API Documentation

This document will outline all the API routes and the expected input and output from them.

Data will be returned from the API in JSON format with a relevant HTTP response code.

Here are some links for a refresher on some of these concepts:

[JSON Intro](https://www.w3schools.com/js/js_json_intro.asp)

[HTTP Response Codes](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status)

[HTTP Request Methods](https://www.w3schools.com/tags/ref_httpmethods.asp)

To test API routes a tool such as [Postman](https://www.postman.com/) can be used. I will also publish the real-world results to a Postman API documentation page linked below, this can be used alongside the info in this document.

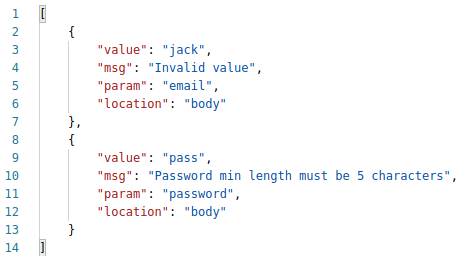
<https://documenter.getpostman.com/view/12745240/TVK8af13>

To check JSON Web Tokens here is a handy tool: <https://www.jsonwebtoken.io/>

**Input validation**

Input validation should be always be performed on the frontend as well as the backend. It is performed on the backend with a plugin called “express-validator” and a brief description of the validation operations being performed are provided in this documentation.

If one or more fields do not pass validation a JSON array will be returned outlining the issue along with a HTTP status code 422 (Unprocessable Entity). Below is the result of validation errors for trying to register a new user with an email value of “jack” and password of “pass”.



Note all items highlighted in this document are subject to change (or still need implementing) as work progresses.

# Demo

Please note because this is a simple GET request you can put this route directly into your browser to see the raw output.

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| GET | https://api-redback-crawler.herokuapp.com/api/demo | ❌ |

Returns:

JSON object of the demo\_table contents from DB

HTTP 200 ok

# User accounts and login

## Register new user

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| POST | https://api-redback-crawler.herokuapp.com/api/user/register | ❌ |

Data to send in JSON format:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| first\_name | String | Not empty, special chars escaped |
| last\_name | String | Not empty, special chars escaped |
| email | String | Must be email, non email special chars removed |
| role | String | Must be one of: user, admin, premiumuser |
| password | String | Minimum length 5 characters, not empty |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | “adduser”: “User created.” |

If user email already exists:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | "adduser": "User already exists." |

Note: the user is not created in this case.

## Login user

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| POST | https://api-redback-crawler.herokuapp.com/api/user/login | ❌ |

Data to send in JSON format:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| email | String | Must be email, non email special chars removed |
| password | String | Minimum length 5 characters, not empty |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | “token”: “*access token string will be here*” |

## Test authentication

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| POST | https://api-redback-crawler.herokuapp.com/api/demo/auth | ✅ |

Data to send in the header:

|  |  |
| --- | --- |
| **Key** | **Value** |
| token | *Access token string returned from login* |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | "auth": **true**,  "role": "user",  "user": "[test@test.com](mailto:test@test.com)",  "message": "User JWT token was verified and access allowed" |

If token has expired or not valid:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 400 | "auth": **false**,  "message": "Token provided is invalid" |

If no token header field present:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 403 | "auth": **false**,  "message": "No token provided" |

The JWT secret is set as an environment variable in heroku and you can access it from there if you wish. Below is what the payload looks like



## Update (change) password

This method allows a logged in user to update their password. It is an authenticated method, meaning a valid JWT must be present. Internally the first and last name as well as email are checked before password is changed. This happens in the backend via the authenticator method so all the front end needs to pass in is the new password (as well as the valid JWT header for the logged in user)

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| PUT | https://api-redback-crawler.herokuapp.com/api/user/updatepass | ✅ |

Data to send in JSON format:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| new\_password | String | Minimum length 5 characters, not empty |

Data to send in the header:

|  |  |
| --- | --- |
| **Key** | **Value** |
| token | *Access token string returned from login* |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | "message": "Password successfully changed" |

If token has expired or not valid:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 400 | "auth": **false**,  "message": "Token provided is invalid" |

If no token header field present:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 403 | "auth": **false**,  "message": "No token provided" |

## Reset Password

This method is not authenticated and resets the user's password to an automatically generated one (aka forgot password) which is emailed to their registered email address.

It requires the user to enter their email, first name and last name. If all match an existing record in the database, the password reset proceeds. Note: I made the names case-insensitive for this one to avoid unnecessary confusion.

Note to team: Make sure the user you are resetting has a real email address or you will never get the new password...

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| POST | https://api-redback-crawler.herokuapp.com/api/user/resetpass | ❌ |

Data to send in JSON format:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| email | String | Must be email, non email special chars removed |
| first\_name | String | Not empty, special chars escaped |
| last\_name | String | Not empty, special chars escaped |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | “resetpassword”: “Password was reset” |

Unsuccessful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 400 | "resetpassword": "User not found" |

## Update User Details

This method updates the details of the logged in user.

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| PUT | https://api-redback-crawler.herokuapp.com/api/user/update | ✅ |

Data to send in JSON format:

You can pass in just one of these fields or any combination you want, just no duplicates or empty fields. If you want to only update the email, then just send the email field with the new email in it.

For Admins they can pass in user\_id field to change any users info, as well as role field. If the admin does pass in a user\_id then it will update the logged in admin instead.

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| email | String | Must be email, non email special chars removed. Optional |
| first\_name | String | Not empty, special chars escaped. Optional |
| last\_name | String | Not empty, special chars escaped. Optional |

Admin Features.

|  |  |  |
| --- | --- | --- |
| user\_id | Int | Must be integer. Optional |
| role | String | Must be one of: user, premiumuser, admin. Optional |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | "message": "Updated" |

If new email is already in the system:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 400 | "message": ”Email not unique in system" |

General failure returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 400 | "message": ”error" |

## Admin Get all users

This method can only be accessed by logged in admin users. It has three different return types to make it easier to integrate to frontend.

**Method 1 explanation:** Get request with no parameters in query string returns a maximum of 100 records starting at an offset of zero.

**Method 2 explanation:** Get request with limit and offset query parameters. This is useful if there are a large amount of results and you for example you want to display 10 per set. The first set would start with a limit of 10, offset of 0, the second set a limit of 10, and offset of 10 and so on. (If you use this I can make a get size route if needed, so you know how many total records there are)

**Method 3 explanation:** Specify the role type you want to get back. For example, on premium users. This can be paired with either of the above methods/approaches. (?limit=10&offset=0&rolesort=admin)

**Method 1**

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| GET | https://api-redback-crawler.herokuapp.com/api/admin/users | ✅ |

**Method 2**

|  |  |  |
| --- | --- | --- |
| GET | api/admin/users?limit=10&offset=0 | ✅ |

Data to send in Query String

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| limit | Integer | Optional, Must be integer |
| offset | Integer | Optional, Must be integer |

**Method 3**

|  |  |  |
| --- | --- | --- |
| GET | api/admin/users?rolesort=admin | ✅ |

Data to send in Query String

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| rolesort | Integer | Must be one of: admin, user, premiumuser |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON Object of results** |
| 200 | {  "id": 4,  "first\_name": "Ned",  "last\_name": "Kelly",  "email": "[ned@bushrangers.com](mailto:ned@bushrangers.com)",  "role": "admin",  "created": "2020-10-01T02:12:20.825Z"  }, |

If there are no results it will return an empty object.

If incorrect user role tries to use route it will return: 403, message: Access denied Admin only

## Admin Delete User

This method allows a logged in account with **Admin** role to delete any user by their ID.

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| DELETE | https://api-redback-crawler.herokuapp.com/api/admin/users/ | ✅ |

Send the user ID to delete as a parameter.

Example: To delete a user with an ID of 12

/api/admin/users/12

|  |  |
| --- | --- |
| **Value data type** | **Input validation** |
| Integer | Must be integer |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | {  "message": "user deleted"  } |

Unsuccessful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 404 | {  "message": ”no user found by that ID"  } |

# EBAY Crawler

## Add Link

This route enables a logged in user to add an ebay link (item) to their account.

If the notify\_change is set as TRUE, then the price\_increase and change\_amount must have values.

**Example of notify\_change logic:**

Ebay item “PC Mouse” initial price $100

User wants to get notified when the price increase by $20. (price on ebay $120 or more)

notify\_change = TRUE, Price\_increase = TRUE, Change\_amount=$20 when adding the product.

User wants to get notified when the price decreases by $20 (price on ebay $80 or less)

notify\_change = TRUE, Price\_increase = FALSE, Change\_amount=$20 when adding the product.

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| POST | https://api-redback-crawler.herokuapp.com/api/user/ebaylink | ✅ |

Data to send in JSON format:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| link | String | Run through a strict REGEX. Must start with: <https://www.ebay.com.au/itm/> |
| update\_interval | Int | Represents hours between updates(scrapes)  Must one of 3, 6, 12 |
| notify\_change | Boolean | If the user wants to be updated via email about price changes. (true or false) |
| price\_increase | Boolean | True for increase False for decrease |
| change\_amount | Float | Must be a valid float. |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON Object of results** |
| 200 | {  "name": "SM Pro Yamaha WR250F WR450F 02-19 Black Rim Blue Hubs Platinum Wheel Set",  "price": "999.95",  "isAuction": **false**,  "user": 1  } |

Note: isAuction is not utilized.

## Get users saved Ebay links

This route returns all of the logged in users saved Ebay links

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| GET | https://api-redback-crawler.herokuapp.com/api/user/ebaylink | ✅ |

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON Object of results** |
| 200 | {  "id": 1,  "user\_id": 1,  "link": "<https://www.ebay.com.au/itm/RC12YC-Spark-Plug/192750321748?_trkparms=aid%3D777001%26algo%3DDISCO.FEED%26ao%3D1%26asc%3D20200211172457%26meid%3De50d4f8f89974aee8771a48ba98fdc76%26pid%3D101213%26rk%3D1%26rkt%3D1%26mehot%3Dlo%26itm%3D192750321748%26pmt%3D0%26noa%3D1%26pg%3D2380057%26algv%3DRecommendingSearch%26brand%3D&_trksid=p2380057.c101213.m46344&_trkparms=pageci%3A675a69b6-1289-11eb-8ee6-3e9d43c584ab%7Cparentrq%3A442e61f61750a6e5744bac3afff9676c%7Ciid%3A2>",  "link\_description": "RC12YC Spark Plug",  "last\_crawled": "2020-10-20T17:05:24.441Z",  "link\_valid": **true**,  "initial\_price": "9.9",  "current\_price": "9.9",  "crawl\_interval": 3  }, |

Note: id represents the products ID.

## Get a links price history

This will get all the price entries for the link (every time it has been updated regardless of if the price changed)

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| GET | https://api-redback-crawler.herokuapp.com/api/user/ebaylink/history | ✅ |

Data to send in Query String

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| link\_id | Integer | Must be integer |

Example: /api/user/ebaylink/history?link\_id=2

Returns 1 or more entries for the links price history

|  |  |
| --- | --- |
| **HTTP Status Code** | **Raw Value** |
| 200 | {  "id": 2,  "link\_id": 2,  "time\_crawled": "2020-10-20T17:08:22.814Z",  "price": "999.95"  },  "id": 3,  "link\_id": 2,  "time\_crawled": "2020-10-20T18:02:22.997Z",  "price": "999.95"  }  ] |

Note: The id field is not used. It represents the links id in the history table. This could be cleaned up if I create a db method for this.

## Update Link crawl interval

This allows a user to update the interval the link is crawled. Every 3, 6 or 12 hours.

Users can only update links that belong to their account.

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| PUT | https://api-redback-crawler.herokuapp.com/api/user/ebaylink/update | ✅ |

Data to send in JSON body:

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| link\_id | Integer | Must be integer |
| Update\_interval | Integer | Represents hours between updates(scrapes)  Must one of 3, 6, 12 |

Returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **Raw Value** |
| 200 | "message": "Link updated" |

On failure:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 400 | "message": ”Error updating link" |

## Delete Link

This route allows a logged in user to delete any links they have associated with their account.

Note: there is logic implemented in the backend that only allows the user access to links that belong to them. Passing in a link\_id for a link associated with a different user will not work.

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| DELETE | https://api-redback-crawler.herokuapp.com/api/user/ebaylink/ | ✅ |

Send the link ID as a parameter.

Example: To delete a link with a ID of 55

/api/user/ebaylink/55

|  |  |
| --- | --- |
| **Value data type** | **Input validation** |
| Integer | Must be integer |

Successful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 200 | {  "message": "Link deleted"  } |

Unsuccessful operation returns:

|  |  |
| --- | --- |
| **HTTP Status Code** | **JSON** |
| 404 | {  "message": Link not found for current user"  } |

# API Tools

## Currency Conversion

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| GET | https://api-redback-crawler.herokuapp.com/api/currency | ❌ |

Data to send in Query String

|  |  |  |
| --- | --- | --- |
| **Name** | **Value data type** | **Input validation** |
| amount | Float | Must be a number |
| from | String | Must match one of the supported currencies codes |
| to | String | Must match one of the supported currencies codes |

Supported currency codes: AUD, USD, NZD, GBP (Note we can easily add more later on)

Example query to convert $120.45 AUD to USD:

/api/currency?amount=120.45&from=AUD&to=USD

|  |  |
| --- | --- |
| **HTTP Status Code** | **Raw Value** |
| 200 | 85.57 |

## Currency History

|  |  |  |
| --- | --- | --- |
| **Method** | **Route** | **Auth Token** |
| GET | https://api-redback-crawler.herokuapp.com/api/currency/history | ❌ |

Returns the contents of the ibanHistory table

|  |  |
| --- | --- |
| **HTTP Status Code** | **Raw Value** |
| 200 | {  "id": 1,  "code": "AUD",  "rate": "1.6563",  "last\_updated": "2020-10-31T21:52:27.503Z"  },  {  "id": 2,  "code": "USD",  "rate": "1.1698",  "last\_updated": "2020-10-31T21:52:27.503Z"  },  {  "id": 3,  "code": "GBP",  "rate": "0.90208",  "last\_updated": "2020-10-31T21:52:27.503Z"  },  etc |

Internal API feature documentation

## Currency Scraper

A web scraper (ibanCrawler.js internally) that pulls data from <https://www.iban.com/exchange-rates> that pulls the currency code and associated value from the currency table on the site.

The entire table of content is available to the scraper however at this point in time only five (AUD, NZD, USD, GBP, CNY) are currently be utilized. Once this data is scraped it is passed into a database function that updates the database with the new data.

Currently this scraper is scheduled to run every three hours. The external iBan rates site updates once per day.