

---

# **Coinmax Documentation**

*Release 0.2.0*

**Coinmax Team**

**Jul 31, 2019**



---

# Contents

---

<b>1</b>	<b>Testing Environment</b>	<b>1</b>
<b>2</b>	<b>Rest API</b>	<b>3</b>
2.1	Public Endpoints . . . . .	4
2.2	Generate Credentials . . . . .	5
2.3	Orders . . . . .	6
2.4	Trades . . . . .	8
2.5	Funds . . . . .	9
2.6	Deposits . . . . .	10
2.7	Withdrawals . . . . .	11
2.8	Sample GET Program . . . . .	12
<b>3</b>	<b>WebSocket</b>	<b>15</b>
3.1	Subscriptions . . . . .	16
3.2	Data Structures . . . . .	18
<b>4</b>	<b>Api Keys</b>	<b>23</b>
<b>5</b>	<b>Fees</b>	<b>25</b>
<b>6</b>	<b>Disclaimer</b>	<b>27</b>



# CHAPTER 1

---

## Testing Environment

---

<https://sandbox.coinmax.com.au> is our sandbox environment, feel free to play with it.



## CHAPTER 2

---

### Rest API

---

Our API follows typical HTTP status codes for success and failure. Below is a table to indicate what each of them means.

Table 1: HTTP status codes

Status code	Description
200	Your request is accepted and body will have data if any
400	Invalid request
403	Unauthorized request
429	API rate limit reached
500	Internal error

**Base URL** /api/

List of API endpoints

Table 2: API list

Endpoint	HTTP method	Authorization required?	Re-	Description
/public/products	GET	NO		Get a list of available asset pairs for trading
/public/assets	GET	NO		Get a list of supported assets
/client/order	POST	YES		Place an order
/client/order/cancel	POST	YES		Cancel an order
/client/orders	GET	YES		Get your orders
/client/trades	GET	YES		Get your trades
/client/funds	GET	YES		Get available funds
/client/deposits	GET	YES		Get your deposits
/client/withdrawals	GET	YES		Get your withdrawals
/client/withdraw	POST	YES		Place a withdrawal

## 2.1 Public Endpoints

You can get the listed products and assets list from our public endpoints.

### Get Products

Get a list of available asset pairs for trading.

HTTP REQUEST:

GET /public/products

Sample Response:

```
{
  "BTC-AUD": {
    "baseAsset": {
      "assetDisplayName": "Bitcoin",
      "assetName": "BTC",
      "blockExplorer": "https://testnet.blockchain.info/tx/",
      "depositServiceStatus": "SUSPENDED",
      "minOrderQty": 1,
      "precisionDigits": 8,
      "precisionValue": 100000000,
      "tickSize": 1,
      "withdrawalServiceStatus": "RUNNING"
    },
    "quoteAsset": {
      "assetDisplayName": "Australian Dollar",
      "assetName": "AUD",
      "blockExplorer": "",
      "depositServiceStatus": "SUSPENDED",
      "minOrderQty": 1,
      "precisionDigits": 2,
      "precisionValue": 100,
      "tickSize": 1,
      "withdrawalServiceStatus": "RUNNING"
    },
    "securityName": "BTC-AUD",
  }
}
```

(continues on next page)

(continued from previous page)

```
    "splitter": "-",
    "tradingServiceStatus": "RUNNING"
  }
}
```

## Get Assets

Get list of supported assets.

---

**Note:** Not all assets may be currently in use for trading.

---

HTTP REQUEST:

GET **/public/assets**

Sample Response:

```
{
  "BTC": {
    "assetDisplayName": "Bitcoin",
    "assetName": "BTC",
    "blockExplorer": "https://testnet.blockchain.info/tx/",
    "depositServiceStatus": "SUSPENDED",
    "minOrderQty": 1,
    "precisionDigits": 8,
    "precisionValue": 100000000,
    "tickSize": 1,
    "withdrawalServiceStatus": "RUNNING"
  },
  "LTC": {
    "assetDisplayName": "Litecoin",
    "assetName": "LTC",
    "blockExplorer": "",
    "depositServiceStatus": "SUSPENDED",
    "minOrderQty": 1,
    "precisionDigits": 6,
    "precisionValue": 1000000,
    "tickSize": 1,
    "withdrawalServiceStatus": "RUNNING"
  }
}
```

## 2.2 Generate Credentials

You need to generate your API credentials before placing orders via API. Please follow the following steps for the same.

1. Go to [coinmax.com.au](https://coinmax.com.au) and login. If you don't have an account, please sign up.

2. Once you log in, on the top right corner, click on the the dropdown and click “Settings”.
3. On the Settings page, go to “API Credentials”.
4. Click “Generate”.
5. Note down the Client Key and Secret somewhere safe, as it will only be displayed once.

## 2.3 Orders

### 2.3.1 Place Order

HTTP REQUEST:

POST `/api/client/order`

1. Create the order request with intended parameters. This will be the body of your HTTP request. Below are the details of each supported parameter.

Order Structure:

```
{
  "qty" : "0.0001",
  "price" : "100",
  "side" : "BUY", //Valid options are "BUY" and "SELL"
  "symbol" : "BTC-AUD", Complete list of supported products can be requested,
  ↳from https://coinmax.com.au/api/clientConfig/ under products key
  "type" : "LIMIT", // Supported parameters are "LIMIT" and "STOP LIMIT"
  "triggerPrice" : "78.99" // Only applicable if type is "STOP LIMIT",
  "validity": "GOOD TILL CANCEL", Supported parameters are "GOOD TILL CANCEL",
  ↳"IMMEDIATE OR CANCEL" and "FILL OR KILL",
  "timestamp": 1538046192974 //Current time in milliseconds
}
```

2. Sign your order request using sha256 and your client secret. Set the hex value of your signature in HTTP header “X-API-SIGNATURE”.
3. Set your API key in HTTP header “X-API-KEY”.
4. Place the request on URL <https://coinmax.com.au/api/client/order>

Sample NodeJS program:

```
const crypto = require('crypto');
let time = new Date().getTime();
const order = {
  "symbol": "BTC-AUD",
  "qty": "1",
  "price": "806",
  "side": "BUY",
  "type": "LIMIT",
  "triggerPrice": "",
  "validity": "GOOD TILL CANCEL",
  "timestamp": time
};
const CLIENT_KEY = "a385061780f7c9da5f1d1ad53ac644e7";
const CLIENT_SECRET =
  ↳"b9a03c3c32de9f2691a6309ba77a5a189b6720ff3a9c2b23e0af0ad7384438ec";
```

(continues on next page)

(continued from previous page)

```
var sign = crypto.createHmac('sha256', CLIENT_SECRET).update(JSON.  
  ↳stringify(order)).digest(  
    'hex')  
  
var request = require("request");  
  
var options = {  
  url: 'https://coinmax.com.au/api/client/order',  
  headers: {  
    'X-API-KEY': CLIENT_KEY,  
    'X-API-SIGNATURE': sign  
  },  
  method: 'post',  
  json: true,  
  body: order  
};  
  
request(options, (error, res, body) => {  
  if (error) {  
    console.warn(error);  
  }  
  console.log(body);  
})
```

### 2.3.2 Cancel Order

HTTP REQUEST:

POST `/api/client/order/cancel`

Follow the same steps as *Place Order*, but alter the body with following parameters:

Cancel Order Structure:

```
{  
  "orderId": "26001", //Order id to cancel  
  "symbol": "ETH-AUD", // Symbol  
  "timestamp": time  
}
```

---

**Note:** A successful result from *cancel* API does not mean that order is cancelled, it just means that your cancellation request is accepted. To know the actual status, subscribe to *orderUpdate* channel on WebSocket.

---

### 2.3.3 Get Orders

HTTP REQUEST:

GET `/api/client/orders`

Table 3: Query parameters

Parameter	Description
timestamp	Current timestamp in milliseconds (Required)
symbol	Asset pair e.g. BTC-AUD (Required)
pageSize	Number of orders to return, default value is 50
page	Directly jump to a particular page number by skipping previous records, default value is 0

Sample Response:

```
[
  {
    "Symbol": "BTC-AUD",
    "FilledQty": "0.00000000",
    "Price": "1000.00",
    "TriggerPrice": "0.00",
    "Qty": "0.00010000",
    "OrderType": "LIMIT",
    "OrderValidity": "GOOD TILL CANCEL",
    "OrderStatus": "CANCELLED",
    "Side": "BUY",
    "ExchangeOrderId": "391708",
    "LastFillQty": "0.00000000",
    "LastFillPrice": "0.00",
    "Timestamp": "1561088065018000",
    "Commission": "0.00",
    "ClientOrderId": "391708",
    "SeqNo": 2,
    "Error": ""
  }
]
```

*Sample GET Program*

## 2.4 Trades

HTTP REQUEST:

GET **/api/client/trades**

Table 4: Query parameters

Parameter	Description
timestamp	Current timestamp in milliseconds (Required)
symbol	Asset pair e.g. BTC-AUD (Required)
pageSize	Number of trades to return, default value is 50
page	Directly jump to a particular page number by skipping previous records, default value is 0

Sample Response:

```
[
  {
```

(continues on next page)

(continued from previous page)

```

"Symbol": "BTC-AUD",
"FilledQty": "0.01086340",
"Price": "5616.89",
"TriggerPrice": "0.00",
"Qty": "1.00000000",
"OrderType": "LIMIT",
"OrderValidity": "GOOD TILL CANCEL",
"OrderStatus": "PARTIAL FILLED",
"Side": "BUY",
"ExchangeOrderId": "180708",
"ExecutionId": "4611686018427535980",
"LastFillQty": "0.01086340",
"LastFillPrice": "5616.89",
"Timestamp": "1557936662842695",
"Commission": "0.15",
"ClientOrderId": "180708",
"SeqNo": 2,
"Error": ""
}
]

```

*Sample GET Program*

## 2.5 Funds

HTTP REQUEST:

GET /api/client/funds

Table 5: Query parameters

Parameter	Description
timestamp	Current timestamp in milliseconds (Required)

Sample Response:

```

{
  "funds": [
    {
      "assetName": "AUD",
      "availableForOrders": "1000005.25",
      "reserved": "-10253.65",
      "pendingDeposits": "100.00",
      "depositAddress": "",
      "seqNo": 80
    },
    {
      "assetName": "ZEC",
      "availableForOrders": "100.000000",
      "reserved": "0.000000",
      "pendingDeposits": "0.000000",
      "depositAddress": "tmYAfLpSziBirkKjNRuoS3mMENNuuTatrND",
      "seqNo": 1
    }
  ],
}

```

(continues on next page)

(continued from previous page)

```

{
  "assetName": "LTC",
  "availableForOrders": "100.000000",
  "reserved": "0.000000",
  "pendingDeposits": "0.000000",
  "depositAddress": "QV33qHRScoxKSyXojS8G7N8FkEHWnrf6Pd",
  "seqNo": 1
},
{
  "assetName": "ETH",
  "availableForOrders": "159.981969",
  "reserved": "0.000000",
  "pendingDeposits": "0.000000",
  "depositAddress": "QVtlySHovNTGAh2cQCaG8QD6N535GyjG83",
  "seqNo": 5
},
{
  "assetName": "BTC",
  "availableForOrders": "1.00000000",
  "reserved": "0.00000000",
  "pendingDeposits": "0.00000000",
  "depositAddress": "2MvjxQopZU3z3TfexpWEEqT4Q9hbSQugsjs",
  "seqNo": 1
}
]
}

```

*Sample GET Program*

## 2.6 Deposits

HTTP REQUEST:

GET /api/client/deposits

Table 6: Query parameters

Parameter	Description
timestamp	Current timestamp in milliseconds (Required)
pageSize	Number of trades to return, default value is 50
page	Directly jump to a particular page number by skipping previous records, default value is 0

Sample Response:

```

[
  {
    "asset": "AUD",
    "paymentAgent": "Pay ID",
    "amount": "5000.00",
    "referenceId": "122692",
    "timestamp": "1556291940840000",
    "error": "",
    "metadata": {

```

(continues on next page)

(continued from previous page)

```

    "code": "4cef02"
  },
  "feeCharged": "0.00",
  "transactionStatus": "INITIATED"
},
{
  "asset": "AUD",
  "paymentAgent": "Pay ID",
  "amount": "10.00",
  "referenceId": "123692",
  "timestamp": "1556291996905000",
  "error": "",
  "metadata": {
    "code": "7cf4de"
  },
  "feeCharged": "0.00",
  "transactionStatus": "INITIATED"
}
]

```

*Sample GET Program*

## 2.7 Withdrawals

HTTP REQUEST:

GET /api/client/withdrawals

Table 7: Query parameters

Parameter	Description
timestamp	Current timestamp in milliseconds (Required)
pageSize	Number of trades to return, default value is 50
page	Directly jump to a particular page number by skipping previous records, default value is 0

Sample Response:

```

[
  {
    "asset": "BTC",
    "amount": "0.00300000",
    "referenceId": "187698",
    "timestamp": "1558023541957000",
    "error": "",
    "feeCharged": "0.00020000",
    "transactionStatus": "CONFIRMED",
    "nodeGeneratedTxId":
    ↪ "41f5bf416c601724351682e50bd324e46ec84567f0f0f93731ed8fc1c76e37fa",
    "address": "2MvjxQopZU3z3TfexpWEEqT4Q9hbSQugsjs",
    "recipientAddress": "2MvjxQopZU3z3TfexpWEEqT4Q9hbSQugsjs",
    "approvedAmount": "0.00280000"
  },
  {

```

(continues on next page)

(continued from previous page)

```

"asset": "ETH",
"amount": "0.045000",
"referenceId": "178703",
"timestamp": "1557937454067000",
"error": "",
"feeCharged": "0.001000",
"transactionStatus": "CONFIRMED",
"nodeGeneratedTxId":
↪ "0x75b7a8fb1bc8b0ec03404212de4d7913b45eb3859c0c3a334e81291c6aaab889",
"address": "0xf2af5b5ebcccade3461b654fd16c42c82408c0a5",
"recipientAddress": "0xf2af5b5ebcccade3461b654fd16c42c82408c0a5",
"approvedAmount": "0.044000"
}
]

```

*Sample GET Program***POST /api/client/withdraw**

Table 8: Body params

Parameter	Description
asset	One of the listed asset's name (Required)
amount	Amount that you want to withdraw from your account (required)
recipient	Valid address for the recipient of this transaction (required)
timestamp	Current timestamp in milliseconds (Required)

## Sample Request:

```

{
  "asset": "BTC",
  "recipient": "2NGZrVvZG92qGYqzTLjCAewvPZ7JE8S8VxE",
  "amount": "0.001",
  "timestamp": time
}

```

**Note:** This api is available for crypto withdrawals and whitelisted recipient addresses only. Moreover, make sure that your api key has the required provisions to use this functionality.

**Note:** As of now users can neither white list withdrawal addresses nor can they update ApiKey provisions by themselves. Kindly contact support if you need to do so.

*Api Keys**Sample GET Program*

## 2.8 Sample GET Program

1. You can fetch funds, orders, trades, deposits, withdrawals using the GET APIs.

2. You can use query params in URL, timestamp is a required parameter to prevent replay attacks.
3. Similar to POST APIs, GET APIs of Coinmax also require signature, but the steps vary.
4. You need to sign the API URL instead of request body, By API URL we mean the part ahead of the base url, For e.g, In “<https://coinmax.com.au/api/client/trades?timestamp=12313443&page=0&symbol=ETH-AUD>” “<https://coinmax.com.au/api/client/>” is the base URL and “[/trades?timestamp=1540472319692&page=0&symbol=ETH-AUD](https://coinmax.com.au/api/client/trades?timestamp=1540472319692&page=0&symbol=ETH-AUD)” is the API URL.

Below is a sample NodeJS program to fetch user trades, you can replace the *apiURL* as per your requirements.

Sample NodeJS program to fetch trades:

```
const crypto = require('crypto');
let time = new Date().getTime();

const CLIENT_KEY = "a64cdc31716649d4c8fd79b89ab965d8";
const CLIENT_SECRET =
  ↪ "d37e9c4f137601f2c59b796a033a99a35e20a6757e754f30d00cff9c438b0cac";

let baseUrl = "https://coinmax.com.au/api/client"
let apiURL = `/trades?timestamp=${time}&page=0&symbol=ETH-AUD`;
let reqURL = `${baseUrl}${apiURL}`;

var sign = crypto.createHmac('sha256', CLIENT_SECRET).update(apiURL).digest(
  'hex')

console.log(sign);

var request = require("request");

var options = {
  url: reqURL,
  headers: {
    'X-API-KEY': CLIENT_KEY,
    'X-API-SIGNATURE': sign
  },
  method: 'get',
  json: true
};

request(options, (error, res, body) => {
  if (error) {
    console.warn(error);
  }
  console.log(body);
})
```



## CHAPTER 3

---

### WebSocket

---

You can connect to our WebSocket to get updates on your orders and receive market data events in real time. This reduces the amount of data transfer and latency as you no longer have to poll our servers for updates.

To connect with WebSocket send the following request:

```
GET wss://coinmax.com.au/ws?X-API-KEY={API_KEY}&timestamp=1544020774432&X-API-  
→SIGNATURE={SIGNATURE}  
Connection: Upgrade  
Upgrade: websocket
```

---

**Note:** API\_KEY can be obtained by *Generate Credentials* and SIGNATURE is the hex value of sha256 of API key + timestamp with Client Secret

---

Sample NodeJS program:

```
var ws = require('ws');  
var crypto = require('crypto');  
let time = new Date().getTime();  
  
const CLIENT_KEY = "dec5b6dad847f157a51735d34fd09e79";  
const CLIENT_SECRET =  
→ "1f9bcfaa4542676463f953224fdbf779e92fbc9898aa56146797a152097182fd";  
  
var sign = crypto.createHmac('sha256', CLIENT_SECRET).update(CLIENT_KEY + time).  
→ digest(  
    'hex')  
  
const coinmaxWebsocket = new ws(`wss://coinmax.com.au/ws?X-API-KEY=${CLIENT_KEY}&  
→ timestamp=${time}&X-API-SIGNATURE=${sign}`);  
coinmaxWebsocket.on('open', function open() {  
    coinmaxWebsocket.send(JSON.stringify({  
        "action": "subscribe",  
        "channels": [{
```

(continues on next page)

(continued from previous page)

```

                "name": "ohlc",
                "productIds": ["ETH-AUD"]
            },
            {
                "name": "trades"
            }
        ]],
        "productIds": ["ZEC-AUD"]
    }));
});

coinmaxWebsocket.on('message', function incoming(data) {
    console.log(data);
});

```

### 3.1 Subscriptions

Once you are connected, you will need to subscribe to the events published by WebSocket.

---

**Note:** All communications with WebSocket are in JSON format.

---

Subscriptions request format:

```

{
  "action": "subscribe",
  "channels": [{
    "name": "ohlc",
    "productIds": ["ETH-AUD"]
  },
  {
    "name": "quoteIncremental"
  }
  ]],
  "productIds": ["ZEC-AUD"]
}

```

**Attention:** In the above structure, ohlc channel will subscribe to ETH-AUD and ZEC-AUD both, whereas trades will only subscribe to ZEC-AUD

Subscriptions reply format:

```

{
  "Type": "subscriptions",
  "subscribe": "ok",
  "subscriptions": [
    {
      "name": "ohlc",
      "productIds": ["ETH-AUD", "ZEC-AUD"]
    },
    {
      "name": "quoteIncremental",
      "productIds": ["ZEC-AUD"]
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    }
  ]
}

```

Subscriptions response includes a *subscribe* field which will be either **ok** or **fail**. It will fail if your request is invalid. An invalid request means you either sent an empty request or all channels in your request were invalid, check *reason* field for more details. If any channel in the request is valid *subscribe* field will be **ok**.

Response will also include a *subscriptions* array which includes responses to each valid channel. If you try to register to an invalid channel, subscriptions array will not include it in the response as we don't entertain invalid requests. But if for some reason your subscription request for a valid channel fails, there will be an entry for it in the subscription array with error *code* and *description*.

---

**Hint:** Every message you will receive from WebSocket will include a **Type** field, You can use it to decide what actions to perform

---

### Subscribe to events

To Subscribe to an event, send a message to WebSocket with `action: subscribe` and the channel and product id you want to subscribe for. A complete list of [Market Data Channels](#), [Authorized Client Data Channels](#) is listed below.

### Unsubscribe from events

Similarly, to Unsubscribe from an event, repeat the message you sent during *subscribe* and change from `action: subscribe` to `action: unsubscribe`

## 3.1.1 Market Data Channels

Following channels are available to listen for market data.

Table 1: Market Data Channels

Data	Channel Name	Supported Product IDs
OHLC	ohlc	All supported IDs
24H Changes	ticker	* <i>Asterisk</i>
Real time market trades	tradeQuote	All supported IDs
Market trades historical data	tradeQuoteBatch	All supported IDs
Depth Snapshot	quoteFullSnapshot	All supported IDs
Depth Incremental updates	quoteIncremental	All supported IDs

## 3.1.2 Authorized Client Data Channels

Following channels are available to listen for real time updates.

Table 2: Authorized Channels

Data	Channel Name	Supported Product IDs
Order Update	orderUpdate	All supported IDs
Asset Deposits	deposit	* <i>Asterisk</i>
Asset Withdrawals	withdrawal	* <i>Asterisk</i>
Cash Components	funds	* <i>Asterisk</i>

## 3.2 Data Structures

### 3.2.1 Market data

#### OHLC

```
{
  "Type": "ohlc",
  "Candles":
  [
    {
      "Asteriskt": Epoch,
      "Interval": Interval in seconds,
      "O": string,
      "H": string,
      "L": string,
      "C": string,
      "V": string
    }...
  ],
  "Symbol": string
}
```

---

**Important:** Candles array will include candles with various time intervals, for e.g. 300, 900, 3600, 14400, 86400

---

#### 24 Hour Changes (Ticker)

```
{
  "Type": "ticker",
  "Tickers":
  [
    {
      "Symbol": "string",
      "PercentChange": string,
      "High": string,
      "Low": string,
      "Ltp": string,
      "Volume": string,
      "Timestamp": Epoch
    }...
  ]
}
```

#### Market Trades

```
{
  "Type": "tradeQuote",
  "Symbol": string,
  "Taker": string,
  "Ltq": string,
```

(continues on next page)

(continued from previous page)

```

"Ltp": string,
"Timestamp": string
}

```

---

**Note:** Possible values for Taker field are maker and taker

---

## Depth Book

Full Snapshot:

```

{
  "Type": "quoteFullSnapshot",
  "Symbol": "string",
  "NoOfBuyLevels": "Number",
  "Buys": [
    {
      "Price": "5741.83",
      "AggrQty": "0.00000993",
      "OrderCount": 1
    }
  ],
  "NoOfSellLevels": "Number",
  "Sells": [
    {
      "Price": "6561.54",
      "AggrQty": "0.00013083",
      "OrderCount": 1
    }
  ],
  "SeqNo": "Number",
  "Timestamp": "Number"
}

```

---

**Note:** Buys and Sells are maps with Price as Key and {AggrQty and OrderCount} Object as value

---

Incremental Update:

```

{
  "Type": "quoteIncremental",
  "Symbol": string,
  "SeqNo": Number,
  "Side": string,
  "Price": string,
  "AggrQty": string,
  "OrderCount": Number,
  "Action": Number
  "Timestamp" : Number
}

```

Table 3: Actions map

Action	Description
0	Insert
1	Update
2	Delete

Depth book updates as well as snapshot both provide a `SeqNo` which you can use to make sure you do not miss any update. If you miss an update you can request a full snapshot by sending the below request to `WebSocket`.

```
{
  "action": "dataFetch",
  "channels": [
    {
      "name": "quoteFullSnapshot"
    }
  ],
  "productIds": ["<Your product Id>"]
}
```

## 3.2.2 Authorized Client Data

### Cash Component

```
{
  "Type": "funds",
  "CashComponents": [{
    "Asset": string,
    "AvailableUnits": string,
    "PendingUnits": string,
    "SeqNo": uint32
  }
  ]
}
```

### Order Update

```
{
  "Type": "orderUpdate",
  "Symbol": string,
  "OrderStatus": string,
  "OrderType": string,
  "OrderValidity": string,
  "Side": string,
  "ExecutionId": string,
  "ExchangeOrderId": string,
  "Qty": string,
  "Price": string,
  "PassiveUnits": string,
  "FilledQty": string,
  "LastFillQty": string,
  "LastFillPrice": string,
}
```

(continues on next page)

(continued from previous page)

```

"Timestamp": string,
"Commission": string,
"ClientOrderId": string,
"SeqNo": uint32,
"Error": string
}

```

Table 4: Possible values for OrderStatus

Status	Description
CANCELLED	Your order was cancelled, Check Error for more details
REJECTED	Order was rejected, Check Error for more details
ACCEPTED	Your order is accepted by Exchange
PARTIAL_FILLED	Your order was filled partially, LastFillQty and LastFillPrice will contain the Quantity Filled and at what Price for this trade
FILLED	Your order is completely filled
CANCEL_REJECTED	Your request for cancelling your order was rejected, Check Error for more details

**Important notes**

- You may receive PARTIAL\_FILLED events multiple times
- Error field will not be available for PARTIAL\_FILLED and FILLED orders
- ExecutionId will be only available for PARTIAL\_FILLED and FILLED orders

---

**Hint:** ExecutionId + ExchangeOrderId = Unique Trade Identifier

---

**Asset Deposit**

```

{
  "Type": "deposit",
  "Asset": string,
  "Amount": string,
  "TxStatus": string,
  "TxId": string,
  "NodeTxId": string,
  "Timestamp": string,
  "Fee": string,
  "Error": string,
  "PaymentAgent": "string",
  "Metadata": {
    "code": string
  }
}

```

---

**Hint:** You can use NodeTxId to view the transaction on block explorer.

---

```
{
  "Type": "withdrawal",
  "Asset": string,
  "Amount": string,
  "ApprovedAmount": string,
  "Fee": string,
  "TxStatus": string,
  "TxId": string,
  "ReferenceId": string,
  "NodeTxId": string,
  "Timestamp": string,
  "Recipient": string,
  "Error": string
}
```

---

**Hint:** You can use `NodeTxId` to view the transaction on block explorer.

---

---

**Note:** Use the `ReferenceId` to uniquely identify withdrawal request.

---

## CHAPTER 4

---

### Api Keys

---

Every generated api key has provisions attached to it, based on whom you may be allowed to access a service or not. Following are the service types and provisions available for a key as of now.

Table 1: Api Key Services & Provisions

Service Name	Provision Type
Trading	ENABLED(default) / DISABLED
Withdrawal	ENABLED / DISABLED(default)

Please contact support if you want to change them.



## CHAPTER 5

---

### Fees

---

We calculate fees as a fraction of the notional value of each trade (i.e., price  $\times$  amount). Any fees will be applied at the time an order is placed. For partially filled orders, only the executed portion is subject to trading fees. Fees we charge for Maker trades is 0.10% while those for Taker trades is 0.15%. If any other users have joined by using your referral link, you gain 0.10% of the fees that we charge them on their trades.



## CHAPTER 6

---

### Disclaimer

---

Please note that this is a beta version of the Coinmax API. The API is provided on an “as is” and “as available” basis. Coinmax does not give any warranties, whether express or implied, as to the suitability or usability of the API, its software or any of its content. Coinmax will not be liable for any loss, whether such loss is direct, indirect, special or consequential, suffered by any party as a result of their use of the Coinmax API. Any use of the API is done at the user’s own risk and the user will be solely responsible for any damage to any computer system or loss of data that results from such activities. Should you encounter any bugs, glitches, lack of functionality or other problems on the website, please let us know immediately so we can rectify these accordingly. Your help in this regard is greatly appreciated.