

Rets Rabbit for Wordpress® User Guide

Author: Patrick Pohler

Version: 1.0.7

URL: <http://retsrabbit.com>

License: Creative Commons Attribution-NoDerivatives 4.0

<http://creativecommons.org/licenses/by-nd/4.0/legalcode>

Thank you for your purchase! If you have any questions about the module, please email us at contact@retsrabbit.com

Table of Contents

1. Prerequisites
 2. Installation Instructions
 3. Connecting to Rets Rabbit
 4. Metadata
 5. Plugin Shortcodes
 6. Tips & Troubleshooting
 7. Support
 8. Changelog
-

1. Prerequisites

Make sure your system meets the following requirements:

- PHP 5.4+
 - Wordpress 4.0 or later
 - An Client ID & Secret for the Rets Rabbit API
-

2. Installation Instructions

Wordpress Plugin Directory

1. From the Wordpress Dashboard 'Add Plugins' page, you can simply search for "Rets Rabbit" and click install.
2. Click the **Activate** link to finish make the Rets Rabbit plugin active.

Install from the .zip file

1. Download the **.zip** file
2. In the Wordpress Dashboard, go to **Plugins -> Add New**. Choose **Upload a New Plugin** and upload the **.zip** file.
3. Click the **Activate** link to finish make the Rets Rabbit plugin active.

Install manually

1. Download the **.zip** file.
2. Extract the **.zip** file and copy the **retsrabbit** folder and its contents to your Wordpress installation's **wp-content/plugins** directory.

Now that you've finished installing the plugin, go to **Connecting to Rets Rabbit** to finish the setup and start to pull your listings!

3. Connecting to Rets Rabbit

You'll need two things to use the Rets Rabbit plugin to pull live listings:

1. A RETS url, login, and password from your local real estate board or MLS. If you don't have an account, contact us at contact@retsrabbit.com and we can help you get an login.
2. A Rets Rabbit account with a Client ID & Client Secret. To signup for an account, visit [RetsRabbit.com](https://retsrabbit.com).

Once you have your Rets Rabbit Client ID & Client Secret, click the Rets Rabbit section in the Wordpress Dashboard and enter them in the "Rets Rabbit Settings" page. Optionally you might have a custom API endpoint (<https://domain.retsrabbit.com/api>), you can enter that into the "Rets Rabbit Settings" as well.

Once that's saved, you can confirm your access is working by click on "Metadata" under the Rets Rabbit admin to view your local MLS's metadata.

4. Metadata

In order to work with Rets Rabbit, you need to know about the data being returned from the servers you're connecting to. In order to help with this, you can use the Metadata tab.

Metadata consists of the following types of information:

Resource: This defines the type of data to be returned. There are nine standard types, a RETS server may return one or more of these types. The most common type is "Property"

Class: A subtype of the data to be returned. Every Resource has at least one class, while some may return more than one. For instance the Property resource may include separate classes for single-family homes, condos, or commercial properties.

Field: The data is organized into fields, each field contains a value that may correspond to an integer, character or some other datatype.

Object: A special datatype that contains binary data, usually Photos or attached documentation.

The Metadata tab provides a quick reference on the data being returned. It also organizes the metadata so you can quickly reference the proper fields in your templates.

Currently the Rets Rabbit API only pulls "Property" resource types, although the Metadata tab will show all resources available from the local real estate board. If you need other resources imported from your real estate server, please email contact@retsrabbit.com for custom solution options.

5. Plugin Shortcodes

Displaying Listings

The `[retsrabbit-listings]` shortcode runs a query to display Property resource data.

Example:

```
[retsrabbit-listings params='{ "PropertyType": "Residential", "ListPrice": "90000-95000" }' orderby="ListPrice" sort_order="desc" template="listing.php" num_photos="1"]
```

The shortcode parses RETS data fields for the Property resource, it can also use any RETS field as part of the search query. In the above example the listings shortcode will display properties matching the search criteria in the **params** parameter. As you can see **params** is formatted in the following way:

```
{ "MLS_FIELD_NAME": "SEARCH_VALUE", . . . }
```

You can combine multiple search parameters by separating them with a comma (,) to narrow down your listings. The field names will differ from board to board and you can use the **Metadata** tab in the Wordpress Dashboard (see #4 Metadata in the guide).

The shortcode also takes the following additional parameters.

- **orderby** (optional): a property field to sort the results by
- **sort_order** (asc|desc): sort in ascending or descending order
- **template** (default: listings.php): the template file in the `plugins\retsrabbit\template` directory used to show the listings
- **limit** (default: 10): number of results to display
- **paginate** (default:true): used w/ the **limit** parameter, setting this to true will generate pagination links for the results
- **num_photos** (default: 1): the max number of photos for each listing. Default is **-1** (all photos) while **0** will display no photos
- **cache** (default: false): enable cache
- **cache_duration** (default: 4 hrs)

Displaying a single listing

Example:

```
[retsrabbit-listing]
```

Single listings are displayed using the `[retsrabbit-listing]` shortcode. You can use the shortcode on any regular Wordpress page. Next, go to the **Rets Rabbit Settings** page in the Wordpress Dashboard and use the **Detail Page** dropdown and select the page containing the shortcode.

- **mls_id** : this is the unique listing id for the property (often called the MLS number). By default the plugin will look at the URL to grab the `mls_id` parameter, however you can override this by passing the **mls_id** parameter in the shortcode
- **template** (default: 'detail.php'): the template file in the `plugins\retsrabbit\template` directory used to show the detail page for the listing
- **num_photos** (default: -1): the max number of photos for each listing. Default is **-1** (all photos) while **0** will display none
- **cache** (default: false): enable cache
- **cache_duration** (default: 4 hrs)

The **detail.php** template can be modified to display the listing details. The listing data will be in an array called **\$result**:

```
<?
setlocale(LC_MONETARY, 'en_US');
?>
<div class="row">
    <h1><?= $result['StreetNumber'].' '.$result['StreetName'] ?> <?=
$result['City']?>, TX</h1>
    <h2><?= money_format('%#10n', $result['ListPrice']);?></h2>
</div>
<div class="row">
    <div class="gallery">
        <?php foreach($result['photos'] as $photo) :?>
            <div></div>
        <?php endforeach; ?>
    </div>
</div>
<div class="row">
    <p><?=$result['PublicRemarks']?></p>
    <ul class="list-unstyled">
        <li><strong>Size:</strong><?= $result['SqFtTotal']?> Ft<sup>2</sup></li>
        <li><strong>Beds:</strong><?= $result['BedsTotal']?> </li>
        <li><strong>Bathrooms:</strong><?= $result['BathsTotal']?> </li>
        <li><strong>Year Built:</strong> <?= $result['YearBuilt']?></li>
        <?php if($result['ParkingSpacesGarage']) : ?>
            <li><strong>Garage:</strong> <?=
$result['ParkingSpacesGarage']?> Spots</li>
        <?php endif;?>
        <li><strong>Features:</strong> <?= $result['InteriorFeatures']?>
    </li>
        <li><strong>Heating Type:</strong> <?= $result['Heating']?></li>
    </ul>
</div>
```

The **\$result** array has fields that match the property fields for your MLS. The fields and their expected values can be found in the **Metadata** tab in the **Rets Rabbit Settings** page in the Wordpress Dashboard.

Also there's another array **\$result['photos']** containing the photos for the property, of which **\$photo['url']** can be used to pull the URL for the photo. The number of photos that are in the **\$result['photos']** array are controlled by the **num_photos** parameter.

Property Search Forms

While the `[retsrabbit-listings]` shortcode are great for running queries in your templates, there are times when you need to create a search form so your users can run their own search queries.

Building the search form

Use the `[retsrabbit-search-form]` shortcode to include a search form on your page.

```
[retsrabbit-search-form]
```

By default the search form that will be generated will come from the `plugins/retsrabbit/template/search-form.php` template file, you can override it with the optional **template** parameter.

When you edit the default `search-form.php` file you can see the form declaration at the top:

```
<form class="search-form" method="post" action'<?= get_admin_url()?'>admin-  
post.php'>  
<input type='hidden' name='action' value='retsrabbit-search'>
```

DO NOT edit or modify these two lines, they are **REQUIRED** for the search form to work properly!

Everything else feel free to edit. In order to have the user select a price to search by, you could create a drop down field like so:

```
<label for="listPrice">List Price:</label>  
<select name="rets:ListPrice" id="listPrice">  
  <option value="">Any</option>  
  <option value="100000-">Less than $100,000</option>  
  <option value="100001-150000">$100,000 - $150,000</option>  
  <option value="150001-200000">$150,000 - $200,000</option>  
  <option value="200001+">Over $200,000</option>  
</select>
```

Look carefully at the **name** attribute of the dropdown. We're searching the `ListPrice` field, which is the name of the price field for our particular board (this could be different depending on your local real estate board so use the **Metadata** tab in the Dashboard), in order for Rets Rabbit to detect it in the form we prepend **"rets:"** to the field name. This makes the form input name

rets:ListPrice. You must add "**rets:**" in the **name** attribute of every form input for every field you wish the user to search on.

If you want to give the user some options to sort you can add **orderby** and **sort_order** fields to the form.

```
<label for="orderby">Order By:</label>
<select name="orderby" id="orderby">
  <option value="">None</option>
  <option value="ListPrice">Price</option>
</select>
<input type="radio" group="sort_order" name="sort_order"
value="desc">Descending
  <input type="radio" group="sort_order" name="sort_order"
value="asc">Ascending
```

There's a special php array **\$form_data** that contains the values of the search. This is useful if you want to save the user's search selection. Here's an example of how you'd use it.

```
<option value="100001-150000" <?php if ( isset($form_data['ListPrice']) &&
$form_data['ListPrice'] == "100001-150000"): ?> selected="selected" <?php endif;
?>>$100,000 - $150,000</option>
```

Displaying the results

You can display the search results using the **[retsrabbit-search-results]** shortcode. In order for Rets Rabbit to recognize the results page, you use the **Search Results Page** dropdown in the **Rets Rabbit Settings** Wordpress Dashboard.

The results are displayed using the **plugins/retsrabbit/results.php** template by default (you can override by using the **template** parameter).

All results are stored in the **\$results** array, you can loop through the array, each row will contain an associative array containing keys that match your local MLS's fields. Here's an example.

Example:

```

<?php
$detail_page_id = get_option('rr-detail-page');
setlocale(LC_MONETARY, 'en_US');

foreach($results as $result) : ?>
<div class="row">
<div class="col2">
    <?php foreach($result['photos'] as $photo) :?>
        
    <?php endforeach; ?>
</div>
<div class="col4">
    <h3><a href="<?= add_query_arg('mls_id', $result['mls_id'],
get_permalink($detail_page_id)) ?>"><?= $result['StreetNumber'].'
'.$result['StreetName'] ?></a>
        <span class="text-muted"><?= $result['City']?>, TX</span></h3>
    <h4><?= money_format('%#10n', $result['ListPrice']);?></h4>
    <p><?= $result['PublicRemarks']?></p>
    <ul class="list-unstyled">
        <li><i class="fa fa-fw fa-th"></i> <?= $result['SqFtTotal']?>
Ft<sup>2</sup>
        </li>
        <li><i class="fa fa-fw fa-columns"></i> <?= $result['BedsTotal']?>
Beds</li>
        <li><i class="fa fa-fw fa-female"></i> <?= $result['BathsTotal']?>
Bathrooms</li>
    </ul>
    <p><a href="<?= add_query_arg('mls_id', $result['mls_id'],
get_permalink($detail_page_id)) ?>" class="btn btn-primary">View More »</a>
    </p>
</div>
</div>
<?php endforeach;?>

```

There's a lot going on in this example. First, you can pull the detail page using a combination of **get_option('rr-detail-page')** and then using the **get_permalink()** method. This will link the property to the detail page you defined in the **Rets Rabbit Settings** page.

Also there's another array **\$result['photos']** containing the photos for the property, of which **\$photo['url']** can be used to pull the URL for the photo.

Finally the **\$result** array has fields that match the property fields for your MLS.

6. Tips & Troubleshooting

If you have any issues with listings, please check to see if you're using the right fields in your templates. You can view the fields for your MLS board in the **Metadata** tab.

7. Support

If you have any questions or issues, please contact us at contact@retsrabbit.com

8. Changelog

1.0.0 - Initial Release

Initial Release of Rets Rabbit Wordpress plugin to users