Neopets® Stuck Pet Finder

yes really

Emily Liu

https://github.com/Ellime/NeopetsStuckPetFinder

Last updated: 01/02/2021

Contents

- Overview
- Technical Outline

Overview

In short

- A tool that finds Neopets in the Pound
- Neopets still exists

Context

Neopets.com is a game website about owning virtual pets that has a Pound (think animal shelter) with unowned pets.

Players are supposed to refresh the Pound site for pets to adopt.

Some pet names are impossible to find via refreshing – referred to as “stuck” pets. You must directly search them by name.

Names that “stick”

More info at:

http://www.neopets.com/~Megablade

http://www.neopets.com//~Lorrenn

Which names will get stuck?

It's complicated and always changing but a current list of strings that stick is as follows:

ANG-, ARG-, ASH-, AUS-, AVA-, AZU-, A_B-
BEN-, BIG-, BLA-, BOB-, BRA-, BUB-
CHE-, CIN-, CLA-, COC-, CRA-, CUD-, CYN-
DIN-, DOL-,DRA-, DUD-
FLI-, FOF-, FRA-, FUI-
GOR-, GRA-, GUI
HUN-
JOS-, JUB-
KIK-, KOK-, KRB-, KUK-, KYL-
LEO-, LIG-, LLA-, LOC-, LUC-, LYN-, L_X-
MAR-, MEE-, MIC-, MOE-, MRL-, MUG-, MYE-
NIP-, NOH-, NUN-
PIN-, PLU-, POK-, PRE-, PTE-, PUF-
RIK-, ROB-, RUB-, RYH-
SCR-, SEB-, SHA-, SIL-, SKE-, SLI-, SME-, SNI-, SOF-, SPA-, SQU-, STA-, SUG-, SWE-, SYN-
TIG-, TOC-, TRA-, TUF-, TWI-, TYL-
_SN-, _TO-, __G-

How to use

1. Enter your first account’s username and password as two lines in credentials.txt
2. Login to a second account on your default browser
3. Launch the program
4. Set up the rules you want to use when generating pet names
5. Hit “0” to run

Breakdown of program

1. Generate potential pet names
2. Search each name using the Pound search
3. Check if the search returned a pet
4. For names that returned a pet, open that link in the browser and save the name and color/species to a file

NOTE: Searches are throttled – purposefully slowed down – so the website isn't overwhelmed.

Methodology (with Neopets lingo)

- The Pound search is the only guaranteed way to check if a pet is in the Pound and adoptable
  - Although default petpages can indicate if a pet is unowned, owned pets may have modified petpages, causing the response body to be unpredictable.
  - The petlookup indicates if a pet is unowned, but the URL is actually a redirect and some pets are glitched (have a lookup but cannot be adopted).
- Limitation: Viewing Pound requires being logged in
  - Same goes for the above sites.
  - The account must also be > 4 months-old (or some pets won't show up).
  - Alternative: Instead of checking if pet is in Pound, check if pet exists by querying its image file.
    - But this will return pets that are not in the Pound (are owned/frozen/glitched).

Impacts

Pros

- Can find some lonely pets.

Cons

- Without limiting the number of requests, can overload the server and create a DoS attack
- Can cause a DDoS if multiple people run it together
- Can be misused by people selling pets/accounts for real $ which is against the ToU and ruins things for innocent players

Is it allowed? (Probably not)

- This program is fan-made and not officially recognized.
- Neopets.com ToU is outdated and useless.
- Neopets has stated all programs/scripts are disallowed.
  - http://www.neopets.com/ntimes/index.phtml?section=editorial&week=383
  - http://www.neopets.com/ntimes/index.phtml?section=editorial&week=797
- Small hope: there is an approved manual search method, even though it can potentially send hundreds of requests a second.
  - http://www.neopets.com/~Megablade

Tips

- Having two accounts is best – one is used by the program and the other allows you to see the pets in your browser as it runs. If you just have one account you will have to check out the pets in your own time.
- Keep in mind what name formats to expect! The machine-suggested names follow the following format: “dragon_12” or “dragon_12_1”

Final words

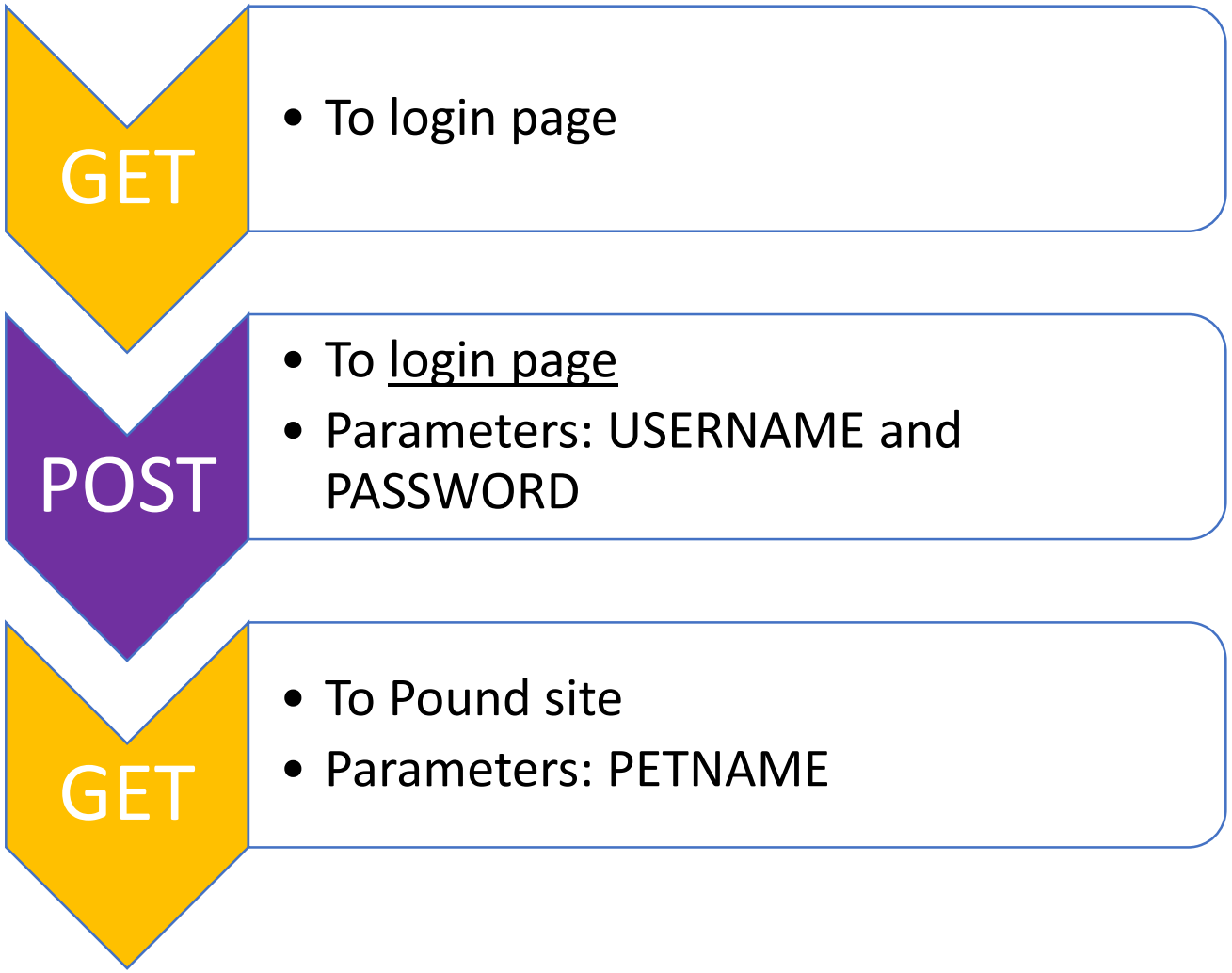
The project code is not publicly available. This is a fanmade project that has not been acknowledged by Neopets.com.

I am willing to discuss the project but will not share the code privately nor publish it unless it is officially supported.

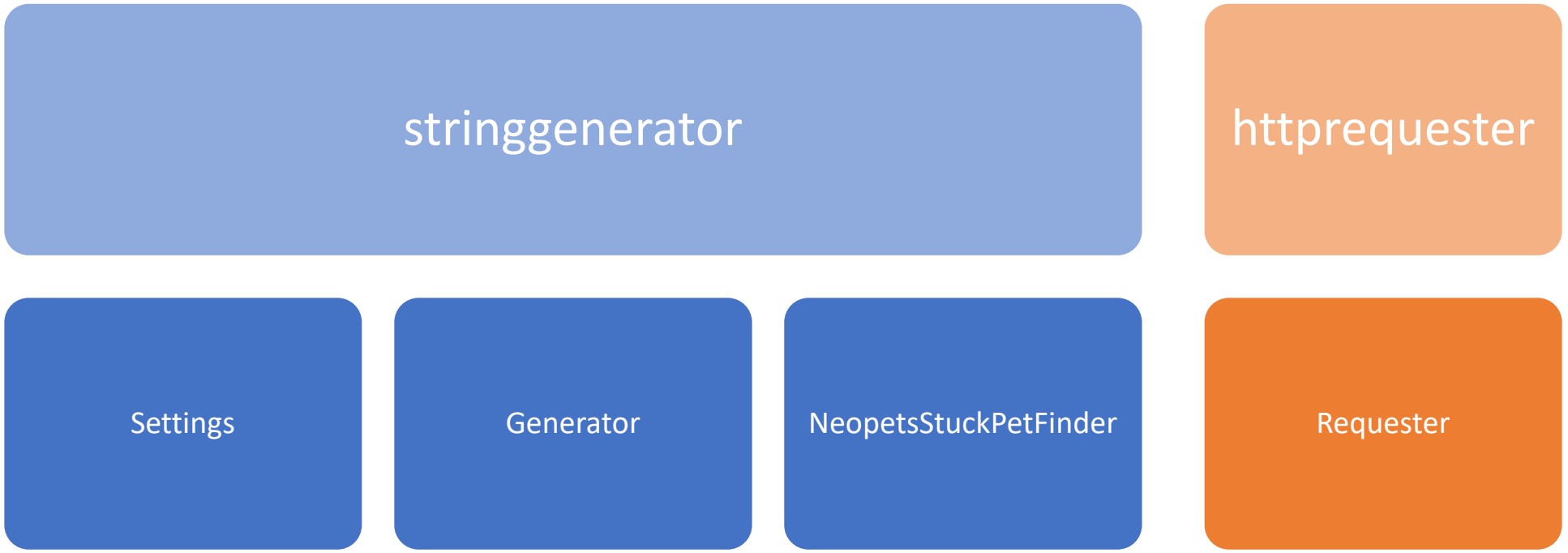
Technical Outline

Program steps

1. Send GET to login URL
2. Send POST to login URL and store the returned session ID
3. Generate strings
4. Send GETs to Pound URL while passing one string at a time



Package/class hierarchy



String generation

String generation is modified from https://github.com/Ellime/StringGenerator

- Settings: Saving/setting string generation setup
- Generator: Generates strings
- NeopetsStuckPetFinder: main() and console output

Modifications:

- Can include '_' in the range of available chars
- Option 0 includes generating strings and calling functions from Requester

HTTP request handling

Requester: Handles HTTP requests/responses, plus what to do with found pets

Login

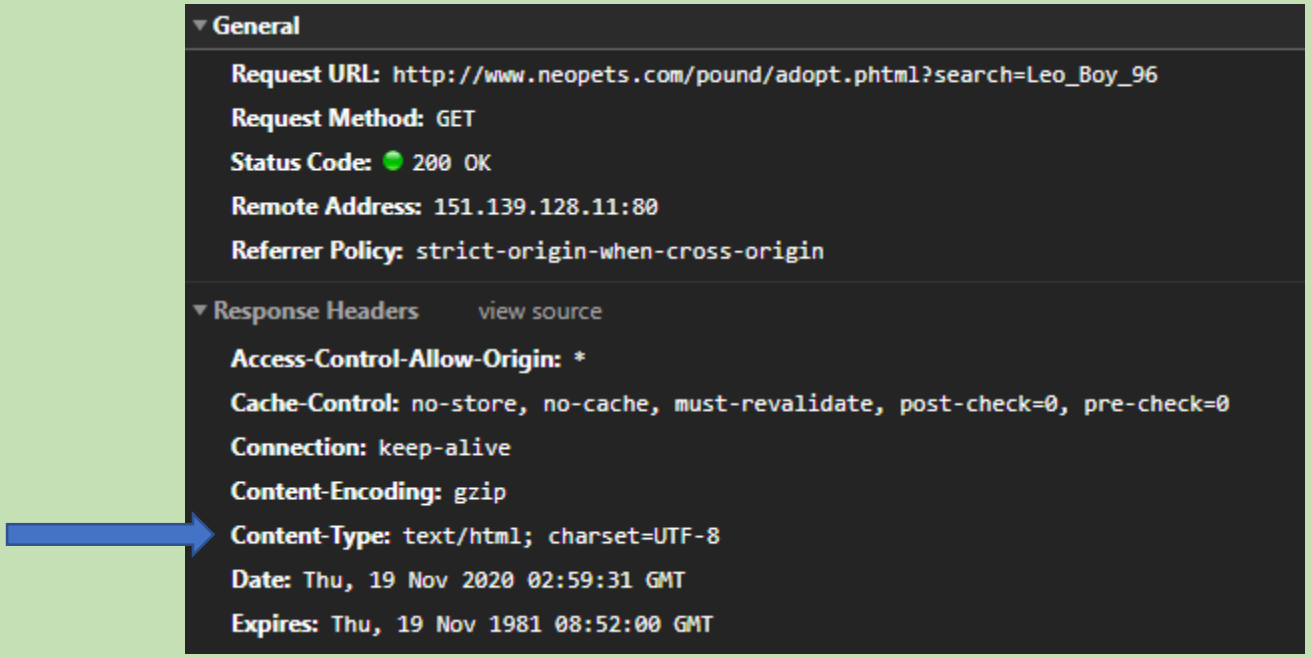
Credentials stored in credentials.txt

- Send GET to http://www.neopets.com/login/
- Prepare username and password as parameters to send in POST
- Send POST to http://www.neopets.com/login.phtml
- Save the cookies for future requests

Search

- Send GET to http://www.neopets.com/pound/adopt.phtml?search=PETNAME
- Use jsoup to check if PETNAME is a pet in the Pound
- If valid, open the link in the browser and save the name

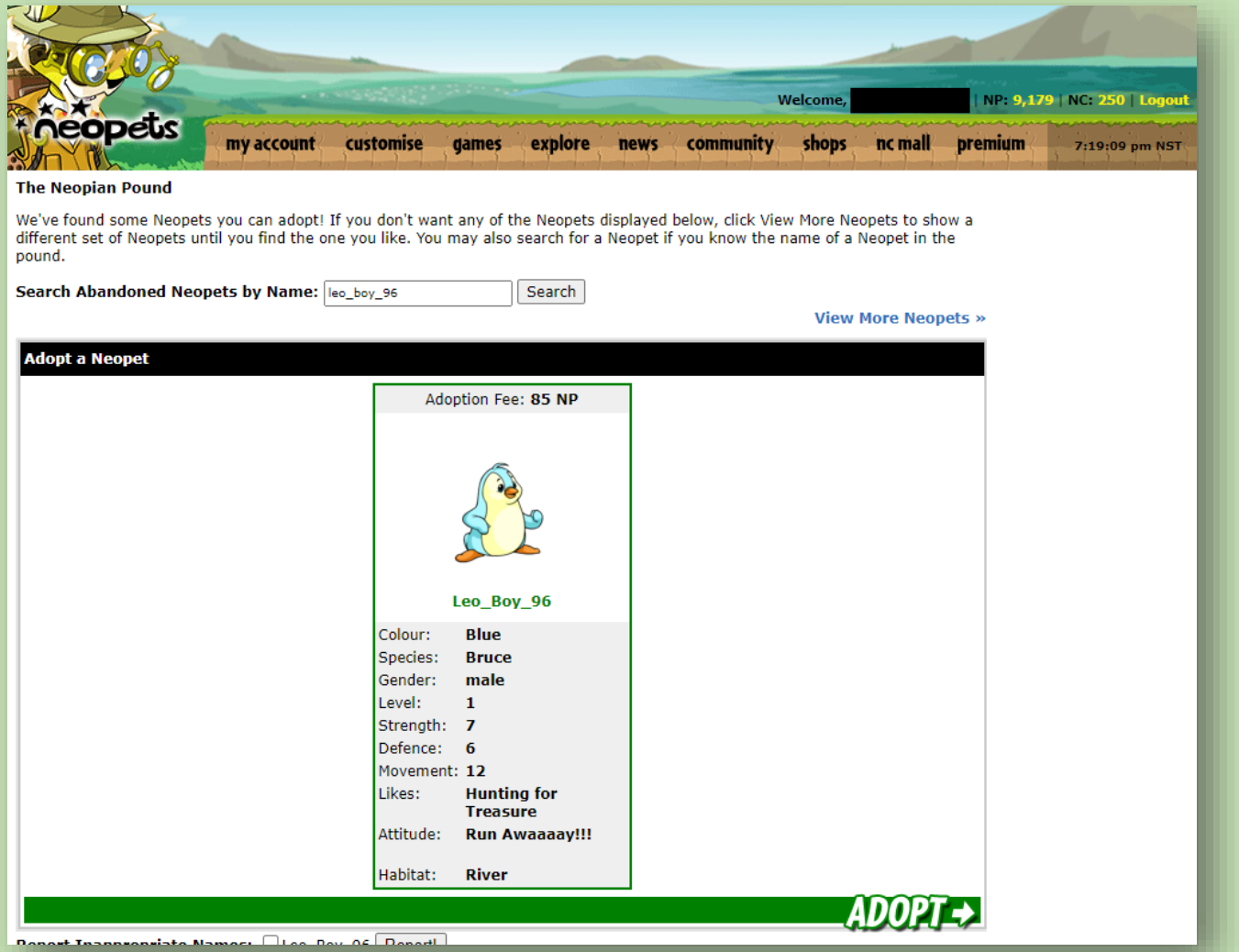
How the search works



Pet found in the Pound

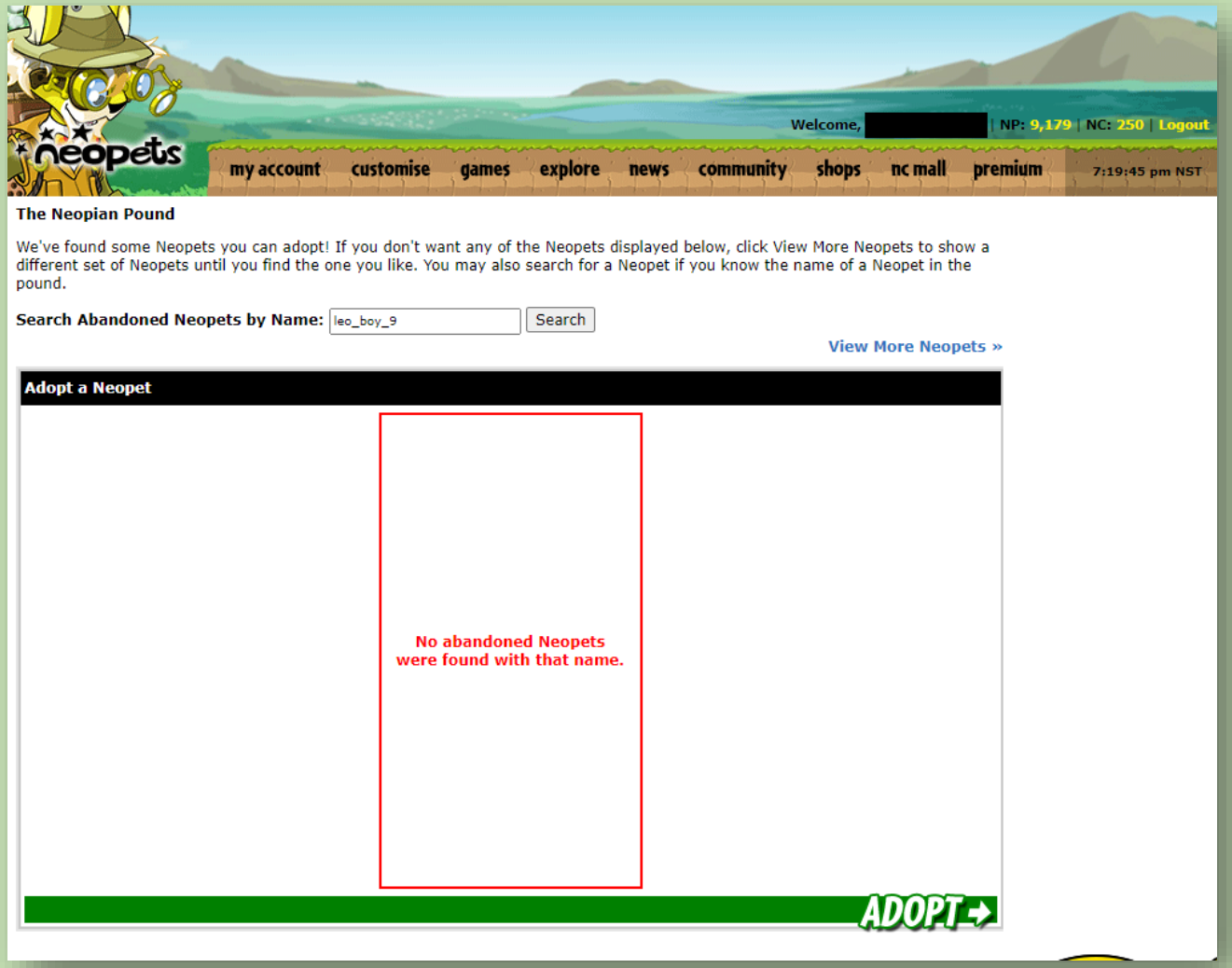
Pet names are unique and caps-insensitive so a query returns at most one pet.

(The site is designed so up to three pets can be displayed, which we will see in the HTTP response body.)



Pet not in Pound

The difference is reflected in the HTTP response body.



HTTP response body (JS)

Pet found: Its features are initialized as the second (middle) Object.

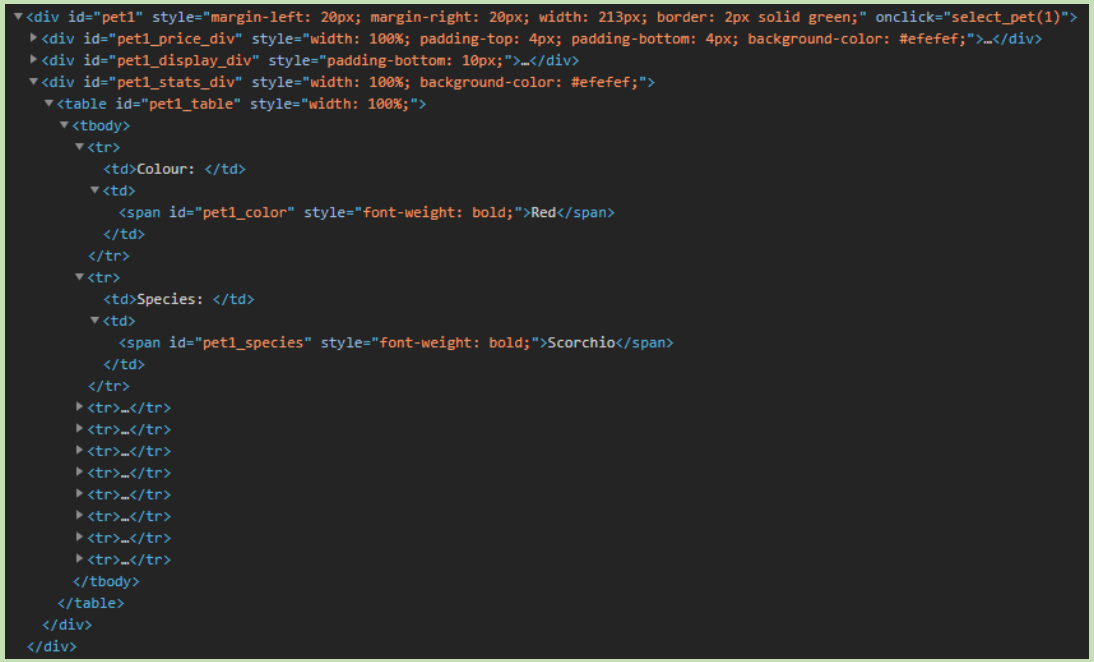
```
511 <script type='text/javascript' src='http://images.neopets.com/pound/adopt.js'></script>
512 <script type='text/javascript'>
513 var selected_pet = 1;
514 var pet_arr = new Array(3);
515 var loadingImg = new Image();
516 loadingImg.src = 'http://images.neopets.com/pound/pound_load_anim.gif';
517 pet_arr[0] = new Object();
518 pet_arr[1] = new Object();
519 pet_arr[1].name = "Leo_Boy_96";
520 pet_arr[1].price = "85";
521 pet_arr[1].color = "Blue";
522 pet_arr[1].species = "Bruce";
523 pet_arr[1].gender = "male";
524 pet_arr[1].level = "1";
525 pet_arr[1].str = "7";
526 pet_arr[1].def = "6";
527 pet_arr[1].speed = "12";
528 pet_arr[1].mood = "2";
529 pet_arr[1].likes = "Hunting for Treasure";
530 pet_arr[1].attitude = "Run Awaaaay!!!";
531 pet_arr[1].terrain = "River";
532 pet_arr[1].happyImg = new Image();
533 pet_arr[1].happyImg.src = 'http://pets.neopets.com/cpn/Leo_Boy_96/1/2.png';
534 pet_arr[1].sadImg = new Image();
535 pet_arr[1].sadImg.src = 'http://pets.neopets.com/cpn/Leo_Boy_96/2/2.png';
536 pet_arr[2] = new Object();
537 </script>
```

Pet not found: Objects remain undefined.

```
511 <script type='text/javascript' src='http://images.neopets.com/pound/adopt.js'></script>
512 <script type='text/javascript'>
513 var selected_pet = 1;
514 var pet_arr = new Array(3);
515 var loadingImg = new Image();
516 loadingImg.src = 'http://images.neopets.com/pound/pound_load_anim.gif';
517 pet_arr[0] = new Object();
518 pet_arr[1] = new Object();
519 pet_arr[2] = new Object();
520 </script>
```

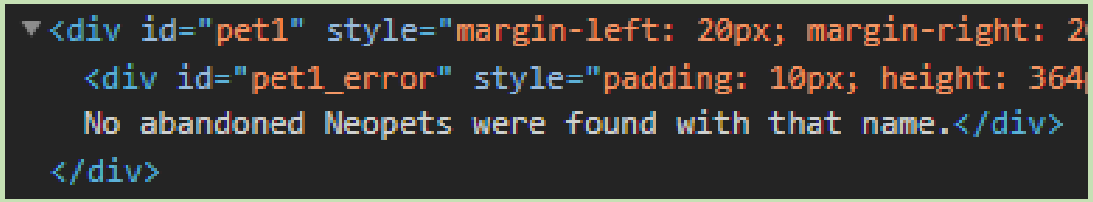
HTTP response body (HTML)

jsoup will read *pet1_color* and *pet1_species*.



```
<div id="pet1" style="margin-left: 20px; margin-right: 20px; width: 213px; border: 2px solid green;" onclick="select_pet(1)">
  <div id="pet1_price_div" style="width: 100%; padding-top: 4px; padding-bottom: 4px; background-color: #efefef;">…</div>
  <div id="pet1_display_div" style="padding-bottom: 10px;">…</div>
  <div id="pet1_stats_div" style="width: 100%; background-color: #efefef;">
    <table id="pet1_table" style="width: 100%;">
      <tbody>
        <tr>
          <td>Colour: </td>
          <td>
            <span id="pet1_color" style="font-weight: bold;">Red</span>
          </td>
        </tr>
        <tr>
          <td>Species: </td>
          <td>
            <span id="pet1_species" style="font-weight: bold;">Scorchio</span>
          </td>
        </tr>
        <tr>…</tr>
        <tr>…</tr>
        <tr>…</tr>
        <tr>…</tr>
        <tr>…</tr>
        <tr>…</tr>
        <tr>…</tr>
        <tr>…</tr>
      </tbody>
    </table>
  </div>
</div>
```

The absence of *pet1_table* tells jsoup we didn't find a pet.



```
<div id="pet1" style="margin-left: 20px; margin-right: 2
  <div id="pet1_error" style="padding: 10px; height: 364
  No abandoned Neopets were found with that name.</div>
</div>
```

Server behavior

- Login POST request returns a 403 – which is fine since all that's needed are the authentication cookies
  - The reason for the 403 is because the URL we send the POST to is not a URL we can actually "visit".
- A 503 is returned if too many requests have been sent recently
  - Throttling is required to avoid this.

Request throttling

Pauses are implemented between searches to prevent flooding the server with requests.

For example:

- Every 2 searches: Wait .5 sec
- Every 20 searches: Wait 20 sec
- Every 100 searches: Wait 1 min

If a 503 is encountered, the program will wait 1 min before retrying the name it failed to search.