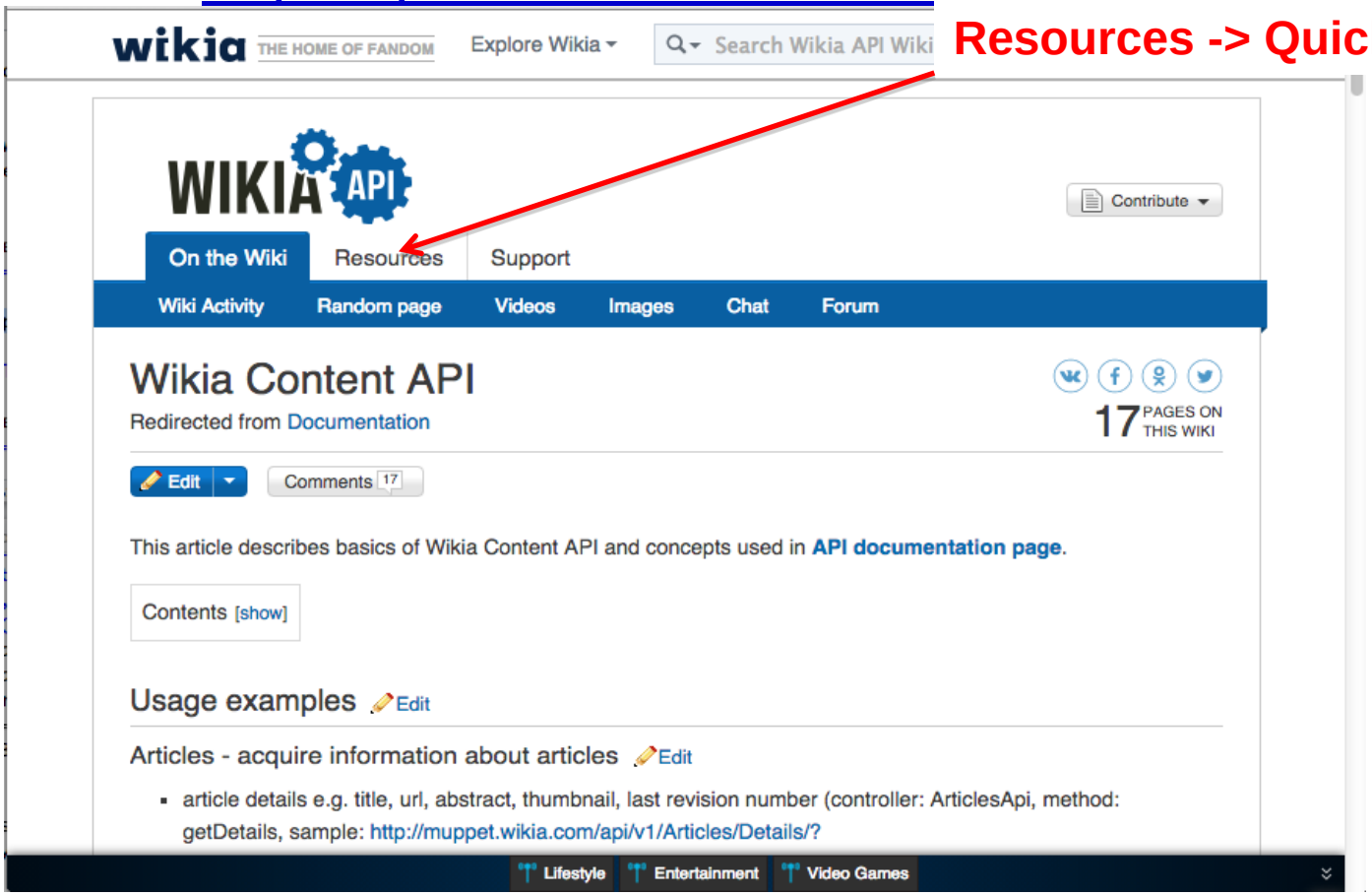

```
    "title": "Гарри Поттер и узник Азкабана",
    "authors": [
     "J. K. Rowling"
    ],
    "publisher": "Rosman",
    "publishedDate": "2002",
    "description": "Открой эту книгу - и ты окунешься в мир чародейства и волшебства, где оживают мифические
чудовища, где школьный учитель - оборотень и где даже время можно повернуть вспять",
```

```
    "title": "Гарри Поттер и Философский Камень",
    "subtitle": "",
    "authors": [
     "Джоан Роулинг"
    ],
    "publisher": "Lires",
    "publishedDate": "2015-05-10",
    "description": "Одиннадцатилетний мальчик-сирота Гарри Поттер живет в семье своей тетки и даже не
подозревает, что он – настоящий волшебник. Но однажды прилетает сова с письмом для него, и жизнь Гарри Поттера
изменяется навсегда. Он узнает, что зачислен в Школу чародейства и волшебства, выясняет правду о загадочной
смерти своих родителей, а в результате ему удается раскрыть секрет философского камня.",
```

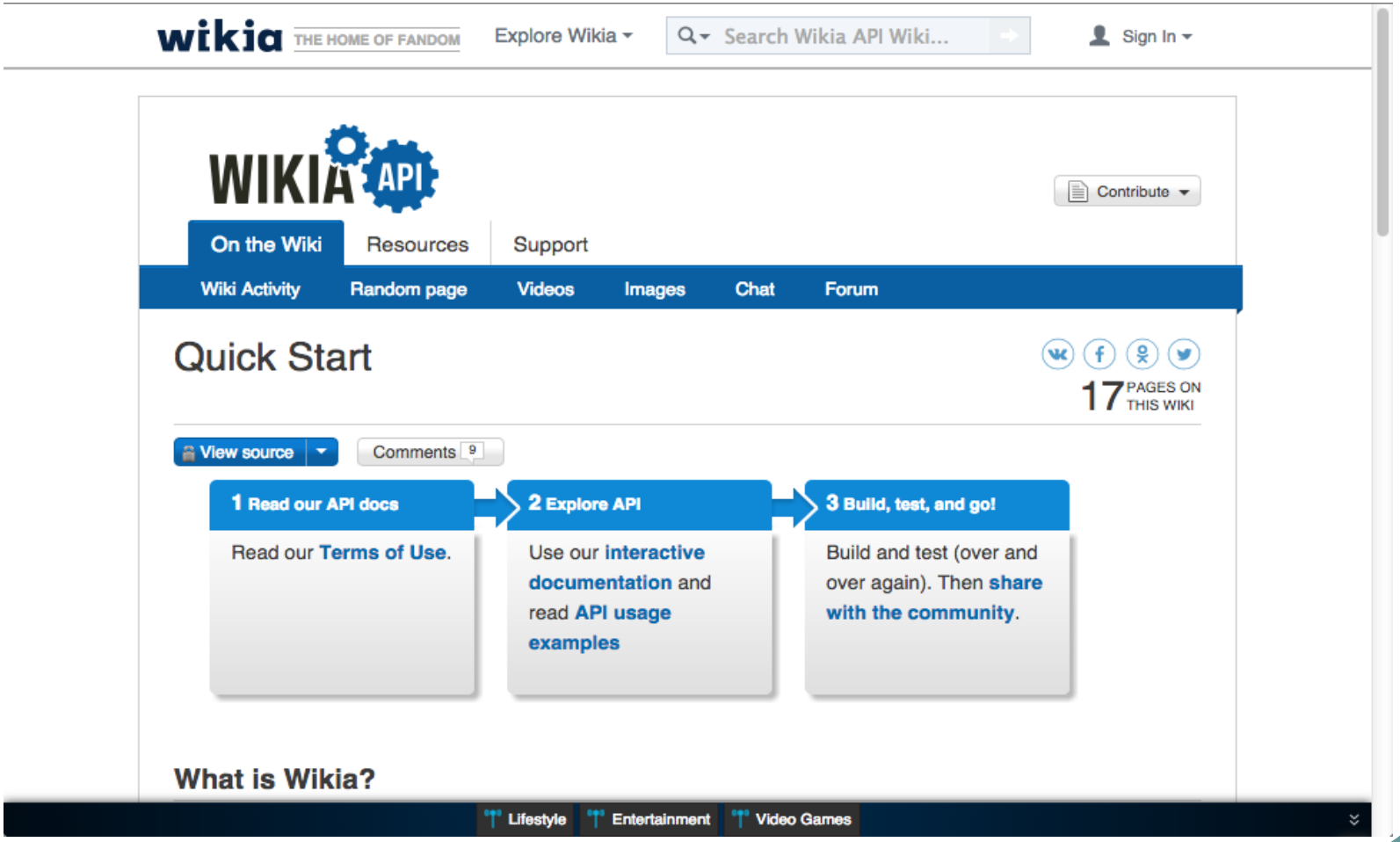
3 API Wiki

Использование API Wiki

http://api.wikia.com/wiki/Documentation



Использование API Wiki



Пример использования API Wiki

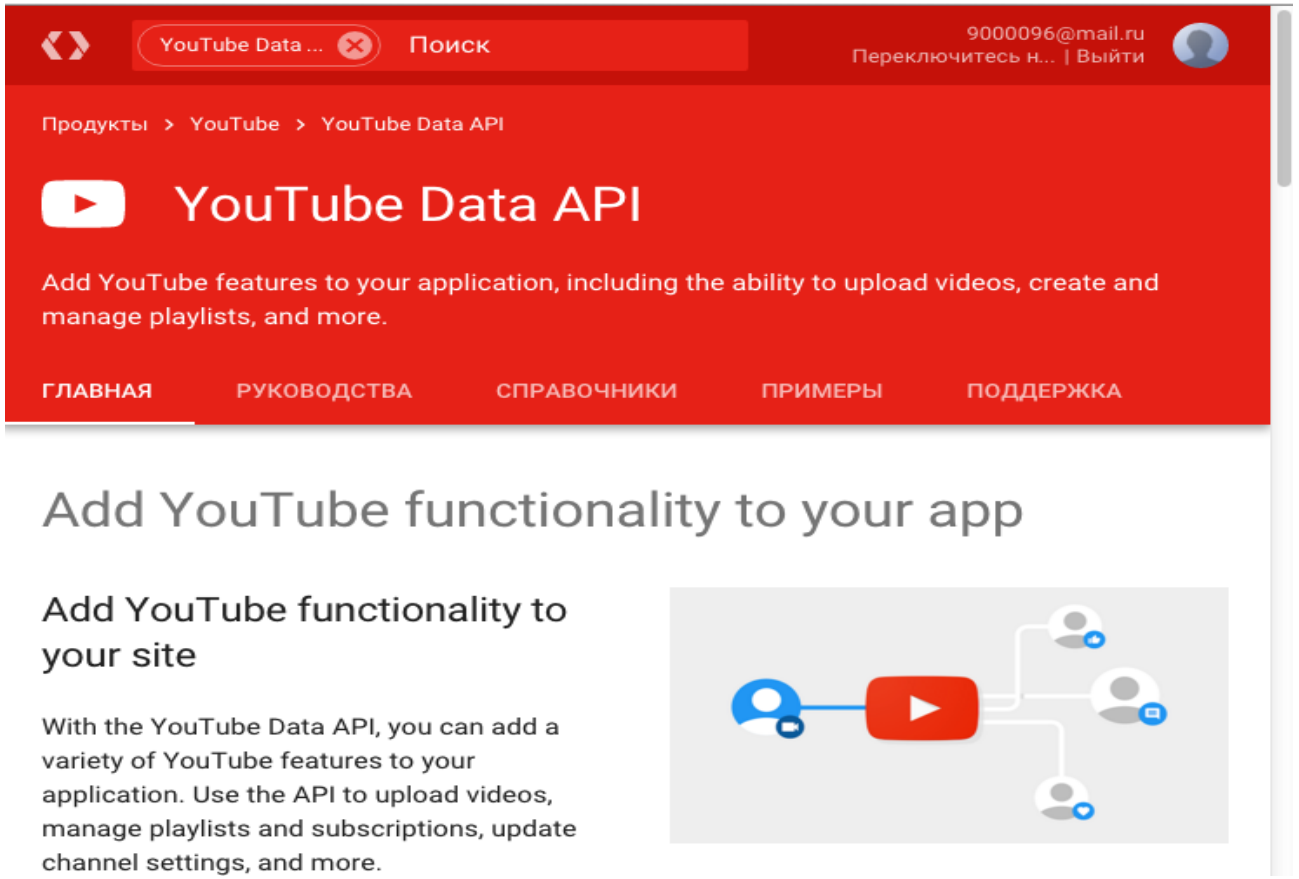
```
* action=query *
  Query API module allows applications to get needed pieces of data from the MediaWiki databases,
  and is loosely based on the old query.php interface.
  All data modifications will first have to use query to acquire a token to prevent abuse from malicious sites

This module requires read rights
Parameters:
  titles             - A list of titles to work on
  pageids            - A list of page IDs to work on
                       Maximum number of values 50 (500 for bots)
  revids             - A list of revision IDs to work on
                       Maximum number of values 50 (500 for bots)
  prop               - Which properties to get for the titles/revisions/pageids. Module help is available below
                       Values (separate with '|'): info, revisions, links, iwlinks, langlinks, images, imageinfo, stashimageinfo, templates,
                           categories, extlinks, categoryinfo, duplicatefiles, pageprops, wklasteditors, globalusage, infobox
  list               - Which lists to get. Module help is available below
                       Values (separate with '|'): allimages, allpages, alllinks, allcategories, allusers, backlinks, blocks, categorymembers,
                           deletedrevs, embeddedin, filearchive, imageusage, iwbacklinks, langbacklinks, logevents,
                           recentchanges, search, tags, usercontribs, watchlist, watchlistraw, exturlusage, users, random,
                           protectedtitles, querypage, wkdomains, wkpoppages, wkvoteart, wkaccessart, wkeditpage, wkedituser,
                           wkmostvisit, checkuser, checkuserlog, abuselog, abusefilters, categoryintersection,
                           unconvertedinfoboxes, allinfoboxes
  meta               - Which metadata to get about the site. Module help is available below
                       Values (separate with '|'): siteinfo, userinfo, allmessages
  generator          - Use the output of a list as the input for other prop/list/meta items
                       NOTE: generator parameter names must be prefixed with a 'g', see examples
                       One value: links, images, templates, categories, duplicatefiles, allimages, allpages, alllinks, allcategories,
                           backlinks, categorymembers, embeddedin, imageusage, iwbacklinks, langbacklinks, recentchanges,
                           search, watchlist, watchlistraw, exturlusage, random, protectedtitles, querypage,
                           categoryintersection
  redirects          - Automatically resolve redirects
  converttitles      - Convert titles to other variants if necessary. Only works if the wiki's content language supports variant conversion.
                       Languages that support variant conversion include gan, iu, kk, ku, shi, sr, tg, zh
  indexpageids       - Include an additional pageids section listing all returned page IDs
  export             - Export the current revisions of all given or generated pages
  exportnowrap       - Return the export XML without wrapping it in an XML result (same format as Special:Export). Can only be used with export
  iwurl              - Whether to get the full URL if the title is an interwiki link
Examples:
  api.php?action=query&prop=revisions&meta=siteinfo&titles=Main%20Page&rvprop=user|comment
  api.php?action=query&generator=allpages&gapprefix=API/&prop=revisions
```

4 API YouTube

Использование API YouTube

https://developers.google.com/youtube/v3/



5 API Требующие установки

API требующие установки

Гнилые помидоры
http://developer.rottentomatoes.com/docs

Переводчик (более 50 языков)
http://blogs.msdn.com/b/translation/p/gettingstarted1.aspx

Погода
http://www.worldweatheronline.com/free-weather-feed.aspx
https://developer.forecast.io/

Музыка
http://www.last.fm/api
http://rcrastinate.blogspot.co.uk/2013/03/peace-through-music-country-clustering.html

Фото/Видео
https://www.flickr.com/services/api/

API Rotten Tomatoes

http://developer.rottentomatoes.com/docs