

Team FX app - Specification

[Project goals](#)

[Dashboard](#)

[Sharing](#)

[Data Entry](#)

[Gamification](#)

[Coaching and Tips](#)

[Phases](#)

[Phase 1](#)

[Phase 2](#)

[Phase 3](#)

[System Architecture/infrastructure](#)

[API](#)

[Global options](#)

[metadata](#)

[token](#)

[watchlist](#)

[GETting](#)

[DATA fields](#)

[POSTing/PUTing](#)

[trade](#)

[GETting](#)

[DATA fields](#)

[POSTing/PUTing](#)

[trade comments](#)

[GETting](#)

[DATA fields](#)

[POSTing](#)

[users](#)

[GETting](#)

[DATA fields](#)

[POSTing](#)

[stats](#)

[GETting](#)

[DATA fields](#)

[POSTing](#)

[instruments](#)

[GETting](#)

[DATA fields](#)

[POSTing](#)

[patterns](#)
[GETting](#)
[DATA fields](#)
[POSTing](#)
[Dialogs and workflows](#)
[Dashboard](#)
[Watchlist entry](#)
[Trade entry](#)
[Trade comment entry](#)
[Trade result entry](#)
[Background tasks](#)
[Database model](#)
[Interfaces to other systems](#)
[Non functional requirements](#)
[Dictionary/Glossary](#)

Project goals

Online platform for sixty minute traders to input their trades and watchlists. This data will be analysed and provide dashboards to members.

Dashboard

Being able to see at a glance how many trades you've taken, how many took profit, which patterns you trade the most and your most profitable patterns are (not always the same!!)... plus the statistical side of things... average weekly and per trade ROI over a period of time. Graphs and pie charts.

Sharing

A way for traders to post completed trades easily to our forum.

Data Entry

Allow traders to enter Journal and Watchlist entries directly as we could take them through the process step-by-step.

Gamification

Interaction with the system will be gamified with competitions within the community of who is completing watch lists most often, who is the most improved etc. Maybe points for completing the watchlist and posting stopped out trades.

Coaching and Tips

Deliver mini training videos out (like mindset inputs every Sunday night)

Phases

The project will be staged into phases to allow for tight scope and milestones to be set.

Phase 1

The aim of phase one is to replace the spreadsheet currently used by the traders. This will centralise the data, allowing the support team to review it and provide feedback. The website will also give immediate feedback encouraging use.

This phase will deliver the following:-

Server production and development environments to include

Database - The API code will include database schema and seed data to enable testing in codeship.

API v1 as specified in this document

Web application to interface with api v1.

This will allow data entry of the watchlist and journal.

Timetable

Server infrastructure: 2-3 days including domain setup with uat and production environments

Database: 1 day

API: 5 days. 2 days testing

Responsive Website: 7 days. 2 days testing.

Phase 2

The aim of phase two is to extend the website functionality to mobile applications for IOS and Android. Dashboard will also be created for the website.

This phase will deliver the following:-

IOS and Android apps to allow data entry matching the functionality of the website in phase 1.

The Apps will be able to relay push notifications of important events and messages from the administrators.

Creation of the website dashboard to give feedback on statistics of the group and individual trader and links to the forum.

Phase 3

TBC. ideas are:-

Expand entry screens to include examples of candlesticks (doji,cross,tailbar etc) for the watchlist entry screen.

Expand the journal screen to include examples of entry patterns, steve mcqueen etc.

Video updates on the dashboard screen

Forum creation to allow posting and analysis of trades.

System Architecture/infrastructure

Online service will be provided by a database backed API. This will be hosted in the cloud.

Clients to the API will be developed.

1. Web application
2. IOS mobile app
3. Android mobile app
4. Windows mobile app

API

The api will be created in PHP and interface the database.
very light code required. e.g. <https://github.com/joindin/joindin-api>

The RESTful style API will support standard methods and authentication will be via an OAUTH2 token. This is sent part of the Authorization header:

Authorization: OAuth [access_code]

BaseURL: <https://api.domain.com/>:version/:model/:id

ALL endpoints require Authentication.

The convention for the urls are based on a version number and the plural of the controller for the model.

The following maps actions you can take on a record to URLs for Trade Journal entries:

Action	HTTP verb	URL
Find	GET	/trades/1
Find all	GET	/trades
Update	PUT	/trades/1
Create	POST	/trades
Delete	DELETE	/trades/1

Global options

A few get parameters are supported by all requests.

`verbose`: set to yes will return more detail. in some cases the isn't more detail

`start`: for collections this will offset the start, default is 0 and used in conjunction with `resultsperpage`

`resultsperpage`: for collections this will set how many results are returned, default is 20, use with `start` to get data chunks.

metadata

Results include `meta` element that contains pagination links and number of results. user detail will also be included.

token

Base URL is `/v1/token`

This is used to get a login token for a user.

GET with 3 parameters

`grant_type` only password is supported in v1

`username` required and is usually an email address

`password` required

`client_id` required - is client ID for the API

`client_secret` required - is client secret for the API

e.g `/v1/token?grant_type=password&username=boris@example.com&password=letmein`

Each data client will be provided with a `client_id` and secret, e.g web, android and iOS

DATA fields

`token` auth token

watchlist

Base URL is `/v1/watchlists`

GETting

This is a collection and can be filtered: (it is always filtered to the authenticated user)

`startdate:` Search for watchlist entries on or after this date

`enddate:` search for watchlist entries before this date

`pair:` filter based on fx pair (Fixed 7 Chars `USD_CAD`)

DATA fields

<code>user</code>	uid of user (email)
<code>id</code>	uid of watchlist entry
<code>date</code>	date in ISO format <code>2014-01-09T23:59:59+01:00</code>
<code>instrument</code>	fx currency pair in 7 char format <code>USD_CAD</code>
<code>weekly</code>	free text field
<code>daily</code>	free text field
<code>candlestick</code>	free text field
<code>resistancemajor</code>	number upto 4 decimals
<code>resistanceminor</code>	number upto 4 decimals
<code>supportmajor</code>	number upto 4 decimals
<code>supportminor</code>	number upto 4 decimals
<code>notes</code>	free text field

example data

```
{ "user": 1, "id": 5, "date": "2014-01-09T23:59:59+01:00", "instrument": "USD_CAD", "weekly": "bears", "daily": "slight bullish pullback", "candlestick": "strong bullish candle with upper wick",
```

"resistancemajor": "1.3045", "resistanceminor": "", "supportmajor": "1.2705", "supportminor": "1.2844", "notes": "waiting for price to creep up or break below support" }

POSTing/PUTing

you can post new watchlist entries. the following fields are required.

date (PHP uses strtotime)

pair must be a valid pair (via instruments api)

weekly min length 1

daily min length 1

candlestick min length 1

resistancemajor number upto 4 decimals

supportmajor number upto 4 decimals

notes min length 1

the response should return a 201 Created header.

trade

base url is /v1/trades

GETting

This is a collection and can be filtered: (it is always filtered to the authenticated user)

startdate: Search for watchlist entries on or after this date
enddate: search for watchlist entries before this date
pair: filter based on fx pair (Fixed 7 Chars USD_CAD)
filled: yes to return filled orders only.

The trade/comments collection is returned by /v1/trades/:ID/comments

DATA fields

user	uid of user (email)
id	uid of trade entry
capital	Capital Balance (Float) when order was placed.Used for ROI calculation
order_date	date in ISO format 2014-01-09T23:59:59+01:00
risk	the percent of capital risked (usually 1% or less)
instrument	fx currency pair in 7 char format USD_CAD
direction	'BUY' or 'SELL'
pattern	one of the pattern strings, See pattern api
entry	number upto 4 decimals
sl	number upto 4 decimals
tp	number upto 4 decimals
rr	Risk reward Ratio (usually calculated)
filled	Bool if trade has been filled
fill_time	date in ISO format 2014-01-09T23:59:59+01:00
exit_time	date in ISO format 2014-01-09T23:59:59+01:00
exit_type	'SL','TP' or 'MANUAL'
exit_price	number upto 4 decimals

example data

```
{ "user": 1, "id": 35, "capital": "3049.59", "order_date": "2014-01-09T23:59:59+1:00", "risk",  
  "1.0", "instrument": "USD_JPY", "direction": "SELL", "pattern": "TAILBAR", "entry": "109.40",  
  "sl": "109.90", "tp": "108.90", "rr": "1:1", "filled": "TRUE", "fill_time":  
  "2014-01-09T23:59:59+01:00", "exit_time": "2014-01-11T21:59:59+01:00", "exit_type": "TP",  
  "exit_price": "108.90" }
```

POSTing/PUTting

When POSTing to create a new trade entry in the journal, there is usually an opening comments to go with it, see trade comments below. When PUTting a trade entry, the existing entry will be overwritten.

required fields are:

capital
order_date

pair
direction
pattern
entry
sl

trade comments

the base url is `/v1/trades_comments/:ID`

GETting

To get a collection of comments this is returned by `/v1/trades/:ID/comments`

To get a specific trade_comment, the base url with the ID should be used.

DATA fields

user	uid of user (email)
id	uid of trade_comment entry
date	creation date in ISO format 2014-01-09T23:59:59+01:00
comment	the comment itself
trade	the uid of the trade this relates to

POSTing

Creating a comment is done via the trades controller at the collection url

`/v1/trades/:ID/comments`

The API will pick up the other details based on the authentication and url Required fields are:

comment

users

Base URL: /v1/users

GETting

Can be used to get an individual user record, or filtered:

email: Search based on email address of user

DATA fields

id	uid of user (email)
email	email address
date	creation date in ISO format 2014-01-09T23:59:59+01:00
full_name	User's name if given
facebook_id	facebook token to be used for posting

POSTing

Not supported. Admin gateway creates users.

stats

These will be defined in phase 2

GETting

DATA fields

POSTing

instruments

GETting

This is a collection that will return a list of the instruments supported by the trade data. you can use this info to build a dropdown list.

DATA fields

id	uid of instrument entry
instrument	fx currency pair in 7 char format USD_CAD
displayname	Human readable name of the instrument
tickvalue	the smallest value available e.g. 0.0001 or 0.01
active	show if instrument is active or not

POSTing

Must be an admin user to create instruments

patterns

GETting

This provides the patterns collection. a list of patterns used for a trade

DATA fields

id	uid of pattern entry
patternname	Short version of the name
displayname	Human readable name of the pattern
altnames	json list of alternative names
video_uri	uri to video describing the pattern
image_uri	uri to image describing the pattern

POSTing

Must be an admin user to create signals

Dialogs and workflows

The following diagrams are 'impressions' of what the pages could look like. The preference is for the designer to use the appropriate UI library. For the website this should be Bootstrap, iOS should use Cocoa, and the clean look is preferred.

Dashboard

This is part of phase 2:

Stats:

ratio of profitable trades.	profitable trades/trades filled
ratio of filled trades:	filled trades/trades placed
av. return month	av. roi on last 30 days of trades
av. return year	av. roi on last 365 days of trades
favourite signal	most used signal over last 30 days

fav signal 2month
profit signal
profit signal 2 month

most used signal over the last 60 days
most used signal on profitable trades 30days
most used signal on profitable trades 60 days

Watchlist entry

The entry screens will be presented in a step by step manner to break up the data entry.
The Date and Pair will be displayed along with one or two questions.

- Who is winning in the Weekly chart
- Who is winning in the Daily chart

- Candlestick Description
- Next Major Resistance
- Next Major Support
- What do i need to happen before I enter a trade

Forward and back buttons will be presented, with a save analysis at the end of each pair entry.

The screen can be advanced by the enter key also.

Each of the pairs will be cycled through in order.

!! We could display either the previous pair data or previous day for that pair !!

Trade entry

Multiscreen entry with the following questions on each:

- Capital balance
- Pair
- Direction (Buy/Sell)
- Strategy
- Entry (Validated to decimal points)
- SL (Validated to decimal points and higher/lower than Entry based on Direction)
- TP (Validated as SL or Dynamic allowed)
- Risk:Reward ratio before next zone/level (could be calculated from previous data)
- Trade entry comments.
- Trade summary displayed for checking or platform entry.

Amount at Risk: capital * Risk %
2000 * 1% = 20

Number of Units for Order:

$$\left(\left(\text{Amount at Risk} / \text{Pip size} \right) / 1\text{-pip value} \right) * 10000$$
$$\left(\left(20 / 45 \right) / 0.89 \right) * 10000 = 4994$$
$$\left(\left(0.44 \right) / 0.89 \right) * 10000 = 4994$$
$$\left(0.49438202 \right) * 10000 = 4994$$

Trade result entry

First displays a list of recent trades to select from. These are filtered to hide trades that have 'exited'.

Two actions are possible. Add a comment and exit results.

trade list screen

Trade comment

Displays ticket detail with previous comments and provides a place for the comment entry. Also contains button to mark trade as Filled with a datetime.

Trade results

Multi screen entry for results of a close trade

- Datetime trade exited
- Method, stopped out, take profit or manual exit
- exit price
- pips won/lost (can be calculated)
- ROI (can be calculated)
- closing comments/ ahas/lessons
- (Phase 2) post to forum button (can be disabled if profitable trade)

trade results screens

Background tasks

Reports and emails will be generated on a batch basis. This will populate dashboard data and send email stat reports.

These will be created/expanded in phase 2.

Database model

As reporting and statistics will be an integral part of the applications, it is recommended that a database system is designed to be optimised for data entry of trades and reporting capabilities.

INT

VARCHAR

FLOAT(10,4)

Trade Journal	Watchlist	Instruments	patterns
id user capital order_date risk instrument direction pattern entry sl tp rr {trade_comments} filled fill_time exit_time exit_type exit_price	id user date instrument weekly daily candlestick resistancemajor resistanceminor supportmajor supportminor notes	id instrument displayname tickvalue active	id patternname displayname altnames video_uri image_uri

Stats	oauth_access_tokens	users
user Av ROI Fav Pattern Profit Pattern Trades Placed Trades Profit Month Av ROI Month Trades Placed Month Trades Profit	id access_token access_token_secret consumer_key user_id last_used_date	id date email full_name password

--	--	--

api_keys
id date expiration provider current next user_id

Interfaces to other systems

Brokers provide data which could be used to measure progress, place orders and give an analysis platform.

Phase 1: no extra interfaces

Phase 2: a forum (hosted on facebook or other) to allow automatic posting

Phase 3: Oanda API to gather chart data, so a png can be produced showing the users trade entry with candlestick data. <http://developer.oanda.com/rest-sandbox/development-guide/>

Non functional requirements

Peak usage expected to be 9pm-11pm

Dictionary/Glossary

FX - Foreign exchange

Instrument - A financial instrument that you can trade via a broker

Currency - A unit of money expressed in a country's legal tender

PIP - a unit of price movement of FX instruments

Entry - a price at which an instrument trade should be opened (Buy or Sell)

Exit - the opposite action to Entry for a given trade

TP - Take profit. a price point at which an exit will be executed when the price is going with the trade.

SL - Stop Loss. a price point at which an exit will be executed when the price is going against the trade.

user - an end user of the application

token - an OAUTH token used to authenticate a user with a provider without passwords

provider or web service - the server software of the api

consumer - the client software of the api

API - Application Program(able) Interface. a standardised way of interacting with a provider