PropertyToolkit

# Development Outline

## Languages / plugins:

* Python
* Scrappy
* JSON
* Mysql

## Standards

* Include comments throughout code, to allow for scraper to be updated by others in future.
* Have each scraper select and use a proxy (if available in mysql database) for each scraping session.

## Sites

* Zoopla : <https://www.zoopla.co.uk/>
* Rightmove : <https://www.rightmove.co.uk/>
* Land Registry : <https://landregistry.data.gov.uk/data/ppi/transaction-record.html>
* EPC Open Data : <https://epc.opendatacommunities.org/>
* findAddress : <https://findaddress.io/>

# Goals

* Accurately collect data from Rightmove and Zoopla very quickly with minimal server load , keeping connections

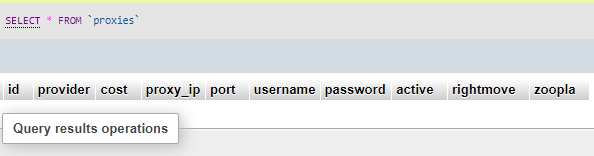
## Objectives

* Write a scraper to collect all property listings from Zoopla triggered by a cron job, adding new listings to a crawl list whilst archiving those missing, sold, let agreed, etc.   
  To be based a search saved in mysql, rotate through saved search in a day to scrape each query.  
  (a slight variation in URL will collect rent or sale properties)
* Write a scraper to collect individual Zoopla listings (to let or for rent, as source code is almost identical)
* Write a scraper to find to let properties full addresses through checking Zoopla Sold data listings
* Write a scraper to collect all property listings from Rightmove triggered by a cron job, adding new listings to a crawl list whilst archiving those missing, sold, let agreed, etc.   
  To be based a search saved in mysql, rotate through saved search in a day to scrape each query.  
  (a slight variation in URL will collect rent or sale properties)
* Write a scraper to collect individual Rightmove listings (to let or for rent, as source code is almost identical)
* Using sold data (price and year) and postcode collected from both listings to obtain full property addresses from Land Registry Linked Data API (using GET HTTP and JSON)
* Insert collated and updated data in to mysql database

# Scraper Breakdown

### Proxy Selection

If a proxy is available, then randomly rotate through proxies available in mysql database, if not then simply scrape using the servers IP address (so without).



Select an active proxy, grab the proxy\_ip, port, username and password – then use in crawler.

If either rightmove or zoopla temporarily suspend access, the sites will through 500 errors, after 3 hours they lift the block and let crawlers continue. This should be noted in the rightmove / zoopla column, so if a proxy is suspended, then another proxy is used until time has passed.

## Property Listing Lists

On each CRON job hit rotate through each search criteria in mysql database to collect property listings, to identify new listings, those reduced, those sold / let and those removed.

Each search criteria provides the following:

* Search **radius**, in miles.
* Minimum **bedrooms**
* Maximum **bedrooms**
* **Location** / Area identifier (latter for rightmove)
* Sale Count (ie number of units for sale in area) – these are updated on each crawl
* Rent Count (ie number of units to rent in area) – these are updated on each crawl
* Site (ie rightmove or zoopla)

### Rightmove

1. Scraper to open connection, and leave open until crawl complete.
2. Depending on crawling for sale or to rent, head to:
   1. For sale:  
      [https://www.rightmove.co.uk/property-for-sale/find.html?locationIdentifier=#LOCATION-IDENTIFIER#&radius=#RADIUS#&maxBedrooms=#MAX-BEDROOMS#&minBedrooms=#MIN-BEDROOMS#&propertyTypes](https://www.rightmove.co.uk/property-for-sale/find.html?locationIdentifier=#LOCATION-IDENTIFIER)=
   2. To rent:  
      [https://www.rightmove.co.uk/property-to-rent/find.html?searchType=RENT&locationIdentifier=#LOCATION-IDENTIFIER#&insId=1&radius=#RADIUS#&minBedrooms=#MIN-BEDROOMS#&maxBedrooms=#MAX-BEDROOMS#](https://www.rightmove.co.uk/property-to-rent/find.html?searchType=RENT&locationIdentifier=#LOCATION-IDENTIFIER)

Simply collect the number of available results



To mysql update the search results for sale and rent count.

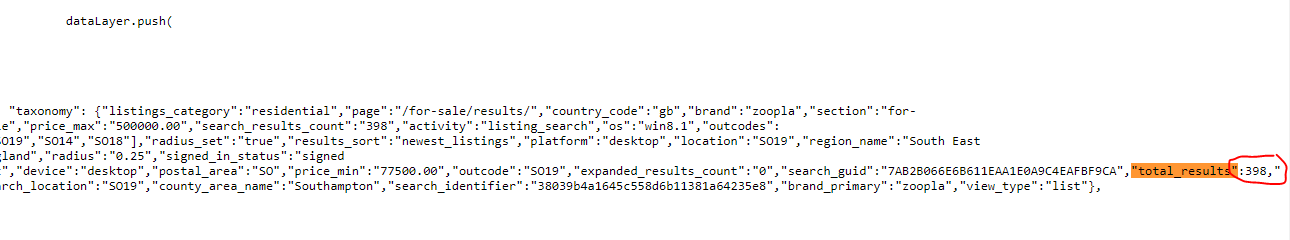
1. Now repeat the search including the sold and let agreed properties, paginating and to index the listings.
   1. For sale:  
      Add &includeSSTC=true  
      to the URL
   2. To Rent:
   3. Add &includeLetAgreed=true  
      to the URL

Pagination can be calculated by taking the new number of results, including those agreed, dividing by 24 and rounding up using the get operator: &index=#PAGINATION#, ie &index=0 returns the first page, &index=24 the second page, &index=48 the third page, etc.

1. Rightmove kindly returns the results as JSON at the end of each page, making it very easy and accurate to collect the results.   
   When looking at the source code search “<script>window.jsonModel =” to find the JSON followed by “</script><script>” where the JSON closes off. Ie:  
   
2. The Json returns a section called “properties” which is the set of listings to add to the crawl list “tbl\_property\_rightmove”.   
   From this we wish to collect, insert or update, as individual property listing rows:
   1. Property\_id – in JSON listed as ID.
   2. date\_cur – The date when the property was found and added to the database by the crawler
   3. property\_link – in JSON listed as propertyURL
   4. prop\_type – sale or rent (based on the search)
   5. site – rightmove, depending where found
   6. criteria\_id – based on the mysql search
   7. archived – set to 0 if displayStatus is blank, set to 1 for any other reason (ie Let Agreed, Sale Agreed or missing)
   8. archived\_date – date set to date crawled when archived set to 1 for reasons mentioned
   9. crawled – set to 0, updated to 1 when individual listings crawler retrieves data
   10. last\_crawl – date previous crawl happened
   11. last\_found – updated each time the Rightmove Listings crawler finds the property
3. If the property last\_found exceeds 3 days, then it should be archived and archived date set.
4. When a property is found and it already exists is the ‘property\_details\_rightmove’ / details list the [price][amount] is checked against the stored property list for the reduction to be inserted in to the database  
   - This could be achieved through querying an array built from one initial mysql lookup (using property ID and price).  
   If the price has changed it should checked whether the reduction has already been inserted into “property\_duplicate\_rightmove” (a list of reductions) if not insert.

### Zoopla

1. Scraper to open connection, and leave open until crawl complete.
2. Depending on crawling for sale or to rent, head to:  
   **note:** the first #location-identifier# in the URL must be lower case, the second in the URL uppercase.
   1. For sale:  
      https://www.zoopla.co.uk/for-sale/property/#LOCATION-IDENTIFIER#/?beds\_max=#BEDROOMS-MAX#&beds\_min=#BEDROOMS-MIN# &q=#LOCATION-IDENTIFIER#&radius=#RADIUS#&results\_sort=newest\_listings&search\_source=refine&page\_size=100
   2. To rent:  
      https://www.zoopla.co.uk/to-rent/property/#LOCATION-IDENTIFIER#/?beds\_max=#BEDROOMS-MAX#&beds\_min=#BEDROOMS-MIN#&price\_frequency=per\_month&q=#LOCATION-IDENTIFIER#&radius=#RADIUS#&results\_sort=newest\_listings&search\_source=refine&page\_size=100

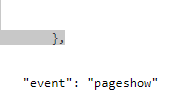
Simply collect the number of available results, available in JSON:  


To mysql update the search results for sale and rent count.

1. Now repeat the search including the sold and let agreed properties, paginating and to index the listings.
   1. For sale:  
      Add &include\_sold=true  
      to the URL
   2. To Rent:
   3. Add &include\_rented=true  
      to the URL

Pagination can be calculated by taking the new number of results, including those agreed, dividing by 100 and rounding up using the get operator: &pn=#PAGINATION#, ie &pn=1 returns the first page, &pn=2 the second, &pn=3 the third, etc.

1. Zoopla returns part of the results as JSON at the start of each page, making it easy to obtain initial listings though not current prices or property status.   
   When looking at the source code search "ecommerce": { to find the JSON followed by }, where the JSON closes off. Ie:



1. The Json returns a section called “ecommerce” which is the set of listings to add to the crawl list “tbl\_property\_rightmove”.   
   From this we wish to collect, insert or update, as individual property listing rows:
   1. property\_id – in JSON listed as id.
   2. date\_cur – The date when the property was found and added to the database by the crawler
   3. property\_link – simply “<https://www.zoopla.co.uk/to-rent/details/>#property\_id#, ie https://www.zoopla.co.uk/to-rent/details/12345678
   4. prop\_type – sale or rent (based on the search)
   5. site – zoopla, depending where found
   6. criteria\_id – based on the mysql search
   7. Archived data will require scraping through the results
   8. Archived updates will require scraping through the results
   9. crawled – set to 0, updated to 1 when individual listings crawler retrieves data
   10. last\_crawl – date previous crawl happened
   11. last\_found – updated each time the Zoopla Listings crawler finds the property
2. Zoopla uses a class called “status-ribbon” which is marked against the listings / rows when a property is no long available and should be marked as archived. At this point ‘archived’ should be set to 0 if “status-ribbon” does not exist for that listing and set to 1 for any other reason (ie Let Agreed, Sale Agreed or missing)
3. archived\_date – date set to date crawled when archived set to 1 for reasons mentioned
4. If the property last\_found exceeds 3 days, then it should be archived and archived date set.
5. When a property is found and it already exists is the ‘property\_details\_rightmove’ / details list the property’s price (pcm or sale price) should be checked against the stored property list for the reduction to be inserted in to the database.  
   - This could be achieved through querying an array built from one initial mysql lookup (using property ID and price).  
   If the price has changed it should checked whether the reduction has already been inserted into “property\_duplicate\_rightmove” (a list of reductions) if not insert.

# Individual Property listings

On each CRON job hit work through to crawl all those new properties added to the database by the property list crawlers.

These should be selected based on:

* Archived – 0
* Archived\_date – 0000-00-00
* Crawled – 0
* Last\_crawl – 0000-00-00
* Last\_found – 0000-00-00
* Query based on prop\_type and site to match appropriate crawler

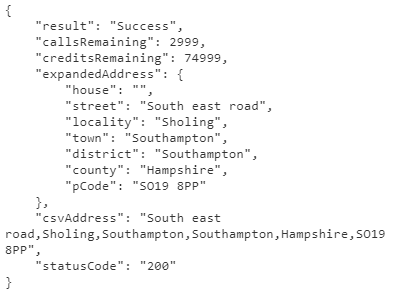
## Rightmove

Rightmove for sale and to rent listings follow the same style, ie: <https://www.rightmove.co.uk/properties/83384590/>

1. Query the collected property list table “tbl\_property\_rightmove” to select all that are:
   1. Site: rightmove
   2. Archived: 0
   3. Crawled: 0
2. Connect to Rightmove, store a cookie, keep the session open, then scrape each uncrawled property listing.
3. Head to the address from the query in “property\_link”, ie:
   1. <https://www.rightmove.co.uk/properties/85711315#/> - note always keep the #/ at the end of a URL
   2. Check whether the header redirects to the new layout to choose which method to scrape
4. Scrape JSON from:  
     
   To:  
   To obtain:
   * ~~Text, description - description~~
   * ~~Images, url, item 1 – main image~~
   * ~~analyticsInfo, analyticsBranch, brandName – agent name~~
   * ~~analyticsInfo, analyticsBranch, displayAddress – agent name~~
   * ~~contactInfo, telephoneNumbers, localNumber – agent number~~
   * ~~analyticsProperty, added – rearrange to add dashes between days and month, ie 20200811 would be 2020-08-11~~
   * ~~lettings, letAvailableDate – lettings available~~
   * ~~analyticsProperty, beds – bedrooms~~
   * ~~analyticsProperty, postcode – postcode~~
   * ~~analyticsProperty, price – property price~~
   * ~~analyticsProperty, propertySubType – property type~~
   * ~~analyticsProperty, propertyId – property id~~
   * ~~analyticsProperty, minSizeAc – divide by 10.76 to add as EPC\_m2~~
   * ~~brochures, url – check through all URLs to find which is a PDF, keep the PDF URL to insert~~
   * ~~analyticsProperty, latitude – latitude~~
   * ~~analyticsProperty, longitude – longitude~~
   * ~~epcGraphs, url – epc link, detect whether it’s an image or url (as will need to be inserted into the db differently)~~
   * ~~floorplans, url – floor plan link~~
   * ~~transactionType – RENT or BUY~~
   * ~~analyticsProperty, letAgreed – if not false, archive property~~
   * ~~analyticsProperty, soldStc – if not false, archive property~~
   * address, deliveryPointId – as this ID allows the sold data to be accesses.
5. Using:  
   https://www.rightmove.co.uk/properties/api/soldProperty/transactionHistory/#deliveryPointId#  
   Scrape the JSON, ie <https://www.rightmove.co.uk/properties/api/soldProperty/transactionHistory/71449219>  
   Returns:  
   The data to collect is the first result.  
   So:

* soldPropertyTransactionHistories, item 1, year – sold year
* soldPropertyTransactionHistories, item 1, soldPrice – sold price

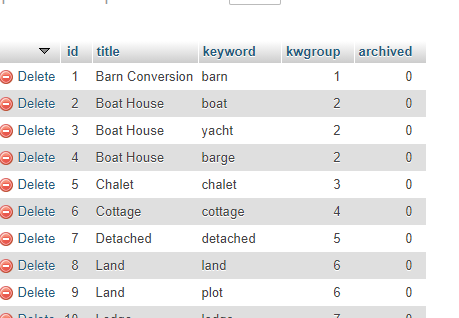
1. Using the https://findAddress.io/API/ JSON API and the postcode collected to find the road name, town and county. Get URLs return the JSON address details using a URL like:   
   https://findAddress.io/API/NULL/##POSTCODE##  
   This returns:

  
The street, locality, town and county are shall be inputted in to the database.

### Rightmove Data Insertion

1. INSERT INTO ` property\_details\_rightmove` :
   * `property\_id` = propertyID
   * `title` = null
   * `address` = Road Name Town County Postcode (from findAddress)  
     ie Obelisk Road Southampton Hampshire SO19 8BL  
     Salhouse Road Little Plumstead Norwich NR6 7BW
   * `address\_count` = null
   * `price` = as collected above
   * `agent` = agent / branch / brand name
   * `agent\_address = agent / branch / brand address
   * `agent\_no` = agent / branch / brand number
   * `agent\_match` = agent name processed
     1. Capitalise text
     2. Strip and remove any html tags
     3. Remove any text after and including a dash -, ie TRISTANS AGENCY – PORTSMOUTH, would become TRISTANS AGENCY
     4. Remove any text after and including a open bracket (, ie TRISTAN ESTATES (PORTSMOUTH), would become TRISTAN ESTATES
     5. Remove any text after and including a comma, ie TRISTAN LETS PROPERTY, Portsmouth would become TRISTAN LETS PROPERTY
     6. Remove any &AMP;
     7. Remove any &
     8. Replace the word company with CO, ie Tristan Lets Company, becomes TRISTAN LETS CO
     9. Remove any of the following words:
        + LTD
        + LIMITED
        + AGENTS
        + AGENCY
        + AGENT
        + ESTATE
        + LETTING
        + LETTINGS
        + SALES
        + RESIDENTIAL
        + INDEPENDENT
        + COMMERCIAL  
          AND
        + MANAGEMENT
        + INVESTMENT
        + CORPORATE
        + .COM
        + FRANCHISE
     10. Remove all white spaces, so TRISTAN LETCO becomes TRISTANLETSCO
     11. If the final letter is a S remove it, ie BELLWAYHOMES becomes BELLWAYHOME
     12. Insert the processed agent name in the database
   * `bedrooms` – beds
   * `property\_link` - photo / image of property
   * `description` - To insert this in to the database later the apostrophe’s and quote marks need to be removed, ie Tristan’s house is in great condition with a kitchen that’s 14”6’ – would be Tristans house is in great condition with a kitchen thats 146
   * `cur\_date` - the date the scrape is being completed, ie CURDATE()
   * `listed` - added / addedon
   * `prop\_address` - same as `address`
   * `url`
     1. if for sale:   
        [https://www.rightmove.co.uk/property-for-sale/property-#propertyId#.html](https://www.rightmove.co.uk/property-for-sale/property-#propertyId)
     2. if for rent:  
        [https://www.rightmove.co.uk/property-to-rent/property-#propertyId#.html](https://www.rightmove.co.uk/property-to-rent/property-21488234.html)
   * `road` - if available paon, soan from land registry, else street / road from findAddress.io  
     Ie: Flat 1, 12 Hook Street  
     Apartment 20, Windsor Court, Bay Road

12 Thorpe Close

* + `town` - locality, district – ie Woolston, Southampton or Thorpe, Norwich
  + `postcode` - enter the postcode
  + `archived` - 0, 1 if letAgreed or stcAgreed
  + `archived\_dated` - 0000-00-00 or CURDATE() if letAgreed or stcAgreed
  + `criteria\_id` - id of the crawl / search to find those results
  + `brocher\_link` - brochures url / pdf link
  + `floorplan\_link` - floorplan image url
  + `epc\_link` - epc url if a PDF or a link
  + `epc\_image` - epc image url, if a jpeg / jpg, gif or png
  + `EPC\_EER\_Current` - 0
  + `EPC\_EER\_Potential` - 0
  + `EPC\_EIR\_Current` - 0
  + `EPC\_EIR\_Potential` - 0
  + `EPC\_m2` - 0
  + `last\_sold\_price` - last sold price
  + `last\_sold\_date` - last sold date
  + `new\_full\_address` - only entered if address was obtained from sold data, if so paon, soan, road, locaility, town, county postcode
  + `property\_type` - propertySubType looked up against keyword list in database called `property\_type` returning the kwgroup number to insert in to the database.   
    Ie  
      
    Barn Conversion returns 1  
    Boat returns 2

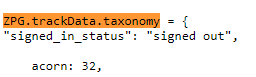
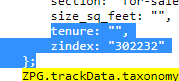
etc

# Zoopla

For sale and to rent listings are very similar, there is only one type of crawling required.

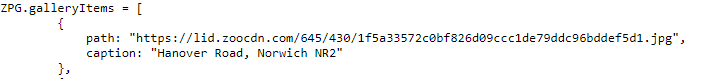
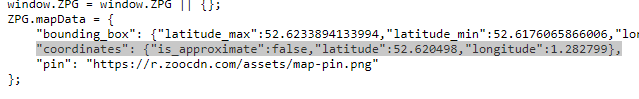
A second Zoopla crawler finds to rent full addresses.

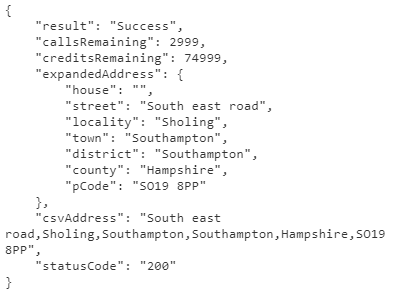
### Sale & Rent

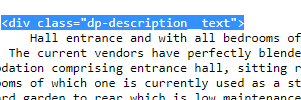
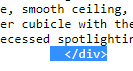
1. Scrape JSON for property details from:  
     
   to:  
   

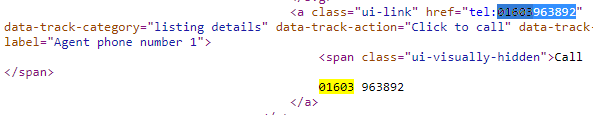
To collect:

* + Brand\_name – agent
  + Outcode + incode – postcode
  + Listing\_condition – pre-owned / new
  + Num\_beds – bedrooms
  + Price – price
  + Property\_type – property type
  + Section – for-sale / to-rent
  + Size\_sq\_ft – divide by 10.76 to insert as EPC\_m2
  + Zindex – propertyid
  + listing\_status - rent\_under\_offer / to\_rent / sold / for\_sale

1. Scrape JSON for first photo:  
   
2. Scrape JSON for lat long, use “coordinates, latitude” and “coordinates, longitude”:  
   
3. Using the https://findAddress.io/API/ JSON API and the postcode collected to find the road name, town and county. Get URLs return the JSON address details using a URL like:   
   https://findAddress.io/API/NULL/##POSTCODE##  
   This returns:

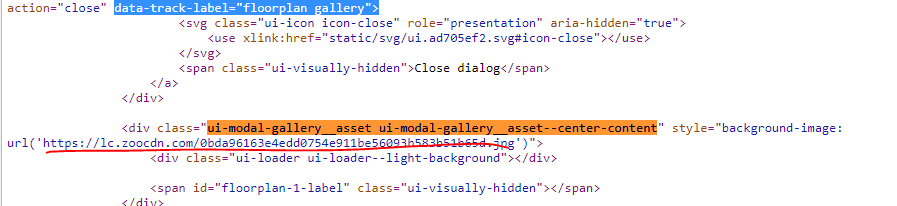
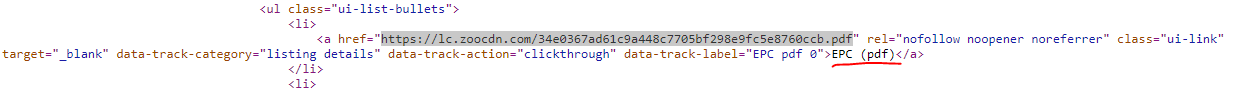
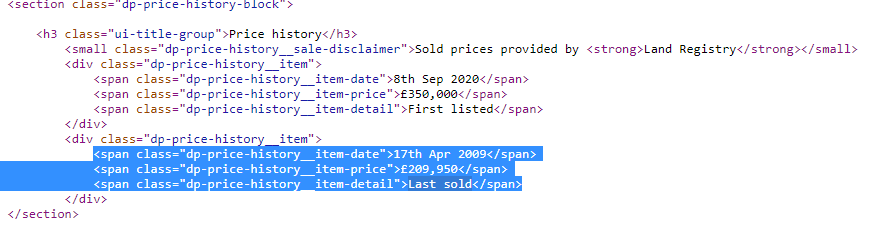
  
The street, locality, town and county are shall be inputted in to the database.

1. Scrape the property description:  
     
   
2. Scrape the agent phone number:

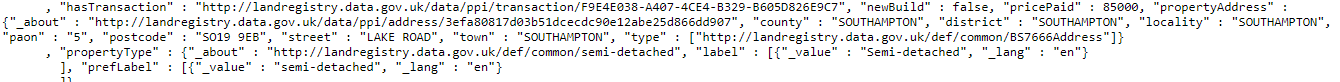


1. Scrape the agent address:



1. Scrape the floorplan link / image (where available):  
   
2. Scrape the first epc image (where available):  
   
3. Scrape the EPC PDF link (where available):  
   
4. Scrape the last sold date and price (where available):  
   With the sold date only keeping the year, ie 17th April 2009 = 2009  
   With the sold price only keeping the digits, ie £209,950 = 209950
5. Scrape a brochure url, these are not marked with a standard name each time though are part of the following code:

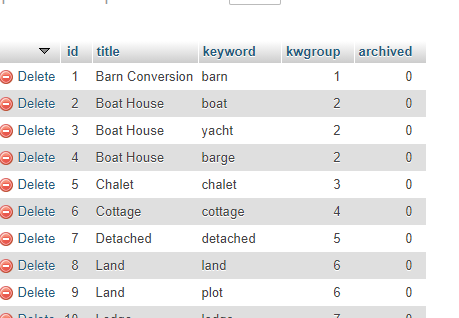
  
Check each link for pdf, that is not an EPC certificate, and return this as the brochure link.

1. Scrape the date listed, as above screen shot, 8th Sep 2020 and convert into: YYYY-MM-DD, so 2020-09-08
2. Put a listing URL together. #section# and #propertyid# is from JSON in step 1 scrape. [https://www.zoopla.co.uk/#section#/details/#propertyid#](https://www.zoopla.co.uk/#section#/details/#propertyid)
3. If a “Sale” Property: Look up full address via sold data and postcode at land registry, again scrape via a GET query for the JSON API, using the following URL:  
   <https://landregistry.data.gov.uk/data/ppi/transaction-record.json?_page=0&min-transactionDate=1996-01-01&max-transactionDate=1996-12-31&propertyAddress.postcode=NR6+7HW&pricePaid=58000>  
   Replace the year with the scraped sold date, the postcode with that from the JSON code and the price paid from the sold data.  
   Ie 2002, 85000, SO19 9EB.  
   The JSON will return a section called “items”, these are the matched rows and address.  
   This will return:  
     
   To which we want to collect the “paon” which should be the door number / building name of the property.   
   Also, if available, collect the “saon” which should be a flat or apartment number.  
   Finally collecting the “transactionDate” which is to replace the sold date.

### Zoopla Data Insertion

1. INSERT INTO ` property\_details\_rightmove` :
   * `property\_id` = propertyID
   * `title` = null
   * `address` = Road Name Town County Postcode (from findAddress)  
     ie Obelisk Road Southampton Hampshire SO19 8BL  
     Salhouse Road Little Plumstead Norwich NR6 7BW
   * `address\_count` = null
   * `price` = as collected
   * `agent` = agent / branch / brand name
   * `agent\_address = agent / branch / brand address
   * `agent\_no` = agent / branch / brand number
   * `agent\_match` = agent name processed
     1. Capitalise text
     2. Strip and remove any html tags
     3. Remove any text after and including a dash -, ie TRISTANS AGENCY – PORTSMOUTH, would become TRISTANS AGENCY
     4. Remove any text after and including a open bracket (, ie TRISTAN ESTATES (PORTSMOUTH), would become TRISTAN ESTATES
     5. Remove any text after and including a comma, ie TRISTAN LETS PROPERTY, Portsmouth would become TRISTAN LETS PROPERTY
     6. Remove any &AMP;
     7. Remove any &
     8. Replace the word company with CO, ie Tristan Lets Company, becomes TRISTAN LETS CO
     9. Remove any of the following words:
        + LTD
        + LIMITED
        + AGENTS
        + AGENCY
        + AGENT
        + ESTATE
        + LETTING
        + LETTINGS
        + SALES
        + RESIDENTIAL
        + INDEPENDENT
        + COMMERCIAL  
          AND
        + MANAGEMENT
        + INVESTMENT
        + CORPORATE
        + .COM
        + FRANCHISE
     10. Remove all white spaces, so TRISTAN LETCO becomes TRISTANLETSCO
     11. If the final letter is a S remove it, ie BELLWAYHOMES becomes BELLWAYHOME
     12. Insert the processed agent name in the database
   * `bedrooms` – beds
   * `property\_link` - photo / image of property
   * `description` - To insert this in to the database later the apostrophe’s and quote marks need to be removed, ie Tristan’s house is in great condition with a kitchen that’s 14”6’ – would be Tristans house is in great condition with a kitchen thats 146
   * `cur\_date` - the date the scrape is being completed, ie CURDATE()
   * `listed` - as per date listed scrape
   * `prop\_address` - same as `address`
   * `url` - as per assembled url
   * `road` - if available paon, soan from land registry, else street / road from findAddress.io  
     Ie: Flat 1, 12 Hook Street  
     Apartment 20, Windsor Court, Bay Road

12 Thorpe Close

* + `town` - locality, district – ie Woolston, Southampton or Thorpe, Norwich
  + `postcode` - enter the postcode
  + `archived` - 0, 1 if listing\_status from json is not to\_let or for\_sale
  + `archived\_dated` - 0000-00-00 or CURDATE()if listing\_status from json is not to\_let or for\_sale
  + `criteria\_id` - id of the crawl / search to find those results
  + `brocher\_link` - scraped brochures url / pdf link
  + `floorplan\_link` - scraped floorplan image url
  + `epc\_link` - scraped epc url if a PDF or a link
  + `epc\_image` - scraped epc image url, if a jpeg / jpg, gif or png
  + `EPC\_EER\_Current` - 0
  + `EPC\_EER\_Potential` - 0
  + `EPC\_EIR\_Current` - 0
  + `EPC\_EIR\_Potential` - 0
  + `EPC\_m2` - from size\_sq\_feet / 10.76
  + `last\_sold\_price` - last sold price
  + `last\_sold\_date` - last sold date
  + `new\_full\_address` - only entered if address was obtained from sold data, if so paon, soan, road, locaility, town, county postcode
  + `property\_type` - property\_type looked up against keyword list in database called `property\_type` returning the kwgroup number to insert in to the database.   
    Ie  
      
    Barn Conversion returns 1  
    Boat returns 2

etc

### Rental Full Address

This crawler finds full property addresses of rental properties by looking through Zoopla Sold listings to find either:

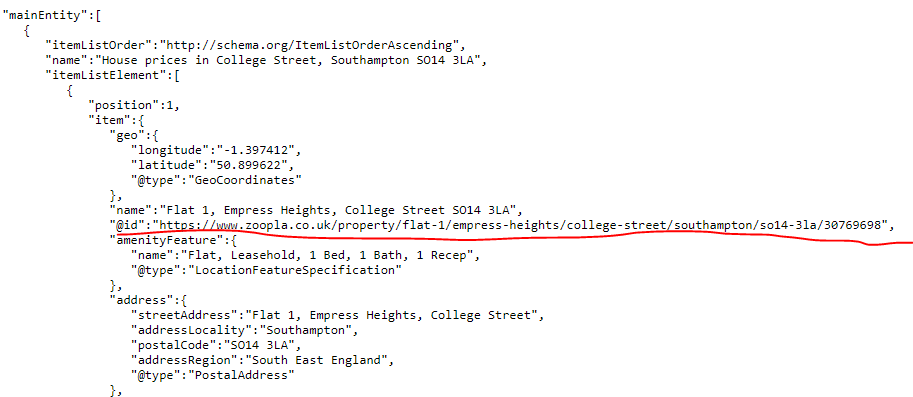
1. Properties linking to the same active listing url
2. Properties in the same postcode, with the same number of bedrooms, same property type, listed by the same agent (typically when looking up rightmove listings)

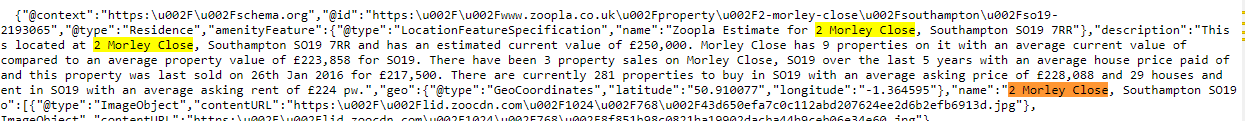
Process:

1. Mysql query to select all “rent” properties from `property\_details\_rightmove` database where “new\_full\_address” is empty and “adr\_lookup” is 0.
2. Crawler heads to:

<https://www.zoopla.co.uk/search/?q=NR6%207BW&section=house-prices>  
Replacing the “NR6%207BW” postcode that that from the mysql query.  
This will redirect to a URL like:  
<https://www.zoopla.co.uk/house-prices/southampton/college-street/so14-3la/>   
Grab the new url.

1. Scrape the total number of results:  
     
   43 in this case, the total number of pages is calculated dividing by 40 and rounding up, ie 43 / 40 = 1.075 rounds up to 2.  
   Now using the grabbed URL the results can be paginated through by appending ?pn=#pagenumber#
2. Scrape the JSON from each page:

  
Grab each of the URLs to crawl through for each paginated page.

1. Now to crawl and run through every grabbed url / page to either:
   1. Find the zoopla ‘url’ from the initial query in the grabbed url:  
      
   2. If a Rightmove link, crawl through each page, find all that contain a `< a href=”https://www.zoopla.co.uk/to-rent/details/’ link.  
      Loop through each grabbed address to find one that matches the property details from the query in step 1, ie property type (flat, house, etc), number of bedrooms, price and `agent\_match` as per Zoopla, Zoopla Data Insertion, Agent Match process.
2. If / when a match is found scrape the JSON:  
     
   Under ‘name’ the address can be found – in this case 2 Morley Close, Southampton SO19 7RR.
3. Whether a match is / or isn’t found ‘adr\_lookup’ for that query / row should be updated and set to 1.
4. If a match is found, the `new\_full\_address` for that property should be updated as per the address found.