**Unclaimed Money - About**

Every state in the USA has millions of dollars of unclaimed money every year. This money/property is initially held by organizations who hadn’t been able to contact its owner. After a certain period of time and as per state rules, this money is turned over to the state. The state then becomes the custodian of this money until it is claimed by the owner. Unclaimed Money is a USA-based organization that helps its citizens claim this money. The website consolidates this information from various state databases and federal resources into a central site. Hence it makes the search for unclaimed money quicker and smoother. For this project, the primary requirement of the client was to build a basic form to demonstrate the data scraping capability. The form will enable users to enter their first and last names, state and business names (if any). Now the program will search in the respective state sites and fetch a list of matched data. Later, this program will be used as an API.

**Advanced Search**

While we had implemented a search by first name, last name, state and business name, a little research made us aware that other similar sites use more search fields. So, we decided to provide an advanced search for users by including numerous other filters. This helped make the search results even more accurate.

**Storing User Search Data**

The basic premise of the user search is using keywords like first name, last name or business name. We used an API to collect the search results from the state/federal database and display it on the website. Once this data is fetched, we stored it in our server. This way, when the user searches with the same keywords again, we would show the result from our database. This reduced the API load. This data is updated at a regular interval to fetch only the latest and most relevant results.

**Implementing OAuth Token**

Security measures need to be implemented to make the search process threat-free. The Key Authorization policy turned out to be an efficient way to make this possible since it restricts access to API endpoints for applications through API keys. This means every request sent by the user to the API needed to be identified by our application. So we provided a private key to each user of the site. This way, we could check and authorize users. To prevent further malicious attack we provided a time-based token key to the users. In order to use the Key Authorization policy, token-auth credentials were to be created for them. After the token expires, the user has to request for another token to continue using the services of the website.

**Post Search Functionality**

So what happens after you find out that you have an unclaimed property in your name? We have implemented a functionality that allows user to directly visit the site that we have pulled the data from and take necessary action as per the site’s guidelines.

**Technology Specifications:**

**Codeigniter, PHP, MySql, Rest API, Simple html dom, DataTable**

**Challenges:**

We had a slew of challenges while working on this project:

* Scraping data from websites that have captcha validations
* Scraping data from US government sites without getting blocked
* Searching the data runtime and showing the latest and most relevant result
* Building the site like an API so that it can be called as REST API at a later stage

**Solution:**

* For scraping data, we used curl calls and simple HTML dom libraries
* We used solving services to scrape data from websites with captcha
* To prevent getting blocked we used the proxy network. The bonus was that it could handle thousands of requests.
* We implemented REST API in the site and added further functionalities

**Result:**

We were able to overcome all the challenges with effective solutions that fulfilled the purpose of the project.

<https://www.unclaimedmoneyfinder.org/>

<https://www.beenverified.com/iframe/unclaimed/>  
<https://www.peoplelooker.com/iframe/unclaimed/>