

Get Strategies

- URL
 - `/api/strategies`
- Method
 - `GET`
- URL Params : none
- Data Params
- Success Response
 - Code : 200
 - Content

```
[
  {
    "name": "CCI",
    "params": "# constant multiplier. 0.015 gets to around 70%
fit\nconstant = 0.015\n\n# history size, make same or smaller
than history\nhistory = 90\n\n[thresholds]\nup = 100\ndown
= -100\npersistence = 0"
  },
  {
    "name": "DEBUG_single-advice",
    "params": ""
  },
  {
    "name": "DEBUG_toggle-advice",
    "params": ""
  }
]
```

```
},
{
  "name": "DEMA",
  "params": "weight = 21\n\n[thresholds]\ndown = -0.025\nup = 0.025\n",
},
{
  "name": "MACD",
  "params": "short = 10\nlong = 21\nsignal = 9\n\n[thresholds]\ndown = -0.025\nup = 0.025\npersistence = 1",
},
{
  "name": "PPO",
  "params": "short = 12\nlong = 26\nsignal = 9\n\n[thresholds]\ndown = -0.025\nup = 0.025\npersistence = 2",
},
{
  "name": "RSI",
  "params": "interval = 14\n\n[thresholds]\nlow = 30\nhigh = 70\npersistence = 1",
},
{
  "name": "StochRSI",
  "params": "interval = 3\n\n[thresholds]\nlow = 20\nhigh = 80\npersistence = 3",
},
{
  "name": "TMA",
```

```
    "params": "short = 7\nmedium = 25\nlong = 99\n",
  },
  {
    "name": "TSI",
    "params": "short = 13\nlong = 25\n\n[thresholds]\nlow = -25\nhigh = 25\npersistence = 1"
  },
  {
    "name": "UO",
    "params": "[first]\nweight = 4\nperiod = 7\n\n[second]\nweight = 2\nperiod = 14\n\n[third]\nweight = 1\nperiod = 28\n\n[thresholds]\nlow = 30\nhigh = 70\npersistence = 1\n"
  },
  {
    "name": "custom",
    "params": "my_custom_setting = 10"
  },
  {
    "name": "noop",
    "params": ""
  },
  {
    "name": "talib-macd",
    "params": "[parameters]\noptInFastPeriod = 10\noptInSlowPeriod = 21\noptInSignalPeriod = 9\n\n[thresholds]\ndown = -0.025\nup = 0.025"
  },
  {
```

```

    "name": "tulip-adx",
    "params": "historySize = 80\noptInTimePeriod = 15\ncandleSize = 10\n\n[thresholds]\nup = 30\ndown = 20"
  },
  {
    "name": "tulip-macd",
    "params": "[parameters]\noptInFastPeriod = 10\noptInSlowPeriod = 21\noptInSignalPeriod = 9\n\n[thresholds]\ndown = -0.025\nup = 0.025"
  },
  {
    "name": "tulip-multi-strat",
    "params": "optInTimePeriod = 2\noptInFastPeriod = 4\noptInSlowPeriod = 7\noptInSignalPeriod = 23.707189677282354\n\ncandleSize = 1\nhistorySize = 4\n\nup = 24.52\ndown = 55.74\nmacd_up = 14.498233170768081\nmacd_down = -9.65220944122072\n"
  },
  {
    "name": "varPP0",
    "params": "momentum = \"TSI\" # RSI, TSI or UO\n\n[thresholds]\nweightLow = 120\nweightHigh = -120\n\n# How many candle intervals should a trend persist\n# before we consider it real?\npersistence = 0"
  }
]

```

- Error Response

- Code : 404
- Content : ``Not Found``
- Sample Call

```
$.ajax({  
  url: "/api/strategies",  
  dataType: "json",  
  type : "GET",  
  success : require('web/routes/strategies')  
});
```

Get “ConfigPart”

- URL
`/api/configPart/:part`
- Method
`GET`
- URL Params
 - Required
`part = 'paperTrader' or 'candelWriter' or 'performanceAnalyzer'`
- Data Params
none
- Success Response
 - Code : 200
 - Content
 - paperTrader

```
{"part": "feeMaker = 0.25\nfeeTaker = 0.25\nfeeUsing = 'maker'\nslippage = 0.05\n\n[simulationBalance]\nasset = 1\ncurrency = 100\n" }
```

- candleWriter

```
{"part": "enabled = true\nadapter = \"sqlite\""} 
```

- performanceAnalyzer

```
{"part": "# Used to calculate sharpe ratio\n# yearly % of riskfree return for example\n# treasury bonds or bank interest.\n\nriskFreeReturn = 2"} 
```

- Error Response

- Code : 200

- Content : `error :(`

OR

- Code : 404

- Content : ``Not Found``

- Sample Call

```
$.ajax({  
  url: "/api/configPart/paperTrader",  
  dataType: "json",  
  type : "GET",  
  success : require('web/routes/configPart')  
});
```

Get API Keys

- URL :

`/api/apiKeys`

- Method :

`GET`

- URL Params

none

- Data Params

none

- Success Response

- Code : 200

- Content : (*example; empty if no API keys saved*)

```
["poloniex", "gdax"]
```

- Error Response

- Code : 404

- Content : ``Not Found``

- Sample Call

```
$.ajax({  
  url: "/api/apiKeys",  
  dataType: "json",  
  type : "GET",  
  success : require('web/routes/apiKeys').get  
});
```

- Notes
 - It appears that the JSON only returns the exchange names in an list, presumably the literal keys are hidden from REST implementation.

ADD API Key

- URL
 - `/api/addApiKey`
- Method
 - `POST`
- URL Params
 - none
- Data Params
 - none
- Success Response
 - Code : 200
 - Content

```
{"status": "ok"}
```

- Error Response
 - Code : 405
 - Content : `'Method Not Allowed'`
 - OR
 - Code : 404
 - Content : ``Not Found``

- Sample Call

```
$.ajax({  
  url: "/api/addApiKey",  
  dataType: "json",  
  type : "POST",  
  success : require('web/routes/apiKeys').add  
});
```

Remove API Key

- URL

/api/removeApiKey

- Method

POST

- URL Params

none

- Data Params

none

- Success Response

- Code : 200
- Content

```
{"status": "ok"}
```

- Error Response

- Code : 405

- Content

'Method Not Allowed'

OR

- Code : 404

- Content : `Not Found`

- Sample Call

```
$.ajax({  
  url: "/api/removeApiKey",  
  dataType: "json",  
  type : "POST",  
  success : require('web/routes/apiKeys').remove  
});
```

Get Imports

- URL

/api/imports

- Method

GET

- URL Params

none

- Data Params

none

- Success Response

- Code : 200

- Content (*example*)

```
[
  {
    "watch": {
      "exchange": "poloniex",
      "currency": "USDT",
      "asset": "BTC"
    },
    "id": "254834834128682",
    "latest": "2018-06-11T20:46:58Z",
    "from": "2018-03-11 21:17",
    "to": "2018-06-11 21:17",
    "done": true
  }
]
```

- Error Response
 - Code : 404
 - Content : ``Not Found``
- Sample Call

```
$.ajax({
  url: "/api/imports",
  dataType: "json",
  type : "GET",
  success : require('web/routes/list')('imports')
});
```

Get Exchanges

- URL
`/api/exchanges`
- Method
`GET`
- URL Params
none
- Data Params
none
- Success Response
 - Code : 200
 - Content : JSON (See file : GetExchanges.md)
- Error Response
 - Code : 404
 - Content : ``Not Found``
- Sample Call

```
$.ajax({  
  url: "/api/exchanges",  
  dataType: "json",  
  type : "GET",  
  success : require('web/routes/exchanges')  
});
```

Start a (Date-Range) Scan

- URL
`/api/scan`
- Method
`POST`
- URL Params
none
- Data Params
none
- Success Response
 - Code : 200
 - Content (JSON)
- Error Response
 - Code : 405
 - Content
`'Method Not Allowed'`
- Sample Call

```
$.ajax({  
  url: "api/scan",  
  dataType: "json",  
  type : "POST",  
  success : require('web/routes/scanDateRange')  
});
```

Start (Data-set) Scan

- URL

/api/scansets

- Method

POST

- URL Params

none

- Data Params

none

- Success Response

- Code : 200

- Content (eample

example)

```
```json
```

```
{
```

```
 "datasets": [
```

```
 {
```

```
 "exchange": "binance",
```

```
 "currency": "BTC",
```

```
 "asset": "NEO",
```

```
 "ranges": [
```

```
 {
```

```
 "from": 1515382020,
```

```
 "to": 1528424700
```

```
 }
```

```
]
```

```
 }
```

```
],
```

```
"errors": []
}
```

- Error Response
  - Code : 405
  - Content : 'Method Not Allowed'
- Sample Call

```
$.ajax({
 url: "/api/scansets",
 dataType: "json",
 type : "POST",
 success : require('web/routes/scanDatasets')
});
```

## Start a Backtest

- URL  
/api/backtest
- Method  
POST
- URL Params  
none
- Data Params  
none
- Success Response
  - Code : 200

- Content (*example*)

```
{
 "trades": [
 {
 "action": "sell",
 "price": 107.41,
 "portfolio": {
 "asset": 0,
 "currency": 207.08777,
 "balance": 205.49
 },
 "balance": 207.08777,
 "date": "2018-06-11T10:43:00.000Z"
 }
],
 "candles": [
 {
 "close": 108.23,
 "start": "2018-06-11T01:43:00.000Z"
 },
 {
 "close": 108.27,
 "start": "2018-06-11T02:43:00.000Z"
 },
 {
 "close": 108.11,
 "start": "2018-06-11T03:43:00.000Z"
 }
]
}
```



```
},
{
 "close": 108.17,
 "start": "2018-06-11T04:43:00.000Z"
},
{
 "close": 107.05,
 "start": "2018-06-11T05:43:00.000Z"
}
],
"report": {
 "currency": "USD",
 "asset": "LTC",
 "startTime": "2018-06-11 01:43:00",
 "endTime": "2018-06-11 19:43:00",
 "timespan": "18 hours",
 "market": 0.5972130059721366,
 "balance": 207.08777,
 "profit": 1.5977699999999997,
 "relativeProfit": 0.7775414862037024,
 "yearlyProfit": "778.09801230",
 "relativeYearlyProfit": "378.65492837",
 "startPrice": 105.49,
 "endPrice": 106.12,
 "trades": 1,
 "startBalance": 205.49,
 "sharpe": 0,
 "alpha": 1.0005569940278605
```

```
},
"roundtrips": []
}
```

- Error Response
  - Code : 405
  - Content : 'Method Not Allowed'
- Sample Call

```
$.ajax({
 url: "/api/backtest",
 dataType: "json",
 type : "POST",
 success : require('web/routes/backtest')
});
```

## Start an Import

- URL  
/api/import
- Method  
POST
- URL Params
  - Required
  - Optional none
- Data Params  
none

- Success Response
  - Code : 200
  - Content (*example*)

```
{
 "watch": {
 "exchange": "poloniex",
 "currency": "USDT",
 "asset": "BTC"
 },
 "id": "719695361793284",
 "latest": "",
 "from": "2018-03-12 20:33",
 "to": "2018-06-12 20:33"
}
```

- Error Response
  - Code : 405
  - ContentSample Call 'Method Not Allowed'
- 

```
$.ajax({
 url: "/api/import",
 dataType: "json",
 type : "POST",
 success : require('web/routes/import')
});
```

- Notes
  - (from source) *"Requires a post body with a config object"*

## Start new Gekko bot

- URL
  - `startGekko`
- Method
  - `POST`
- URL Params
  - none
- Data Params
  - none
- Success Response
  - Code : 200
  - Content (*example*)

```
{
 "watch": {
 "exchange": "poloniex",
 "currency": "USDT",
 "asset": "BTC"
 },
 "id": "165504961964151",
 "startAt": "",
 "latest": "",
 "mode": "realtime",
 "type": "watcher"
}
```

```
}
```

- Error Response
  - Code : 405
  - Content: 'Method Not Allowed'
- Sample Call

```
$.ajax({
 url: "/api/startGekko",
 dataType: "json",
 type : "POST",
 success : require('web/routes/startGekko')
});
```

## Get Running Gekko bots

- URL
- /api/gekkos
- Method
  - GET
- URL Params
  - none
- Data Params
  - none
- Success Response
  - Code : 200
  - Content (*example*)

```
[
 {
 "watch": {
 "exchange": "gdax",
 "currency": "USD",
 "asset": "LTC"
 },
 "id": "249270745859179",
 "startAt": "2018-06-11T01:01:56Z",
 "latest": "2018-06-11T22:47:02Z",
 "mode": "realtime",
 "type": "watcher",
 "firstCandle": {
 "start": "2018-06-11T01:01:00.000Z",
 "open": 104.78,
 "high": 104.78,
 "low": 104.75,
 "close": 104.75,
 "vwp": 104.7664932330549,
 "volume": 18.18927793,
 "trades": 2
 },
 "lastCandle": {
 "start": "2018-06-11T22:46:00.000Z",
 "open": 105.54,
 "high": 105.67,
 "low": 105.51,
```

```
 "close": 105.67,
 "vwp": 105.64127875100343,
 "volume": 46.2163000000000004,
 "trades": 24
 }
},
`` - Error Response
 - Code : 404
 - Content : `` `Not Found` ``
- Sample Call
```javascript  
$.ajax({  
  url: "/api/gekkos",  
  dataType: "json",  
  type : "GET",  
  success : require('web/routes/list')('gekkos')  
});
```

Kill (Stop) a Gekko bot

- URL
 - `/api/killGekko`
- Method
- URL Params
- Data Params
- Success Response
 - Code : 200

- Content (N/A; *see notes*)
- Error Response
 - Code : 405
 - Content : 'Method Not Allowed'
- Sample Call

```
$.ajax({  
  url: "/api/killGekko",  
  dataType: "json",  
  type : "POST",  
  success : require('web/routes/killGekko')  
});
```

- Notes
 - While the API interface seems to be implemented, there is yet no way kill a gekko (trading bot) from the UI, the author (*askmike*) has said it is on the project's TODO list as of Feburary, 2018.

Get Candles

- URL
1/api/getCandles
- Method
POST
- URL Params
- Data Params
- Success Response

- Code : 200
- Content (*example*)

```
[
  {
    "id": 178424,
    "start": 1528835100,
    "open": 6559.14436475,
    "high": 6566.8089823,
    "low": 6559.14436475,
    "close": 6565.89112461,
    "vwp": 6565.880612572557,
    "volume": 2.0530371,
    "trades": 30
  },
  {
    "id": 178425,
    "start": 1528835160,
    "open": 6565.89112461,
    "high": 6566.6,
    "low": 6542.14140955,
    "close": 6545.88363339,
    "vwp": 6549.85162973142,
    "volume": 9.075842959999997,
    "trades": 44
  },
  {
    "id": 178426,
```

```
"start": 1528835220,
"open": 6558.4967535,
"high": 6558.4967535,
"low": 6554.43784124,
"close": 6554.43784124,
"vwp": 6556.471354951906,
"volume": 0.0006102,
"trades": 3
}
]
```

- Error Response
 - Code : 405
 - Content : 'Method Not Allowed'
- Sample Call

```
$.ajax({
  url: "/api/killGekko",
  dataType: "json",
  type : "POST",
  success : require('web/routes/getCandles')
});
```

- Notes
 - From source :

```
//simple POST request that returns the candles requested
// expects a config like:
```

```
let config = {  
  watch: {  
    exchange: 'poloniex',  
    currency: 'USDT',  
    asset: 'BTC'  
  },  
  daterange: {  
    from: '2016-05-22 11:22',  
    to: '2016-06-03 19:56'  
  },  
  adapter: 'sqlite',  
  sqlite: {  
    path: 'plugins/sqlite',  
  
    dataDirectory: 'history',  
    version: 0.1,  
  
    dependencies: [{  
      module: 'sqlite3',  
      version: '3.1.4'  
    }]  
  },  
  candleSize: 100  
}
```