Версия от 12.10.2020

Установка и настройка

Работа скриптов проверена в Windows 7 и 10.

1. Установка и настройка Python

Установить последний стабильный релиз из списка https://www.python.org/downloads/windows/

**Увага! Не ставьте питон 3.9 – в нём не работает библиотека Pillow. Поддержка этой версии питона обещана после 15.10.2020. Версия 3.8.6 работает отлично.**

Работа скриптов проверена на релизах питона 3.7 и 3.8. Установить можно любой вариант x86 или x86-64: x86 будет работать на любом издании Windows, x86-64- только на 64-разрядном.

Download Windows x86-64 executable installer

Download Windows x86 executable installer

Почти все установки и параметры – по умолчанию: обязательно устанавливаем среду IDLE и обязательно ставим галочку «добавить в PATH».

Примечание 1: В среде «Visual Code» скрипты не работают.

Примечание 2: Если галочку «Path» не поставить, путь к python придётся указать вручную: свойства мой компьютер — дополнительные параметры — переменные среды — для пользователя – изменить — добавить в PATH папку расположения python

1. Установка Chrome webdriver

Для работы скриптов нужен установленный Google Chrome, а также Chrome webdriver соответствующей версии.

Скачать с <https://chromedriver.chromium.org/downloads> нужную версию драйвера, соответствующую установленной версии Chrome, извлечь содержимое архива (1 файл) в папку, доступную в PATH – например, в папку с установленным python. При обновлении Chrome (например, с версии 83.\* до 84.\* и т.д.) нужно обновить webdriver.

1. Установка библиотек python.

После выполнения предыдущих шагов нужно перезагрузиться (или завершить сеанс пользователя – это нужно для применения изменений в PATH).

Библиотеки устанавливаются через командную строку Windows:

python -m pip install --upgrade pip

pip install openpyxl

pip install Pillow – именно с большой буквы!

pip install selenium

Некоторые скрипты работают автономно – т.е. запускаются как обычная программа, некоторые – только через IDLE и управляются командами в консоли.

# Поиск кадастровых номеров по адресу (SIpON addr)

Можно запускать автономно.

Этот скрипт генерирует большое количество мусора.

На вход даётся книга KNaddr.xlsx с подготовленным списком адресов на первом листе. Результат помещается на второй лист. При повторном запуске список обрабатывается с начала, а для результата создаётся новый лист.

Адреса в списке разбиты на поля в соответствии с формой поиска на сайте, части адрес должны быть точно в том виде как это вводится на сайте.

Например, название области можно сократить до «волг», район – до «жирн» и т.д., но нельзя «волгоградская обл».

Поскольку адреса в росреестре представлены в очень разнообразном виде, то одна и та же улица может быть написана по-разному, кроме того, возможны опечатки в названиях улиц. В этом случае следует искать все возможные варианты. Например, «улица имени генерала Карбышева» следует искать по строкам «гене» и «карб». Хотя в этом случае будут найдены также и другие улицы, где встречается слово «генерала».

При поиске домов с литерами следует учесть, что дом «5а» может быть внесён в росреестр множеством разных способов, так что такие варианты почти всегда приходится искать вручную, либо использовать строку «5\*», что сильно увеличивает количество мусора в результатах. Так же вручную приходится искать номера квартир вида «8-9», «12,13», «7б» и т.п.

В качестве результата берётся список КН, выданных поиском на первой странице, т.е. максимум 10 штук. Никакой проверки на данном этапе не производится. Часто из-за каких-то особенностей сайта в поиск попадают объекты из других районов и населённых пунктов. На лист с результатами записывается пара «номер ЛС»+»КН». Если «КН» пусто, значит, в росреестре этого адреса нет. Такие ЛС нужно проверять вручную с учётом возможных вариантов написания адреса, переименования улиц, угловых домов и т.д.

Скрипт нормально обрабатывает почти все ошибка сайта, сам перезапускается, делает до 10 попыток поиска каждого адреса в случае ошибок сайта. Запускать можно как автономно, так и через IDLE. Можно запускать несколько копий (до 6) скрипта из разных папок.

Список найденных КН передаётся следующему скрипту.

# Поиск сведений об объекте по кадастровому номеру (SIpON)

Можно запускать автономно.

Основной скрипт, собирающий полезную информацию.

На вход даётся книга KN.xlsx с подготовленным списком кадастровых номеров на первом листе. Результат помещается на второй лист. При повторном запуске список обрабатывается с начала, а для результата создаётся новый лист.

В результат сохраняется информация, необходимая и достаточная для дальнейшей работы: тип объекта, актуальность объекта, адрес, дата перехода прав.

## Проверка полученной информации.

Категория объекта

Объект капитального строительства, Земельный участок

Учтённый

Ранее учтенный, Учтенный, Временный, Снят с учёта, Ликвидирован, Аннулирован и т.п.

Адрес, Тип объекта

С учётом огромного количества ошибок в адресах, внесённых в базу росреестра, необходимо сверить 100% адресов, полученных скриптом, с исходными адресами, запрошенными по списку лицевых счетов. Возможные ошибки кратко описаны <https://habr.com/ru/post/505752/>. Данный этап не автоматизирован и выполняется вручную.

Форма собств.

Частная, Муниципальная, Не указана и т.п.

Дата последнего перехода права собств.

В случае, если дата указана, можно делать запрос в ФГИС ЕГРН для получения выписки о переходах прав. Если вместо даты есть строка «Данные отсутствуют. Найти объект на публичной кадастровой карте», то это означает, что сведений нет и делать запрос не имеет смысла. Строка «Право» означает, что запрос в ФГИС ЕГРН, скорее всего, не имеет смысла, т.к. там нет информации – поиск в базе ФГИС ЕГРН выдаст «объект отсутствует»

Если в каких-то полях есть строки «Данные не актуализированы», это означает, что информация по объект недвижимости и переходам прав внесена в базу росреестра не полностью. В этом случае делать запрос в ФГИС ЕГРН бесполезно.

Я использую такой порядок обработки списка:

1. Удалений всех строк, в которых
   1. Категория объекта = земельный участок
   2. Адреса из других районов и населённых пунктов
   3. Отсутствует дата перехода права
2. Сверка адресов
3. Сохранение информации в АИС РНГ

В случае выявления ошибок в адресах следует делать запрос в техподдержку росреестра для уточнения адресов, внесения актуальной информации и т.д.

Скрипт нормально обрабатывает почти все ошибка сайта, сам перезапускается, делает до 10 попыток поиска каждого объекта в случае ошибок сайта. Запускать можно как автономно, так и через IDLE. Можно запускать несколько копий (до 6) скрипта из разных папок.

# Сохранение информации в АИС РНГ

Обработка «ЗаписатьИнформациюИзРосреестра.epf» записывает всю собранную информацию из файла «Шаблон.xlsx»: кадастровый номер в соответствующий реквизит справочника «Абоненты», даты перехода - в примечание ЛС и общую информацию - в Журнал Посещений. В последней колонке шаблона может стоят дата выполнения запроса. Если колонка пустая, то используется текущая дата. В дальнейшем можно, например, сверить дату перехода права, полученную из росреестра, с датой переоформления ЛСМ в АИС РНГ

Обработка «ЗаписатьКадастровыйНомер.epf» просто записывает кадастровый номер из файла «ШаблонКН.xlsx»

Для выборки сохранённых КН можно использовать отчёт «ВыборкаКадастровыхНомеров.erf». В отчёте нужно исправить следующий момент: при ручном редактировании кадастрового номера в карточке абонента в реквизит «КадастровыйНомер» справочника «Абоненты» записывается пробел – отчёт его не убирает, это может привесит к ненужным ошибкам при обработке полученного списка.

При записи информации обработками этот момент учитывается и обрабатывается корректно.

# Запрос информации о переходах прав в ФГИС ЕГРН (EGRN bot v.2.1)

Бот для выполнения запросов в (полу)автоматическом режиме.

Бот запускается только через IDLE.

После запуска открывается окно Chrome. Необходимо авторизовался своим ключом, перейти в раздел «Поиск объектов недвижимости». Можно записать ключ доступа в текстовый файл auth.txt – в этом случае бот сам дойдёт до страницы поиска. Бот делает до 10 попыток подключится к сайту, при этом окна Chrome и webdriver могут несколько раз закрыться – открыться.

Ввести список КН (максимум 200), выбрать регион (обязательное требование сайта), нажать кнопку «Поиск».

После появления списка найденных объектов недвижимости нужно вернуться в консоль «Python Shell» и выполнить команду GetAll() – запустится процедура создания запросов с отработкой таймаута по умолчания (300 секунд). Таймаут можно изменить GetAll(280) – 280 секунд. К этому значению добавляются накладные расходы – примерно 16-17 секунд. Т.е. по умолчания таймаут составляет 316 секунд. Минимальный реальный таймаут на сайте составляет 315 секунд. В случае, если запрос не укладывается в таймаут, бот каждые 10 секунд будет автоматически повторять запрос до успешного его завершения.

Скрипт обрабатывает много ошибок, которые могут возникнуть при выполнении запросов, кроме появления красного баннера «Communication Error».

Контролировать выполнение скрипта нужно: в консоли по времени запросов, по появлению ошибок и сообщений в окне с сайтом и т.д., в другом браузере по созданным запросам и т.д. При нормальной работе сайта обрабатываются запросы по всем КН, и дополнительное вмешательство в работу не требуется. Также возможна отправка email после каждого запроса (с верии 2.1).

Все созданные запросы записываются в файл «rq.xlsx» в формате: кадастровый номер, номер запроса, дата запроса. Файл сохраняется после каждого запроса, так что потери сделанных запросов быть не должно. Если при запуске скрипта фал отсутствует, он создаётся автоматически. Если при запуске скрипта или после команды GetAll() файл открыт в Excel и недоступен для записи, выдаётся соответствующее предупреждение и работа скрипта останавливается. В дальнейшем доступность файла не проверяется.

В случае возникновения ошибок возможны варианты:

- закрыть все окна, кроме собственно IDLE со скриптом (консоль Shell, консоль webdriver, окно Chrome), и перезапустить процесс с новым списком КН;

- убедиться, что выполнение скрипта в Shell остановлено, в окне Chrome устранить ошибку – авторизоваться и т.п., ввести новый список КН, выполнить команду GetAll().

В консоль Shell выводится вся информация о выполнении скрипта, там же появляется сигнал о завершении работы.

**Новое в версии 2.1**

Для контроля работы бота и оповещения об аварийной остановке используется электронная почта. Письма отправляются:

* при аварийной остановке в ситуации, когда бот не может исправить ошибку (красный баннер на сайте);
* по окончании работы;
* опционально: после каждого успешного запроса.

За логику отправки писем отвечает библиотека send\_email.py. Настройка оповещений выполняется через файл email.ini в папке бота:

[smtp]

server = smtp.yandex.ru

addr = my\_box\_name@yandex.ru

apppass = my\_password

sendOK = yes

Назначение параметров:

* server – имя SMTP сервера;
* addr – адрес почты, он же используется в качестве логина при подключении к серверу, на него же бот отправляет оповещения;

apppass – пароль;

sendOK – если значение этого параметра равно yes, то отправляются все сообщения, в том числе после каждого запроса. Если значение иное или параметр вообще отсутствует или закомментирован, то отправляются только сообщения о завершении работы и об аварийной остановке.

Текст сообщений прописан в коде основного скрипта:

* после каждого запроса: тема = «EGRN bot OK», текст содержит кадастровый номер и номер запроса. Отправка этих писем контролируется по наличию в теме строки «OK»;
* после окончания работы: тема = «EGRN bot all done», текст = «All done!»;
* после аварийной остановки: тема = «EGRN bot FATAL ERROR», текст = «Fatal error!».

Рекомендации по настройке почты.

Проверена работа с Яндекс.Почтой, дальнейшие рекомендации сделаны исходя их этого.

1. Создать для бота отдельный ящик. Желательно, но не обязательно, в целях безопасности привязать к ящику номер телефона, резервный email, включить в настройках Яндекс.Паспорт двухфакторную аутентификацию (2ФА).
2. В качестве пароля рекомендуется использовать пароль приложения, а не основной пароль. Пароль приложения создаётся в настройках параметров безопасности ящика.
3. Создать правило обработки почты: поскольку бот отправляет письма сам себе, то нужно настроить пересылку писем на любой другой (основной, рабочий и т.п.) ящик. Отправка писем на свой же адрес продиктована тем, что другие почтовые системы могут отказываться принимать письма от бота. При пересылке такой проблемы нет.

# Скачивание отработанных запросов (EGRN downloader)

Скрипт для скачивания отработанных запросов.

Бот запускается только через IDLE.

После запуска открывается окно Chrome. Необходимо авторизовался своим ключом, перейти в раздел «Мои запросы». Можно записать ключ доступа в текстовый файл auth.txt – в этом случае бот сам дойдёт до списка запросов. Бот делает до 10 попыток подключится к сайту, при этом окна Chrome и webdriver могут несколько раз закрыться – открыться.

После запуска можно запустить загрузку результатов запросов:

* dnd("12.03.2020") - все запросы за эту дату
* dnd("08.03.2020-17.03.2020") - все запросы в диапазоне дат
* dnd("17.03.2020-08.03.2020") - порядок дат не имеет значения
* dnd("12.03.2020-15.04.2020", 5, 12) - только те, что находятся на страницах 5..12 (по умолчанию - 1..200)
* dnd("12.03.2020-15.04.2020", 5) - только со страниц 5..200
* dnd("12.03.2020-15.04.2020", , 12) - только со страниц 1..12

После отработки скрипта его можно повторно запустить той же командой. В некоторых случаях в консоль выводится ошибка, нужно просто повторить команду dnd.

Процесс загрузке сохраняется в файл «dn.xlsx». Файлы повторно не загружаются. Т.е. если результат был скачан и запись об этом событии есть в файле, то при повторном запуске этот результат не будет загружен повторно. В конце работы выводится статистика по запросам за указанные даты: сколько запросов в каком статусе находится, сколько результатов загружено.

После завершения загрузки нужно проконтролировать:

- соответствие количества фактически скачанных результатов и того, что написано в консоли (Завершена и загружена): иногда сайт не скачивает файл при нажатии на ссылку. При этом запись о загрузке сохраняется в файл «dn.xlsx». Решение – скачать пропущенный результат вручную.

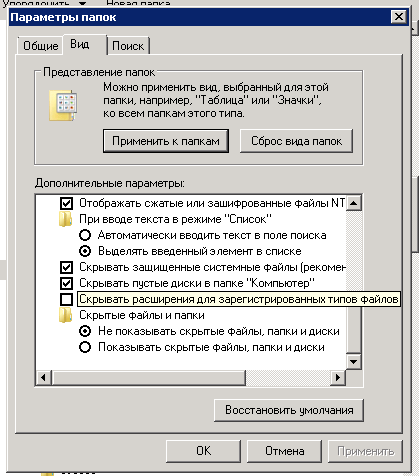
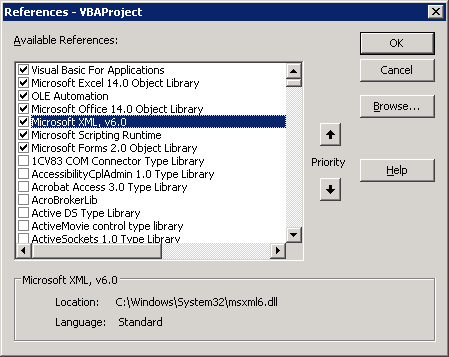
- соответствие количества сделанных запросов за эти даты и того, что написано в консоли (Всего): иногда сайт не успевает сформировать все 25 строк с запросами, и скрипт пропускает некоторые запросы. Такое случается достаточно редко, но тем не менее… Самое простое – отслеживать в консоли количество строк на страницу (Обработана страница 1 :: строк 25). Можно повторно скачать результаты только с этой страницы:

dnd("01.06.2020-30.06.2020", 7, 7) – страница 7

# Обработка результатов запросов (книга Excel)

Набор макросов для обработки.

Для работы необходимо:

1. разрешить макросы в Excel
2. включить отображение расширений для файлов (необходимо для работы встроенного архиватора)  
   
3. подключить в Excel дополнительные компоненты для работы с XML (Alt+F11 > Tools > References, найти в списке нечто с буквами XML, чекнуть);  
   

Сложить все файлы «Response-80-.zip», полученные от Росррестра, в одну папку; в процессе работы из этой папки будут удалены все файлы, кроме «Response-80-.zip»; выписки будут сохранены там же; обработаны будут все файлы.

Порядок работы:

1. Начинаем работу с листа «#Список адресов для запроса»:
   * ЛС и Адрес – исходные данные, которые нам поступают от заказчика; полученная выписка будет называться так, как написано в столбце «Адрес» (но это не точно, см. ниже);
   * Кадастровый номер – тот, что мы нашли в открытой части Росреестра – первый этап процесса;
   * Остальные столбцы в работе не участвуют.
2. Лист «#Главный»: нажимаем кнопку «Обработать все файлы». Со скоростью от 5 до 50 штук в минуту (в зависимости от скорости доступа к хранилищу файлов) получаем выписки в формате PDF.
3. На двух последних листах формируются отчёты.

# Дополнительно: выборочное копирование результатов

Если нужно обработать не все запросы, можно сохранить их список в текстовый файл и скопировать в другую папку:

В текстовый файл «1.txt» сохранить список номеров запросов в том виде, как они сохраняются в файлах rq.xlsx и dn.xlsx.

Сохранить этот файл и скрипт 1.cmd в папке с результатами. После выполнения скрипта будет создана папка 1 и в неё будут скопированы все файлы из списка.

if not exist 1\ (

mkdir 1

) else (

echo y | del 1\\*.\*

)

for /f %%f in (1.txt) do (

echo %%f

copy Response-%%f%.zip 1\\*.\*

)

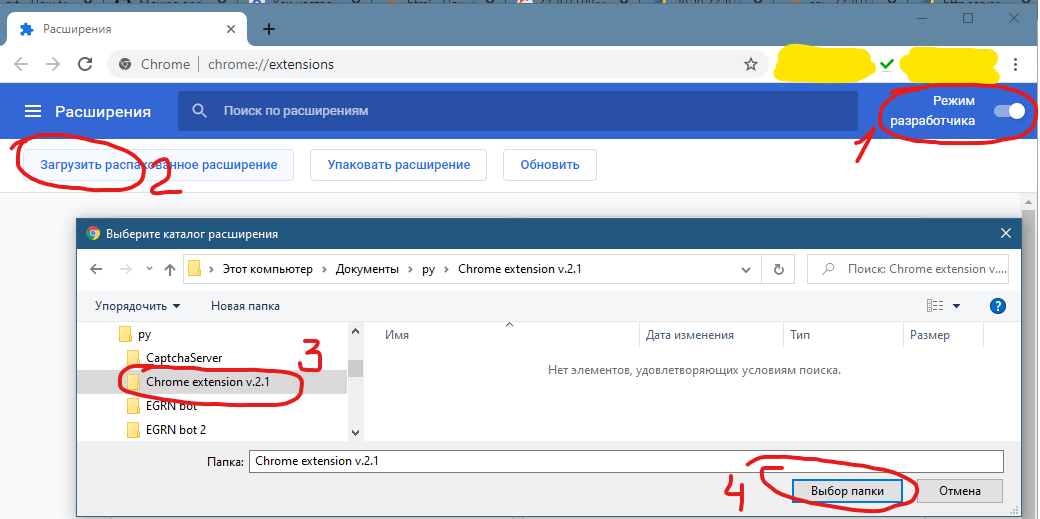
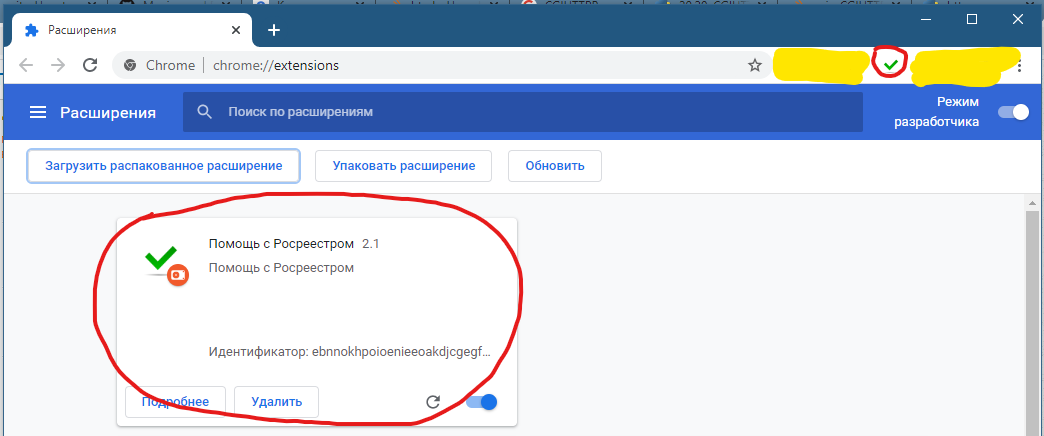
pause

# Дополнительно: расширение для Chrome для работы с сайтом

Данное расширение работает только на страницах открытой части росреестра:

1. распознаёт капчу (при помощи дополнительного web-сервера) – достаточно заполнить поля формы и нажать Enter;
2. на страницах с объектами раскрывает раздел «Права и ограничения», что позволяет сразу оценить полезность найденного объекта, и скрывает лишние элементы сайта.

Использование:

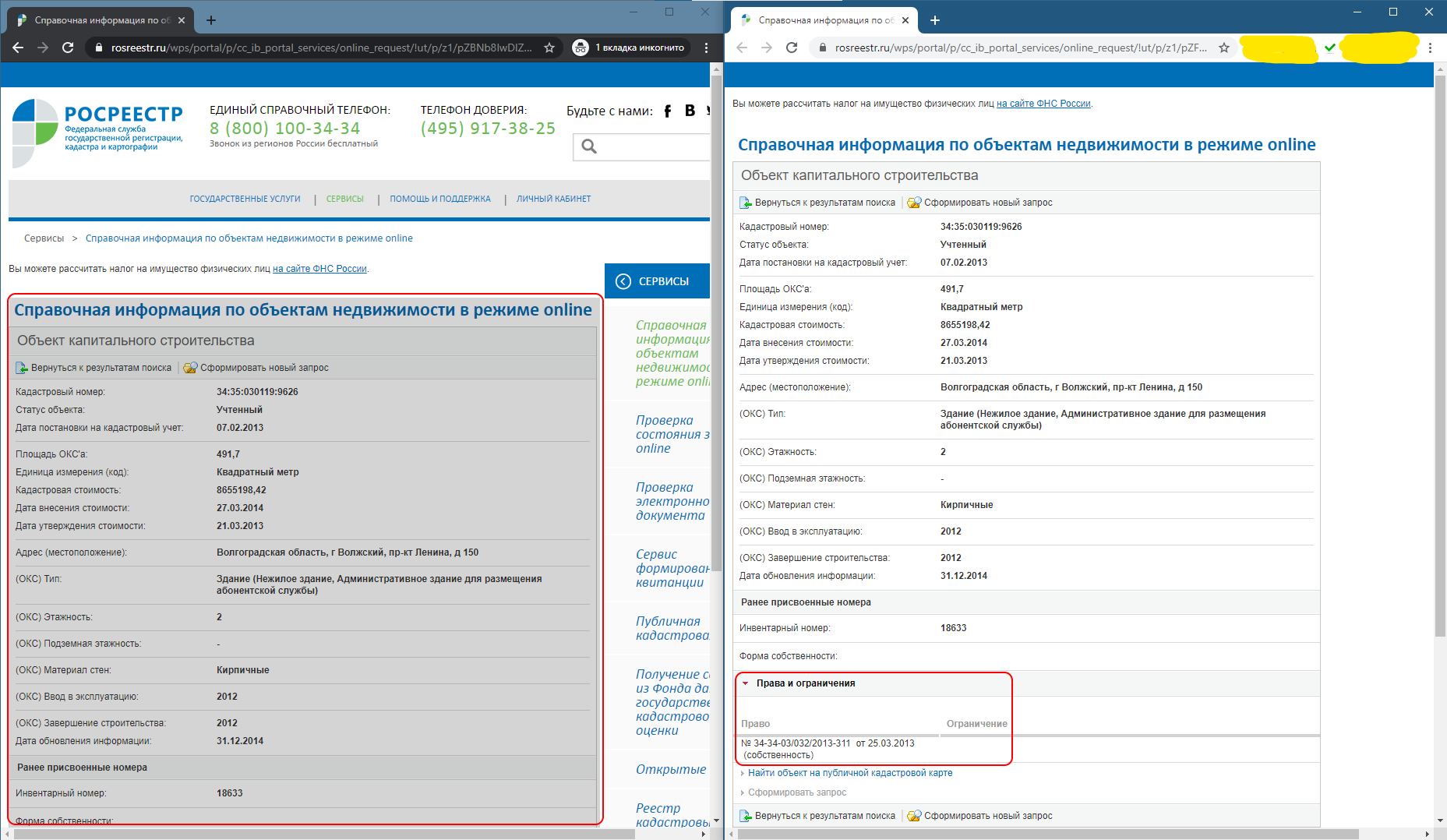
1. Установить расширение (делается один раз). Для этого:
   1. сохранить папку с расширением в любое удобное место;
   2. в Chrome включить режим разработчика и загрузить распакованное расширение  
        
        
      правильно установленное расширение  
        
        
      При каждом запуске Chromeбудет выдаваться предупреждение о включенном режиме разработчика и установленном расширении – с этим придётся смириться. Либо отключить расширение и режим разработчика и включать при необходимости.
2. Запустить сервер распознавания капчи. При запуске открывается консольное окно –если его закрыть, сервер не работает.

Результат работы расширения:

|  |  |
| --- | --- |
|  | 1. после автоподстановки распознанной капчи можно в любой момент нажать Enter, целиться в кнопку "Сформировать запрос" не обязательно. 2. Капча распознаётся и подставляется автоматически в течение не более одной секунды после появления или изменения картинки. |

|  |  |
| --- | --- |
|  | 1. после автоподстановки распознанной капчи можно в любой момент нажать Enter, целиться в кнопку "Сформировать запрос" не обязательно. 2. Капча распознаётся и подставляется автоматически в течение не более одной секунды после появления или изменения картинки. |

Сравнение страниц при отключенном и включенном расширении: отсутствуют ненужные элементы интерфейса сайта, раздел «Права и ограничения» развёрнут.



Настройка.

Сервер распознавания капчи – простой http-сервер, который получает от расширения картинку и отдаёт распознанную капчу. По умолчанию сервер отвечает на запросы на порту 8111.

Если нужно изменить порт и адрес:

1. В коде сервера исправить номер порта
2. В коде расширения в файле «contentScript.js» исправить адрес сервера и номер порта.

# Copyright notice

Весь код свободен от любых ограничений, распространяется “as is” без всяких обязательств и ограничений. Код доступен по адресу <https://github.com/0-6-1-7/rosreestr>