

Database and Web Techniques Report



Proxy Aggregator

By
Vidyadhar Mallappa Kadasani
614386
Web Engineering
Summer Semester – 2020

Content

Introduction	-----	1
Application Walkthrough	-----	3
Appendix	-----	12
References	-----	18

Introduction

Internet services come with their boons and banes. A person browsing over the internet using personal devices is exposed through his device IP address and which may lead to a security breach or phishing scam. Over the years, we have started using Proxy servers to avoid these things and gain privacy while surfing the internet. The problem with the internet is that the data theft occurs in most usual click-bait that leads you to compromise your privacy, such as " know your fortune in five mins", " unrealistic discounts on expensive products" or even on the link to a cute dog picture.

What is a Proxy ? Probably a well-known question that most of the people might know the answer. A proxy or a proxy server provides functionality to browse the content over the internet without exposing your IP address, in other words, it is a gateway between you and the internet acting as an intermediate server eavesdropping the end-user from the website that they want to browse. Proxy servers are supplied with a diversified level of functionality, security, and privacy which are contingent on use cases, company policies and needs

Proxy servers are in high demand after its ability to ensure user privacy and data security, and the order was so tall that it led to business opportunities for many companies to provide proxy servers to the end-users and gain profit from them. There are tons of providers in the market that deliver proxies which are not active more than a month and provide fresh proxies every other month. For this, the buyer had to have a monthly subscription. Whenever a person needs to buy such proxies, he needs to search for the provider, look for sample data and then buy it. Since all the providers have similar tasks to provide proxies to end-users, they maintain the same standard format, which is used in this application to give a one-stop platform for the End users.

The process will help to see the list of proxy providers and their information such as IP address, Port, Base Url address, number of proxies the providers have to offer and whether the proxies work or not. This web application helps users to go through varieties of providers in the market and offers help to choose the required provider as their needs and avoid hassle to search for providers and map those providers for manually.

These Proxies can be created using software like Adobe, Premiere, and Media encode in using formats like QuickTime. After creating your proxies, you can provide it to the end-users typically. Two types of proxy servers define proxies categorically, such as open proxies that act as forwarding proxies that are used by millions of users around the globe.

Reverse Proxies also are known as surrogate proxies are the servers that forward one or multiple servers in response to the request made by the user. Even though the reaction is from the proxies server, it looks as if it initially came from the Main server. While setting up the reverse proxy server, we need to keep several aspects that influence the need of these reverse proxies. Such as Encryption for the server using secure socket layer, load balancing to evenly distributed through the web servers, compression helps in speeding up the load time by optimising the servers, security policies to maintain standard protocols. Spoon feeding which helps in the promotion of slow loading clients on web servers. Extranet publishing which is mainly for the proxies that are used to face the internet and made to communicate with the firewall server that was posed by the internet organisation.

Application Walkthrough

The Proxy Feed aggregator web application gathers all the latest proxies from various proxy providers taking their base website as a source. And exhibit required fields of information, such are IP Addresses, Port Number, Country name, Updated Date and time, created date and time and last accessed date and time. Only by Fetching data from the Base URL and putting into a structured table arrangement. The application provides two different views which are Proxy information view, and the other one is Providers view. Fig. 1 shows the information that is in Proxy Details View.

Refresh Proxies :

 Refresh

Last Refreshed at : 2020-06-23 23:15:45.750

Proxy Data						
IP	PORT	COUNTRY	LIVE	CREATED DT TM	UPDATE DT TM	PROVIDER
<input type="text" value="46.10.199.90"/>	<input type="text" value="34607"/>	<input type="text" value="Bulgaria"/>	<input type="text" value="No"/>	<input type="text" value="Jun 23 2020 3:44PM"/>	<input type="text" value="10:41:22 30.05.2020"/>	<input type="text" value="byteproxies.com"/>
185.130.105.119	65357	Netherlands	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
1.1.1.215	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
88.198.24.108	3128	Germany	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
104.27.142.28	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
1.1.1.76	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
104.28.10.88	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
1.0.0.253	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
1.1.1.18	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com
1.0.0.251	80	United States	No	Tue, 23 Jun 2020 15:02:37 GMT	Tue, 23 Jun 2020 15:02:37 GMT	proxy11.com

Fig. 1 Proxy Data

Table:

The above Fig 1 shows the structured data that is being displayed in the tabular format. The Features of the Table are: it provides the number of data entries accumulated into the Table from various providers or sources. The end-user has command over the Table and can control on listings that displayed in a single page. And can even modify according to the needs and perspective. The Table also provides information about the total number of pages that it holds. The navigation controls on the Table help the user to navigate to next or previous pages conveniently. The columns of the Table are also very accessible, which can be used to filter or sort the data that are in the columns. The Table also provides a dropdown box to sort the column data as needed by the user. The Table can also hold the data which are URL links embedded into it so that users can have more information if required that will redirect to the particular Websites where the data fetched initially been. By default, while bringing the data from the websites, the data is sorted and displayed into the Table.

Refresh and Update Proxy Data:

The Refresh button on the left top, as shown in Fig. 1 which will fetch new data into the Table every time if there is an addition to the proxies list in the original website. This functionality will help to get the latest proxies that the proxy providers will update on their website. If the data published on the website are aged more than 7 days from the present date, then those data will not be fetched. This application also makes sure that the duplicate data is removed while bringing the data by comparing whether the information which is being delivered is the database or not. Once the data is successfully written into the database table will be updated with new data that has been added to the database.

The functionality of not letting the user refresh for another 10 mins once it's already been refreshed is also included to avoid the user hitting the refresh button in a small interval of the time. The application will send an alert notification to the user the refresh is not available for another 10 mins of the time. Figure Fig 2 will show the Alert message for the same.

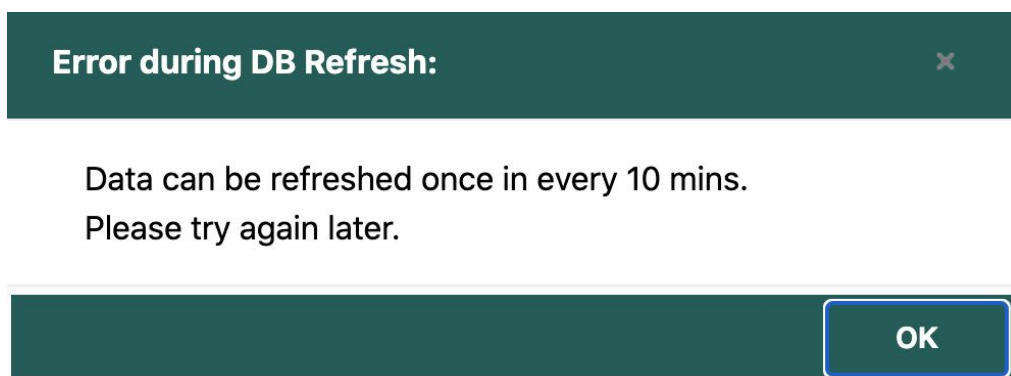


Fig 2 Alert Message to Refresh after 10 mins.

Last Refreshed Data-Time :

The application also provides the last refreshed date time information which is present right next to the Refresh button. The data is about to check when was the last time when the user has added data into the database, or date-time information of the previous refresh button hit as shown in the figure Fig 3 which helps the user to determine whether or not to refresh the Table for the new data. Every time the refresh button has used the information about the time and date in this field is set.

Last Refreshed at : **2020-06-25 16:24:42.440**

Fig 3 Last Refreshed Date-time

Exporting the Data :

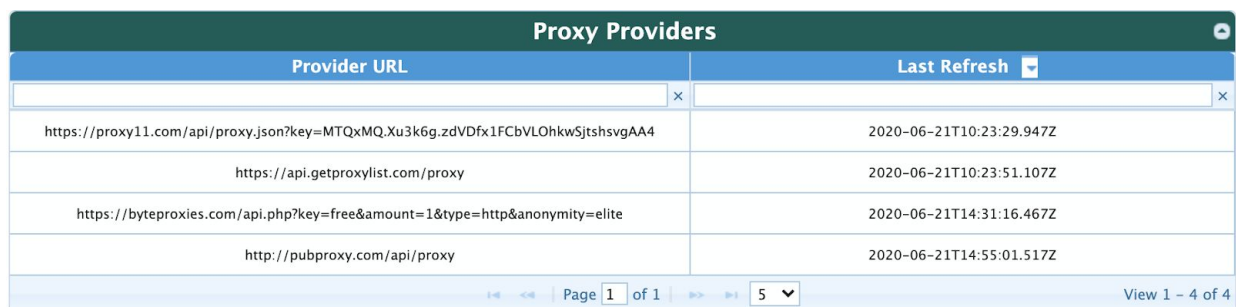
The application also provides the feature to export the data that is being displayed in the tabular form into Excel CSV format. The user can export the proxy information by merely clicking on the button in the left bottom of the Table, and a.CSV file will be downloaded for the user preference. The exported file consists of all the required fields such as IP address, Port number, Base URL, Country Name and so on. The following figure Fig 4 shows the look of the CSV file.

	A	B	C	D	E	F	G
1	ID	IP	PORT	CREATED_DT	UPDATE_DT	PROVIDER	
2	1	201.217.55.9	8080	Jun 25 2020	#####	pubproxy.com	
3	2	88.12.19.206	4145	Jun 25 2020	Jun 25 2020	api.getproxylst.com	
4	3	190.187.253.	3128	Jun 25 2020	#####	pubproxy.com	
5	4	1.0.0.211	80	Jun 25 2020	Jun 25 2020	api.getproxylst.com	
6	5	134.209.29.1	8080	Jun 25 2020	#####	pubproxy.com	
7	6	88.212.61.14	4145	Jun 25 2020	Jun 25 2020	api.getproxylst.com	
8	7	46.10.199.90	34607	Jun 23 2020	10:41:22 30.	byteproxies.com	
9	8	185.130.105.	65357	Tue 23 Jun 2	Tue 23 Jun 2	proxy11.com	
10	9	1.1.1.215	80	Tue 23 Jun 2	Tue 23 Jun 2	proxy11.com	
11	10	88.108.24.10	3128	Tue 23 Jun 2	Tue 23 Jun 2	proxy11.com	

Fig 4 Exported CSV file.

Proxy Providers View :

The proxy provider view is the information panel that provides the details of the proxy providers in the tabular form. The information such as the Provides base URL and the last added data-time for the specified provide information. Figure Fig 5, as shown below.

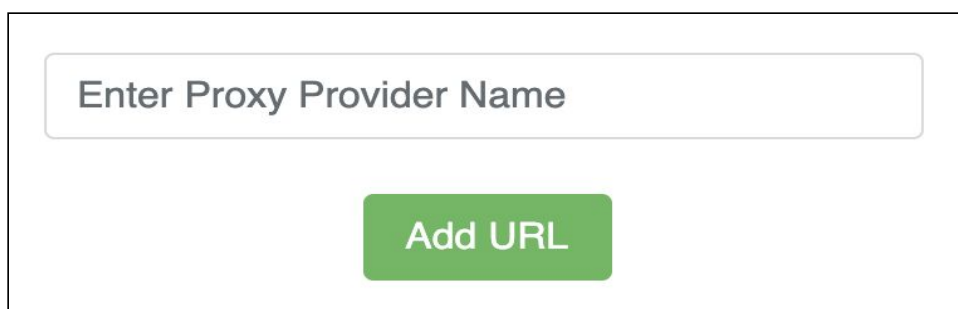


Proxy Providers	
Provider URL	Last Refresh
https://proxy11.com/api/proxy.json?key=MTQxMQ.Xu3k6g.zdVDFx1FCbVLOhkW5jtsHsvgAA4	2020-06-21T10:23:29.947Z
https://api.getproxylst.com/proxy	2020-06-21T10:23:51.107Z
https://byteproxies.com/api.php?key=free&amount=1&type=http&anonymity=elite	2020-06-21T14:31:16.467Z
http://pubproxy.com/api/proxy	2020-06-21T14:55:01.517Z

Fig 5 Providers View.

Add Provider:

Users can add new providers with additional functionality. All new providers will be added to the database, and the proxies that they provide will be fetched into the proxy feed aggregator table. The user must provide the base URL in order to bring the proxies. The wrong proxies will not be taken into the application, and an alert message will be sent to the End-user. Figure Fig 6 shows how to add the provider.



Add URL

Fig 6 Add New Provider

Delete Old Records :

The SQL server is programmed in such a way that it can detect the records which are older than seven days from the present day will be erased from the database. A stored procedure is written to carry out this operation, as shown in Figure Fig 7 below.

```
1  SET ANSI_NULLS ON
2  GO
3  SET QUOTED_IDENTIFIER ON
4  GO
5  CREATE PROCEDURE [dbo].[DELETE_OLD_RECORDS]
6  AS BEGIN
7      DELETE FROM TBL_PROXY
8      WHERE DATEDIFF(DAY, ACTIVITY_DT_TM, GETDATE()) > 7
9  END
10 GO
11
```

Fig 7 Delete records older than 7 days.

Technology Utilized

Web Applications are a new norm in this era of the internet, and we need to be very careful in choosing the technology to build a modern web application that supports futuristic functionality and is durable for long term usage. If the technology is old and cannot be used to provide the required features and functionality, then it would be hard to adapt to the application environment. There are tons of technologies available on the internet which offer future proof functionality and make it difficult for the developer to choose from them. Hence it is essential to decide based on requirement and which technology can meet the expectation to build the application. The following table 1 shows the stack used to create this web application.

Database	Backend
MsSQL Server 2014	Node.js 12.18.1 LTS
Rest Interface	Frontend
Get, Post, Delete	HTML 5, CSS 3, BOOTSTRAP 4, AJAX, JQuery 2.2.4, Jqgrid 4.14.4, FontAwesome

Table 1 Technologies Used

MSSQL Server :

MSSQL Server Released in 1989 with its first version name 1.0. Many reputed companies use Microsoft's MSSQL across the globe due to its stability and ease of usage. MSSQL has a lot to provide for the user with vast inbuilt functionalities. The main reason to use MSSQL for this application is that it allows for a better storage structure for data which follows a standard structure. RDBMS is better for this kind of data when compared to NoSQL. Addition to this stored procedure, index etc. makes the execution smooth and fast when in consideration of massive data that is being thrown from the web sources.

Node.js :

When we think about computational capabilities on the web applications, we always keep the development on the client-side, and it is well known to everyone that javascript is one of those languages that does wonders in development of applications enhancing the computational abilities. When we want to add some computational capabilities on the server-side, the extension of javascript will come into the picture, and the expansion is known as Node.js. Node.js acts as an asynchronous and single-thread operated event. Since node.js looks promising on career development and a well-known technology that is being adopted by many mainstream players in the IT industry, it is without a doubt a right choice, to begin with, Node.js. Node.js is known for its resources which can be accessed by NPM (Node Package Manager). In this library, you can find a considerable number of packages that are available to use in your project just by installing those packages into your project files. Along with Node.js, a well-versed framework known as Express.js is being adopted into this project to deal with middleware setups and routing to develop a sturdy backend functionality. The usage of these two technologies will aid in accessing the data from the database frequently.

Frontend :

User interface (UI) and User Experience (UX) are the two deciding factors that define a web application appearance, the more comfortable and interactive the application is the more the user will be happy to use it. The frontend development involves technologies like HTML, CSS and Javascript. HTML being a skeleton of every web page, undergone through profound changes over the decades. The current version of the HTML that is used in this project is HTML 5—using HTML features in this application, all the interactive and visual elements like Table, buttons and home page of the web application. CSS (Cascading style sheet) As mentioned before HTML acts as a skeleton of a web page a CSS will be its flesh and bone. CSS is mainly used to beautify the HTML pages for User Interface (UI) it is used to style the attributes of the HTML elements as needed. This application is built on CSS 3 for styling the characteristics of the factors such as colour coding, table design, floating buttons, button background colour etc. All these elements are defined in external files and not in inline code. You can access the external CSS file into your HTML only by using the link element in the header section of the HTML. Javascript is a scripting language to create dynamic and interactive web applications. The codes in the Javascripts are known as scripts. We can include Javascript into an HTML file as part of it and get it loaded along with the HTML or we can write the code in a separate file and can access it through the HEAD tag element in the HTML file using the LINK tag. In this application a popular javascript extension known as jQuery. jQuery has been written into a separate file and accessed into HTML. jQuery is a javascript library that helps the developers to use the javascript into their application quickly. jQuery can make code into a single function by compressing it. jQuery is used in this application for interaction purposes such as refresh, delete, insert data etc. Along with HTML, CSS and jQuery, I used Bootstrap 4 for a responsive web application which uses the concept of Grid so the web application can be used in mobile web browsers also.

JqGrid:

To represent and manipulate the data in tabular form on the web, I used jqgrid. jqGrid is a jQuery plugin that is AJAX-enabled. To load and render the data onto the client-side, we use AJAX callback. jqGrid is compatible with any of the server-side technologies, such as PERL, NODE.JS, PHP etc. Being a javascript library jqGrid is supported by most of the modern-day web browsers. I use jqGrid to represent the structured data into the tabular format the application contains columns such as IP, PORT, Created date and time, Updated Data time and base URL. jqgrid inbuilt functionality helps developers to build web pages with a tabular form quickly.

FontAwesome:

FontAwesome is a third party plugin for ICONS and Fonts; they are innovative and are free to use in our web application. Fontawesome is built using CSS and LESS. To improve the appearance of the form, I have used icons for user experience. In this application, the Refresh button Symbol is one such example.

Alertify:

Notifications or alert messages are a very crucial part of the application, which will guide the user to certainty and provide valuable information about the application process. For example, if the data is fetched from the database and the user should be notified about the number of records that are brought then an alter message will do the job. The primary web application alert dialogue box is not promising in appearance; the main object of using Alertify is to beautify the appearance and make the alert dialogue box more visually appealing.

Appendix

Rest Interface:

REST abbreviated as Representational state transfer or you may come across this terminology as RESTful API is built to utilise the protocols that are already existing. Based on HTTP requests such as GET(get the data from the server), POST(add any data to the server), DELETE (remove data from the server) PUT(for update). The primary object of the REST is to provide the functionality to be used with a variety of formats such as JSON, XML, YAML, unlike SOAP which is only bound to XML. The REST methods such as GET, DELETE and PUT are idempotent which means any request to change in the server for multiple times with the same request will not affect the server; it will take place only once. On the other hand, POST is not idempotent, making multiple requests will add data to the server for the number or request it has made. Since REST is a stateless protocol, it reduces the redundant code by reusing the components, thereby increasing the performance and reliability. REST API acts as a bridge between frontend and the database. This application Uses GET, POST methods for all the operations dealing with JSON data.

GET:

To retrieve information from the server the GET Method is used, once the requested information is found status of 200 OK is sent confirming the data is found else 404 error messages will be posted if no information is available. Changes will not take place by the GET method on the server.

POST:

The POST method, which will be helpful to add any new data to the server, send the data in a form to add file contents into the server. GET method requests to retrieve data from the server and POST method requests to add the data into the server.

Retrieving HOME Page:

The below Code snippet Appendix 1.0 is written to route the home page for this application which is named as INDEX.HTML. When a user wants to access the URL to this page, the GET method will invoke and load this page on the web browser. This method does not take any parameter.

```
//Display the Index page when user hits the URL in browser
app.get('/', function(req,res){
    res.render('index.html');
});
```

Appendix 1 HOME Page Route.

Fetch Data :

Fetching new data from the web and adding to the Table is done by the GET method that gives functionality for the Refresh Button. Every time when a user hits the Refresh button to get new proxies into the application, a GET method will be triggered and will read all the URLs and then reads the data in the dump and adds new data by the proxy_feed_data_push function that is not in the database already. The Get method Refresh time as a parameter to verify if the last refresh time is less than 10 mins then returns the data will be fed into jqgrid to display. The below code snippet will help you understand the flow.


```

//Reftesh the data based on the click
app.get('/Refresh_Data', function(req,res){
  //Check if there is 10 mins gap
  proxyFeedRetrieve.TimeChecker(function(data){
    var dateDiff = data.recordset[0].DATEDIFF;
    if(dateDiff > 10)
    {
      proxyFeedRetrieve.Proxy_Feed_Data_Push(function(data){
        //After resding the proxy right the inserted time intp the the table
        proxyFeedRetrieve.Refresh_Time_Write(function(data){
          res.send(data);
        });
      });
    }
    else
    {
      var status = 303;
      res.status(200).send((status).toString());
    }
  });
});
});

```

Appendix 2 Fetching New Data.

Loading Data into the Table:

The data is retrieved from the database using the GET method and provided to the jqgrid to display the content on the web browser. The information retrieved will be in JSON format. This method will be invoked as soon as the user lands into the home page. This GET method does not use any parameter and fetches all the data from the database. The below code snippet to demonstrate the data pull request from the database.

```

//get the proxy data from db and populate the Jqgrid table
app.get('/GET_PROXY_DATA', function(req,res){
  proxyFeedRetrieve.Proxy_Feed_Data_Pull(function(data){
    res.send(data);
  });
});

```

Appendix 3 Load data into the table

Export Data :

This application has a functionality to export its data into an EXCEL or CSV file. All the data from the database will be transported into a CSV file using a GET method request made to the URL '/Export_proxy_data'. The appendix 4 code snippet will give an idea of how it is achieved.

```
//Export the data to excel
app.get('/Export_Proxy_Data',function(req,res){
    proxyFeedRetrieve.Export_Proxy_Data(req,res)
});
```

Appendix 4 Export data into CSV file

Display Last Refresh Time :

A GET request is used to achieve this last refresh time that is displayed on the homepage. The GET method is not having any parameter to this request and will execute the function Last_refresh_time—appendix 5, as shown in the below code snippet.

```
//Function to get information about the Last refresh time
app.get('/Last_Refresh_Time', function(req,res){
    proxyFeedRetrieve.Last_Refresh_Time(function(data){
        res.send(data);
    });
});
```

Appendix 5 Last Refresh time

Load Providers Information:

Now comes the second view of the application where the user can see the provider information like base URL form, which the data is being fetched. In order to achieve this, I have made a get request using the URL `/Provider_info` to retrieve information such as base URL and last accessed time for that provider—appendix 6 as shown in the below code snippet.

```
//Get the Providers information
app.get('/Provider_Info',function(req,res){
    proxyFeedRetrieve.Provider_Info(function(data){
        res.send(data);
    });
});
```

Appendix 6 Providers Information

Add New Provider :

In order to add a new provider to the list to fetch the proxies from that provider, I have written a post method that takes a single parameters `Provider_url` which takes a URL of a source as an input and will be added to the database. Once the provider is successfully added into the database, an indicator will be sent using `res.send()` to alter that a new provider is added to the database. Appendix 7 code snippet shown below.

```
//Add a new provider URL to DB
app.post('/ADD_NEW_URL', function(req,res){
    var Provider_URL = req.body.Provider_URL;
    proxyFeedRetrieve.ADD_NEW_URL(Provider_URL,function(data){
        res.send(data);
    });
});
```

Appendix 7 Add New Provider

Reference

- [1] [online] <https://www.tutorialspoint.com>
- [2] [online] <https://perishablepress.com/>
- [3] [online] <https://stackoverflow.com/>
- [4] [online] <https://nodejs.org/en/docs/>

