

Pineapple

Software Document

Table of Contents

[Code Documentation](#)

[Documentation Format](#)

[Controllers](#)

[Authentication \(Laravel Generated\)](#)

[ForgotPasswordController](#)

[LoginController](#)

[RegisterController](#)

[ResetPasswordController](#)

[General](#)

[AdminDashboardController](#)

[AllOrdinariesController](#)

[BrokerController \(Unutilised\)](#)

[CompaniesController \(Unutilised\)](#)

[CompanyController](#)

[CompanyHistoryController \(Unutilised\)](#)

[DatabaseStockCodes](#)

[FriendController](#)

[HomeController](#)

[LeaderboardController](#)

[MessagesController](#)

[ShowStock](#)

[TopASXController](#)

[TradeAccountController](#)

[TransactionController](#)

[UserAccountController](#)

[Commands](#)

[getCompanies](#)

[getCompany](#)

[stockChangeCheck](#)

[updateAllHistory](#)

[updateTopLists](#)

[Command Scheduling](#)

[Kernel](#)

[Middleware](#)

[Admin](#)

[Helpers](#)

[CurrencyConverter](#)

[CurrentTableScraper](#)

[getTotalGrowth](#)
[CurrentCompanyStockInformation](#)

[Routes](#)

[api](#)
[web](#)

[Views](#)

[Authentication](#)

[email](#)
[reset](#)
[login](#)
[register](#)

[Emails](#)

[accountDeleted](#)
[adminAlert](#)
[stockchanged](#)

[Layouts](#)

[app](#)
[background](#)
[body](#)
[footer](#)
[header](#)
[header-404](#)
[header-navbar](#)
[navbar](#)

[General](#)

[404](#)
[admin-dashboard](#)
[contact](#)
[dashboard](#)
[friends](#)
[home](#)
[leaderboard](#)
[messages](#)
[privacy-and-policy](#)
[profile](#)
[profiles](#)
[stock](#)
[stockinfo \(Unutilised\)](#)
[trade-account](#)
[Useraccount \(Unutilised\)](#)
[welcome \(Unutilised\)](#)

[Javascript](#)

[JQuery](#)

[Three.JS](#)

[Chart.JS](#)

[External API Documentation](#)

[Stock Data Retrieval](#)

Code Documentation

Documentation Format

The explanation for each file, takes the following format:

Filename (excluding file extension)

[File Location from the root of the project]

Brief Description of the File

File Methods

Controllers

Authentication (Laravel Generated)

ForgotPasswordController

[~/website/app/Http/Controllers/Auth/ForgotPasswordController.php]

This controller is responsible for handling password reset emails and includes a trait which assists in sending these notifications from your application to your users. Feel free to explore this trait.

public __construct()

LoginController

[~/website/app/Http/Controllers/Auth/LoginController.php]

This controller handles authenticating users for the application and redirecting them to your home screen. The controller uses a trait to conveniently provide its functionality to your applications.

public __construct()

RegisterController

[~/website/app/Http/Controllers/Auth/RegisterController.php]

This controller handles the registration of new users as well as their validation and creation. By default this controller uses a trait to provide this functionality without requiring any additional code.

```
public __construct()  
protected validator(array $data)  
protected create(array $data)
```

ResetPasswordController

[~/website/app/Http/Controllers/Auth/ResetPasswordController.php]

This controller is responsible for handling password reset requests and uses a simple trait to include this behavior. You're free to explore this trait and override any methods you wish to tweak.

```
public __construct()
```

General

AdminDashboardController

[~/website/app/Http/Controllers/AdminDashboardController.php]

This controller is responsible for a variety of functionality to enable Admins to have regulate the game.

```
private getUsersCount()  
private getAllUsers()  
public view()  
public deleteUser(Request $request)  
public modifyRole(Request $request)  
Public emailUsers(Request $request)
```

AllOrdinariesController

[~/website/app/Http/Controllers/AllOrdinariesController.php]

This controller is responsible for the retrieval of the price of the All Ordinaries.

```
public getCurrentAllOrdinaries()  
public getPrice($dom)
```

BrokerController (Unutilised)

[~/website/app/Http/Controllers/BrokerController.php]

This controller was to be responsible as the in-between broker to handle the buying and selling of stocks in a particular fashion.

```
public buy()  
public sell()
```

CompaniesController (Unutilised)

[~/website/app/Http/Controllers/CompaniesController.php]

This controller is responsible for the retrieval of all companies on the ASX.

```
public index()  
public getAllListedCompanies()
```

CompanyController

[~/website/app/Http/Controllers/CompanyController.php]

This controller is responsible for retrieving current stock listing information for a selected company.

```
public currentDetails($code = null)  
public getPrice($dom)  
public fatalError($message = "Could not find ASX item", $status = 404)
```

CompanyHistoryController (Unutilised)

[~/website/app/Http/Controllers/CompanyHistoryController.php]

This controller was to be responsible as the in-between broker to handle the buying and selling of stocks in a particular fashion.

```
public buy()  
public sell()
```

DatabaseStockCodes

[~/website/app/Http/Controllers/DatabaseStockCodes.php]

This controller gets all stock listings from the database, and responds with their name, symbol and market.

```
public getAllStockCodesAndCompanyNames()
```

FriendController

[~/website/app/Http/Controllers/FriendController.php]

This controller handles the sending and accepting of friend requests, as well as giving the appropriate view.

```
public sendFriendRequest(Request $request)  
public acceptFriendRequest(Request $request)  
public view(Request $request)
```

HomeController

[~/website/app/Http/Controllers/HomeController.php]

This controller routes the dashboard back to a user.

```
public __construct()  
public index()
```

LeaderboardController

[~/website/app/Http/Controllers/LeaderboardController.php]

This controller recalculates the leaderboard and returns the view associated.

```
public index()
```

MessagesController

[~/website/app/Http/Controllers/MessagesController.php]

This controller is responsible for the handling of messages whether it is being sent or received, handling the sending of money and returning the message view.

```
public view(Request $request, $id = null, $error = null)
public sendMessage(Request $request, $id = null)
public saveMessage($message, $id = null)
public first(Request $request, $id = null)
public acceptMoney(Request $request, $id = null)
```

ShowStock

[~/website/app/Http/Controllers/ShowStock.php]

This controller is responsible for getting the relevant stock and returning a stock page with its associated data.

```
public __invoke($code)
```

TopASXController

[~/website/app/Http/Controllers/TopASXController.php]

This controller is gets the top 20, companies by market capitalisation, and returns the response of those companies sorted by position.

```
public getList($count = 20)
public fatalError($message = "Could not fine ASX item", $status = 404)
```

TradeAccountController

[~/website/app/Http/Controllers/TradeAccountController.php]

This controller is responsible for the handling of the creation, editing and viewing of Trade Accounts.

```
public create(Request $request, $id = null, $error = null)
public edit(Request $request, $id = null)
public saveMessage($message, $id = null)
public first(Request $request, $id = null)
public acceptMoney(Request $request, $id = null)
```

TransactionController

[~/website/app/Http/Controllers/TransactionController.php]

This controller is responsible for the buying and selling of shares from the available stocks, as well as retrieving transactions within a certain period.

```
public apiAddBuyTransaction(Request $request)
public apiAddSellTransaction(Request $request)
public getSingleStockQuality(Request $request)
Public getTransactionsInDateRange(Request $request)
```

UserAccountController

[~/website/app/Http/Controllers/UserAccountController.php]

This controller is responsible for the buying and selling of shares from the available stocks, as well as retrieving transactions within a certain period.

```
public edit(Request $request)
public apiEdit(Request $request)
public searchUserWithQuery(Request $request, $query = null)
```

Commands

getCompanies

[~/website/app/Console/Commands/getCompanies.php]

Command: php artisan company:getAll

This command is responsible for filling our database with all of the possible stocks in the stock market.

```
public __construct()
public handle()
public getAllListedCompanies($address, $firstLines)
```

getCompany

[~/website/app/Console/Commands/getCompany.php]

Command: php artisan company:get {stock code}

This command is responsible for retrieving the current data for a particular stock and storing the current data in the database.

```
public __construct()
public handle()
public historyHour($code, $market)
public hourly($code = null, $market)
public fatalError()
```

stockChangeCheck

[~/website/app/Console/Commands/stockChangeCheck.php]

Command: php artisan stockChange:check

This command is responsible for checking whether any user has a significant pre-defined difference in value for stocks they own. If so, this command will email all relevant users.

```
public __construct()
public handle()
```

updateAllHistory

[~/website/app/Console/Commands/updateAllHistory.php]

Command: php artisan company:updateAllHistory

This command is responsible for updating all possible stocks currently stored in the database.

```
public __construct()  
public handle()
```

updateTopLists

[~/website/app/Console/Commands/updateAllHistory.php]

Command: php artisan company:updateTopLists

This command is responsible for updating the current top lists of the ASX to be stored in the database.

```
public __construct()  
public handle()  
public getList($count = 20)
```

Command Scheduling

Kernel

[~/website/app/Console/Kernel.php]

This file is responsible for storing the current commands usable by our application, as well as allowing commands to be scheduled with Laravel that will be run through Cron.

```
protected schedule()  
protected commands()
```

Middleware

Admin

[~/website/app/Http/Middleware/Admin.php]

This middleware is responsible for redirecting a user if they are an admin to the appropriate webpage.

```
public handle()
```

Helpers

CurrencyConverter

[~/website/app/Helpers/CurrencyConverter.php]

This helper was designed to easily allow us to convert USD to AUD within our application.

```
public USDtoAUD($amount = null)
```

CurrentTableScraper

[~/website/app/Helpers/CurrentTableScraper.php]

This helper was written to allow the easy extraction of data from a given DOM object, returning in the form of an array.

```
public extractCurrentTableRowFromDOM(&$dom, &$dataArray, $heading = null, $reactID = null)
```

getTotalGrowth

[~/website/app/Helpers/getTotalGrowth.php]

This helper was written to calculate the current total growth of all of that user's owned stocks.

```
public getTotalGrowth($id = null)
```

CurrentCompanyStockInformation

[~/website/app/Helpers/CurrentCompanyStockInformation.php]

This helper returns the current listing information for a particular stock.

```
public currentDetails($code = null, $market)
public getPrice($dom)
private scrapeTableRow(&$dataArray, $table)
public fatalError($message = "Could not find ASX item", $status = 404)
```

Routes

api

[~/website/app/routes/api.php]

This file contains all the API routes for our application.

web

[~/website/app/routes/web.php]

This file contains all of the web routes for our application.

Views

Authentication

email

[~/website/resources/views/auth/passwords/email.blade.php]

This is the template for sending an email to reset your password.

reset

[~/website/resources/views/auth/passwords/reset.blade.php]

This is the template for re-entry of your password.

login

[~/website/resources/views/auth/login.blade.php]

This is the template for logging into our Application.

register

[~/website/resources/views/auth/register.blade.php]

This is the template for registering for our Application.

Emails

accountDeleted

[~/website/resources/views/emails/accountdeleted.blade.php]

This is the template email for an account deletion email.

adminAlert

[~/website/resources/views/emails/adminalert.blade.php]

This is the template email for an admin sending a message to all users of our application.

stockchanged

[~/website/resources/views/emails/stockchanged.blade.php]

This is the template email for the alert of a change in stocks.

Layouts

app

[~/website/resources/views/layouts/app.blade.php]

Default Laravel Homepage.

background

[~/website/resources/views/layouts/background.blade.php]

This is the background of the homepage.

body

[~/website/resources/views/layouts/body.blade.php]

This is the body of content for the homepage.

footer

[~/website/resources/views/layouts/footer.blade.php]

This is the footer of our application.

header

[~/website/resources/views/layouts/header.blade.php]

This contains the head tag.

header-404

[~/website/resources/views/layouts/header-404.blade.php]

This is the part of the content of the 404 page.

header-navbar

[~/website/resources/views/layouts/header-navbar.blade.php]

This is an alternative navbar that is used within our application.

navbar

[~/website/resources/views/layouts/footer.blade.php]

This is the main navbar of our application.

General

404

[~/website/resources/views/404.blade.php]

This is the main 404 Error Page of our application.

admin-dashboard

[~/website/resources/views/admin-dashboard.blade.php]

This is the body of content for our administration dashboard for moderating users.

contact

[~/website/resources/views/contact.blade.php]

This is the body of content for contact us page, with our details and location.

dashboard

[~/website/resources/views/dashboard.blade.php]

This is the body of content for the user dashboard.

friends

[~/website/resources/views/friends.blade.php]

This is the listing of this user's friends.

home

[~/website/resources/views/home.blade.php]

This is the template that combines views to display the homepage.

leaderboard

[~/website/resources/views/leaderboard.blade.php]

This template displays the leaderboard of all users.

messages

[~/website/resources/views/messages.blade.php]

This template lists a user's messages, from friends, to read and send messages.

privacy-and-policy

[~/website/resources/views/privacy-and-policy.blade.php]

This is our applications privacy policy.

profile

[~/website/resources/views/profile.blade.php]

This template is for displaying a user's information.

profiles

[~/website/resources/views/profiles.blade.php]

This is the listing of all profiles in our application.

stock

[~/website/resources/views/stock.blade.php]

This template is for the displaying of a stock page with it's relevant data and graph, as well as having front-end buy and sell functionality.

stockinfo (Unutilised)

[~/website/resources/views/stockinfo.blade.php]

This was to test the look and feel of stock information.

trade-account

[~/website/resources/views/trade-account.blade.php]

This template displays the information for a users' trade account, along with it's transactions.

Useraccount (Unutilised)

[~/website/resources/views/Useraccount.blade.php]

This was a template to design/template our front-end design for a user's account.

welcome (Unutilised)

[~/website/resources/views/welcome.blade.php]

This is an unused welcome page.

Javascript

JQuery

JQuery is utilised for by a variety of our third party javascript libraries (Three.JS, Bootstrap, Chart.JS). It is also used for a lot of our form post requests, due to the complexity of some of these requests.

Three.JS

On our homepage (layouts/background.blade.php), we utilise Three.JS to give our frontpage the colour bursting look that it only Three.JS can achieve. On every page load, it is unique.

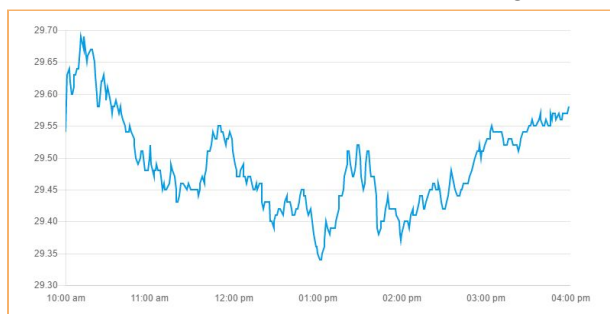
Below is an example.



Chart.JS

On our stock page (stock.blade.php), we utilise Chart.JS in order to visualise the data for each stock which we store in the database.

Below is an example of a stock market graph, from ANZ.



External API Documentation

Stock Data Retrieval

One of the core features of any stock market application, is it's ability to get the most recent data for each possible stock. In our implementation, we utilise the data retrieved from the URL:

`http://chartapi.finance.yahoo.com/instrument/1.0/{stock}/chartdata?type=quote;range=1d/json`

`{stock}` = Stock Code

If it's an Australian Stock, the code is `{stock}.AX`

Return data = CSV File

***** It's important to know that if Yahoo pulls down this closed API, our application will be strongly affected, meaning that a new API or way of retrieving stock data will need to be found *****