Back-end

For API I used Laravel ^9.

For API documentation I used Scribe ^3

API routes

Route::post('login',[UserController::class,'login']);

Route::group(['middleware' => ['auth:api']], function ()

{

    Route::get('currency', [CurrencyController::class, 'currency']);

    Route::post('convert', [CurrencyController::class, 'convert']);

    Route::post('verify', [TokenController::class, 'verify']);

    Route::post('refresh', [TokenController::class, 'refresh']);

});

For testing reasons I have create a subdomain and upload the API

I have test the API using Postman

http://homechallenge.databees.uk

API Documentation

<http://homechallenge.databees.uk/docs/>

To do:

* Creating random rates for currency, at this moment I have create a table with currencies and rate. The main currency is GBP with rate 1.00
* Refresh token, for testing reasons I have create a custom token and controller, but here is still work to be done. For small projects we can use sanctum or for bigger projects we can use passport. The refesh token has to have a scope like view-dashbord(to be implemented)

# Front-end

In store/api/api.js we have the API connection,here we setting up the token in header and all the logic for authorization like refresh token

In store/feature/userSlice.js setting the user in global state

In store/feature/login.js endpoints for login

Store/index.js the place where we configure global state and add in reducers user – store/features/userSlice.js

In hooks we have three custom hooks useLogin and useToken and useCurrencies

For API I have used RTK – Redux Toolkit

To do:

* Implementing the scope for refresh token and changing the route logic when I have 403 from API , like token invalid, do not see the page any more, or come back to login
* Tests