

# Technical Documentation - Caramel Capital Trading Engine

|   |          |
|---|----------|
| <b>Database Settings</b>                        | <b>2</b> |
| Database Connection                             | 2        |
| <b>Database Tables</b>                          | <b>2</b> |
| users   | 2        |
| deposits  | 3        |
| mfeurcandlestick                                | 3        |
| open_orders                                     | 4        |
| pgeur_ask && pgeur_bid                          | 4        |
| trades  | 5        |
| withdrawals                                     | 5        |
| <b>Trading Settings</b>                         | <b>6</b> |
| How to add new trading pairs                    | 6        |
| Create a new virtual wallet                     | 6        |
| Create a webpage for the new trading pair       | 6        |
| Create database tables for bids and asks        | 6        |
| Get data from database for the new trading pair | 7        |
| Setup the webpage with the correct data         | 7        |
| <b>Admin Management</b>                         | <b>9</b> |
| Admin Dashboard                                 | 9        |
| Credit Deposits                                 | 9        |
| Credit Bonus                                    | 9        |
| Confirm Withdrawal                              | 9        |
| Mint New Royalties                              | 9        |
| Freeze account                                  | 10       |

# Database Settings

## Database Connection

We connect to the database through the database.php file.

We define the server, username, password and database name.

You need to put your own credentials in order to connect.

```
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'your_username');
define('DB_PASSWORD', 'your_password');
define('DB_DATABASE', 'your_database');
```

## Database Tables

### users

We store the basic information of the users.

- Unique ID
- Username
- Email
- Encrypted password (We can't see the user's password)
- Balances
  - EUR
  - Pg (Royalties)
- Role
  - Admin
  - Investor
- Account freezed or unfreezed

| id | username | email             | password                         | eur          | pg            | role     | freeze |
|----|----------|-------------------|----------------------------------|--------------|---------------|----------|--------|
| 1  | test1    | test1@gmail.com   | 5a105e8b9d40e1329780d62ea2265d8a | 41092.6686   | 3932.5000     | admin    | 0      |
| 2  | test2    | test2@hotmail.com | ad0234829205b9033196ba818f7a872b | 3535000.2000 | 45278257.0000 | investor | 0      |
| 3  | test4    | test4@gmail.com   | 81dc9bdb52d04dc20036dbd8313ed055 | 537.0000     | 0.0000        | investor | 0      |
| 4  | pablo    | pablo@gmail.com   | 7e4b64eb65e34fdad79e623c44abd94  | 0.0000       | 0.0000        | investor | 0      |
| 5  | test5    | test5@gmail.com   | 202cb962ac59075b964b07152d234b70 | 1000.0000    | 1000.0000     | investor | 0      |

## deposits

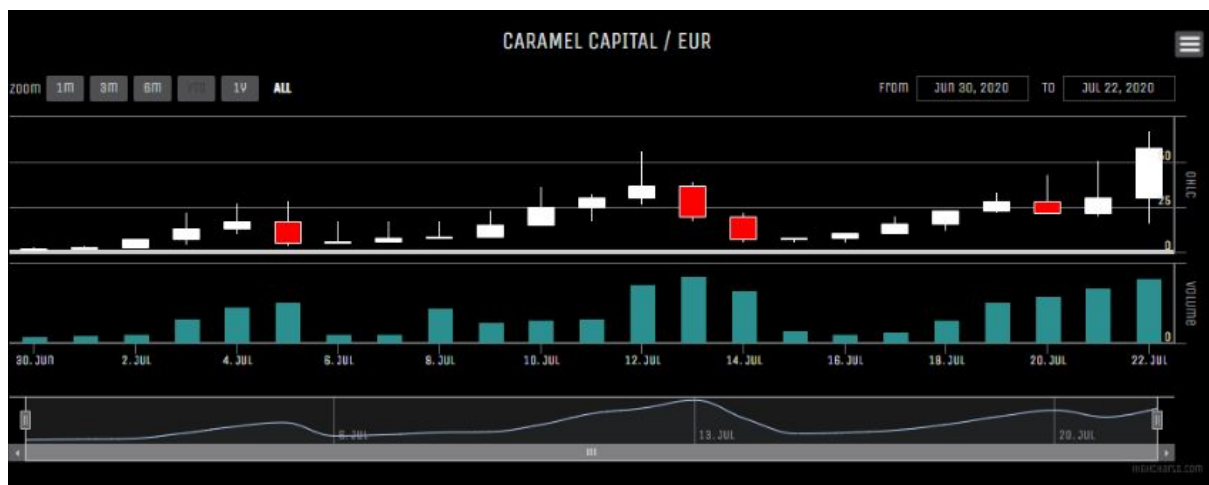
We store the deposit history of the users.

- ID
- Username
- Amount in EUR
- Reference (Needed to match the user's bank account with their virtual wallet)
- Status
  - Confirmed
  - Rejected
  - Pending
- Date

| id | username | amount_EUR | reference | status    | date                |
|----|----------|------------|-----------|-----------|---------------------|
| 1  | test1    | 1500.00    | NULL      | confirmed | 2020-02-23 16:37:54 |
| 2  | test1    | 25000.00   | NULL      | confirmed | 2020-02-23 16:39:20 |
| 3  | test1    | 7500.00    | NULL      | confirmed | 2020-02-23 16:40:48 |
| 4  | test2    | 3500.00    | NULL      | confirmed | 2020-02-23 16:41:17 |
| 5  | test2    | 72.00      | NULL      | rejected  | 2020-02-23 16:43:57 |

## mfeurcandlestick

We use that table to display the candlestick graph showing the history prices in a visual form.



In this case, we show prices, volume and date in daily candles (Each candle represents one day).

We have a different table for each trading pair chart.

| open  | close | high  | low   | volume  | date ▾ 1   | timestamp  |
|-------|-------|-------|-------|---------|------------|------------|
| 54.87 | 52.37 | 54.87 | 52.37 | 882.857 | 2020-08-17 | 1597664578 |
| 60    | 54.87 | 60    | 54.87 | 3345.4  | 2020-08-16 | 1597607884 |
| 30    | 54.87 | 67    | 15.7  | 103307  | 2020-07-22 | 1595432847 |
| 22    | 30    | 50.87 | 20    | 70000   | 2020-07-21 | 1595346447 |
| 28    | 22    | 42.72 | 21.77 | 60000   | 2020-07-20 | 1595260047 |

## open\_orders

We store all the open orders in every trading pair from every user.

This table is compatible with multiple trading pairs, the program will just update the "ticker" variable as seen in the screenshot example.

| id | price | amount_RP | date ▲ 1            | username | ticker | order_type | order_id |
|----|-------|-----------|---------------------|----------|--------|------------|----------|
| 83 | 63.00 | 10.0000   | 2020-07-16 17:23:50 | test1    | pgeur  | sell       | 67       |
| 84 | 62.50 | 15.0000   | 2020-07-16 17:24:05 | test1    | cceur  | sell       | 68       |
| 90 | 69.00 | 25.0000   | 2020-07-16 17:25:45 | test1    | rr eur | sell       | 70       |
| 91 | 67.25 | 7.0000    | 2020-07-16 17:26:20 | test1    | pgeur  | sell       | 71       |
| 92 | 45.00 | 37.0000   | 2020-07-17 13:18:42 | test1    | pgeur  | buy        | 114      |

## pgeur\_ask && pgeur\_bid

We have a different bid and ask tables for each pair. We do it for simplicity.

For example, if we have 3 trading pairs we would create the following table distribution:

- tp1 (Trading pair 1)
  - tp1\_bid
  - tp1\_ask
- tp2 (Trading pair 2)
  - tp2\_bid
  - tp2\_ask
- tp3 (Trading pair 3)
  - tp3\_bid
  - tp3\_ask

Each table has the id, username, price, amount (in royalties) and date of order creation.

| id | username | price | amount_RP | date                |
|----|----------|-------|-----------|---------------------|
| 67 | test1    | 63.00 | 10.0000   | 2020-07-16 19:24:05 |
| 68 | test1    | 62.50 | 15.0000   | 2020-07-16 19:24:15 |
| 70 | test1    | 69.00 | 25.0000   | 2020-07-16 19:26:20 |
| 71 | test1    | 67.25 | 7.0000    | 2020-07-16 19:26:33 |
| 72 | test1    | 65.48 | 38.0000   | 2020-07-17 15:21:43 |

## trades

We use that table to display last trades visually.



| Price | Amount | Value   | Date     |
|-------|--------|---------|----------|
| 57.99 | 1      | 57.99€  | 13:52:22 |
| 58    | 5      | 290€    | 13:52:22 |
| 57.99 | 1      | 57.99€  | 13:52:22 |
| 58    | 5      | 290€    | 13:52:22 |
| 57.12 | 5      | 285.6€  | 13:52:06 |
| 57.12 | 5      | 285.6€  | 13:52:06 |
| 57.99 | 5      | 289.95€ | 13:51:21 |
| 57.99 | 5      | 289.95€ | 13:51:21 |
| 57.57 | 2      | 115.14€ | 13:50:51 |
| 56.78 | 3      | 170.34€ | 13:50:51 |

We store id, username, ticker, price, amount (in royalties), date of order creation and type (buy, sell).

| id  | username | ticker | price   | amount_RP | date                | 1 | type |
|-----|----------|--------|---------|-----------|---------------------|---|------|
| 196 | test3    | pgeur  | 52.3700 | 5.0000    | 2020-08-17 11:42:58 |   | sell |
| 195 | test1    | pgeur  | 52.3700 | 5.0000    | 2020-08-17 11:42:58 |   | buy  |
| 193 | test1    | pgeur  | 54.8700 | 3.2728    | 2020-08-17 11:42:58 |   | buy  |
| 194 | test3    | pgeur  | 54.8700 | 3.2728    | 2020-08-17 11:42:58 |   | sell |
| 191 | test1    | pgeur  | 54.8700 | 5.0000    | 2020-08-16 19:58:46 |   | buy  |

## withdrawals

Not implemented for this demo.

# Trading Settings

## How to add new trading pairs

### Create a new virtual wallet

First of all, if we add a new currency or asset, we need to add a wallet for it in the “users” table in the database.

If we list a new trading pair between already existing assets, we can skip that.

```
ALTER TABLE users
ADD new_asset float(18);
```

### Create a webpage for the new trading pair

Then, we need to create the webpage for that new trading pair. So we go to `invest/equity/` and create a new php file with the name of the trading pair. For example: `eurusd.php`

We copy and paste the code from `market-pro.php` and it should now be working, but it will be showing the old trading pair so we need to update it.

At this moment, it displays the data of `cceur`, and we want to change it to `eurusd`.

### Create database tables for bids and asks

We need to create two new tables for bid and ask, named “`bids_eurusd`” and “`asks_eurusd`”.

```
CREATE TABLE bids_eurusd(
    ID int(11) PRIMARY KEY UNIQUE NOT NULL AUTO_INCREMENT,
    username varchar(30) NOT NULL,
    price decimal(4,2) NOT NULL,
    amount_RP decimal(9,2) NOT NULL,
    date datetime
);
```

You will need to adjust the values depending on your needs.

## Get data from database for the new trading pair

We need to get data for that specific pair from database, so we go to

`invest/includes/market_data/` and create the files `orderbook_eurusd.php` and `last_trades_eurusd.php`

We just need to change the tables from which we gather data from. In order to do that, we change the table names inside the files.

It was set like that by default:

```
//We get bids from database
$sql = "SELECT * FROM secondary_market_pgeur_bid ORDER BY price DESC
LIMIT 10";
$result = $conn->query($sql);
```

So we just change it like that

```
//We get bids from database
$sql = "SELECT * FROM bids_eurusd ORDER BY price DESC LIMIT 10";
$result = $conn->query($sql);

//We get asks from database
$sql = "SELECT * FROM asks_eurusd ORDER BY price ASC LIMIT 10";
$result = $conn->query($sql);
```

We then go to `invest/equity/charts` and do the same: create `eurusd_chart.html` and `eurusd_market_depth.php`

The next step is to go to `invest/equity` and do it one more time: create the files `datachart_candlestick_eurusd.php` and `datachart_marketdepth_eurusd.php`

In all the new files created you have to update the database sql queries from the old tables to the new ones as explained before.

## Setup the webpage with the correct data

Now that we are gathering data from the database, we just need to reflect it correctly in the live website.

For that, we open `eurusd.php` inside `invest/equity/` and modify all the queries and includes so they no longer request the old files but the new ones that you just created.

For example, in this file we include royalty\_chart.html and orderbook\_cceur.php, but we need eurUSD\_chart.html and orderbook\_eurUSD.php.

```
<div class="card candlestick_graph">
  <?php
    //Display candlestick chart
    include 'charts/royalty_chart.html';
  ?>
</div>
<div class="card orderbook">
  <div id="orderbook">
    <?php include_once'../includes/market_data/orderbook_cceur.php'; ?>
  </div>
</div>
```

So we update it to request the new data:

```
<div class="card candlestick_graph">
  <?php
    //Display candlestick chart
    include 'charts/eurUSD_chart.html';
  ?>
</div>
<div class="card orderbook">
  <div id="orderbook">
    <?php include_once'../includes/market_data/orderbook_eurUSD.php'; ?>
  </div>
</div>
```

You have to do this to all the includes and eventually you will have a new trading pair!  
It will be hosted in [www.yourwebsite.com/invest/equity/eurUSD.php](http://www.yourwebsite.com/invest/equity/eurUSD.php)



# Admin Management

## Admin Dashboard

This area is restricted, it can only be accessed by admins.

Any user or investor that tries to go here will be redirected out of this page.

Admin Panel

WARNING!  
This is a restricted area! Extreme caution, actions here could lead to unwanted problems!

Credit Deposit

Use this only to credit deposits to investors.

ID

1423

Amount

2500,00€

Reference

Deposit

Change Status

Confirmed

Beneficiary

user1

Credit Deposit

Pending Deposits

| ID   | Username | Amount   | Reference      |
|------|----------|----------|----------------|
| 6728 | lauren   | 230,57€  | id2345deposit  |
| 6739 | bob      | 500,00€  | id96845deposit |
| 6740 | martha   | 213,15€  | id75345deposit |
| 6741 | khaled   | 50,00€   | id88345deposit |
| 6742 | toni     | 1500,00€ | id28754deposit |
| 6743 | chris    | 10,00€   | id27546deposit |
| 6744 | francesc | 750,00€  | id97453deposit |

Credit Bonus

Use this only to credit bonuses to users.

ID

1423

Amount

20,00€

Type

Referral

Change Status

Confirmed

Beneficiary

user1

Credit Bonus

Pending Bonuses

| ID   | Username | Bonus Amount |
|------|----------|--------------|
| 4233 | alfred   | 25,00€       |
| 4236 | robert   | 50,00€       |
| 4237 | john     | 10,00€       |
| 4238 | maria    | 5,00€        |
| 4239 | ricky    | 50,00€       |
| 4240 | joseph   | 30,00€       |
| 4241 | laura    | 100,00€      |

Confirm Withdrawal

Sends a bank transfer to the investor.

ID

1423

Amount

2500,00€

Reference

SEPA Withdrawal

Change Status

Confirmed

Beneficiary

user1

Approve Withdrawal

Pending Withdrawals

| ID   | Username | Amount  | Reference      |
|------|----------|---------|----------------|
| 2378 | amanda   | 150,00€ | withdrawal2378 |
| 2379 | cillian  | 50,57€  | withdrawal2379 |
| 2380 | harry    | 13,25€  | withdrawal2380 |
| 2381 | albert   | 3,98€   | withdrawal2381 |
| 2382 | mark     | 31,00€  | withdrawal2382 |
| 2383 | elon     | 10,00€  | withdrawal2383 |
| 2384 | alex     | 45,00€  | withdrawal2384 |

Mint New Royalties

Creates new royalties and lists them for sale at primary market. If the order is cancelled, royalties are burned out of the system again. Useful for capital increases at companies.

Amount

5.000.000 Royalties

Price

0,50€

Mint Royalties

Freeze account

Use this only to freeze/unfreeze accounts if there is suspect of fraud or if it's requested by law enforcement.

Username

username

Unfreeze Freeze

## Credit Deposits

When a user sends a bank transfer to the company bank account, it should be credited into the user's virtual wallet so they can use the money inside the website.

That function adds money to the selected user's virtual wallet.

## Credit Bonus

If you run a referral or bonus program, it allows you to send payments and rewards to the users that are engaging with your program.

## Confirm Withdrawal

\*This function is not implemented in the demo.

It allows you to send bank transfers to users.

When a user requests a withdrawal, it will appear on "pending withdrawals", then the admin would accept the request and the system would connect to the bank through API and send the bank transfer.

## Mint New Royalties

You can mint new royalties that will be listed for sale in the market.

As this demo has only one trading pair, the minted royalties will be listed for sale in the default market.

If you list more trading pairs, that function will need to be updated to support the new trading pairs.

## Freeze account

It allows the admin to freeze or unfreeze user accounts.

It should only be used if a user is suspected of committing fraud or law enforcement asks you to freeze their account.

Once their account is freezed, they can no longer trade (buy or sell) nor deposit or withdraw funds.

Once their account is unfreezed again, they will be able to trade, deposit and withdraw as any other user.