

# WebSocket API

This article explains how to use the WebSocket API provided by the Fusion SDK to receive real-time updates about order events and to query active orders. The WebSocket API enables developers to build applications that can respond instantly to order creation, filling, and cancellation events.

## Real world example

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

wsSdk.order.onOrder((data) => {
  console.log("received order event", data);
});
```

## Creation

With constructor:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});
```

Custom provider:

User can provide custom provider for websocket (by default we are using [ws library](#))

```
import { WsProviderConnector, WebSocketApi } from "@1inch/fusion-sdk";

class MyFancyProvider implements WsProviderConnector {
  // ... user implementation
}

const url = "wss://api.1inch.dev/fusion/ws/v1.0/1";
const provider = new MyFancyProvider({ url });

const wsSdk = new WebSocketApi(provider);
```

With new static method:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = WebSocketApi.new({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});
```

Lazy initialization:

By default when user creates an instance of `WebSocketApi`, it automatically opens websocket connection which might be a problem for some use cases

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  lazyInit: true,
  authKey: "YOU-API-KEY",
});

ws.init();
```

## Methods

Base methods

on

Description: You can subscribe to any event

Arguments:

- [0] event: string
- [1] cb: Function

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.on("error", console.error);

ws.on("open", function open() {
  ws.send("something");
});

ws.on("message", function message(data) {
  console.log("received: %s", data);
});
```

## off

Description: You can unsubscribe from any event

Arguments:

- [0] event: string
- [1] cb: Function

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.on("error", console.error);

ws.on("open", function open() {
  ws.send("something");
});

function message(data) {
  console.log("received: %s", data);
}

ws.on("message", message);

ws.off("message", message);
```

# onOpen

Description: subscribe to open event

Arguments:

- [0] cb: Function

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.onOpen(() => {
  console.log("connection is opened");
});
```

# send

Description: send event to backend

Arguments:

- [0] message: any message which can be serialized with JSON.stringify

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.send("my message");
```

# close

Description: close connection

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
```

```
url: "wss://api.1inch.dev/fusion/ws",  
network: NetworkEnum.ETHEREUM,  
authKey: "YOU-API-KEY",  
});  
  
ws.close();
```

## onMessage

Description: subscribe to message event

Arguments:

- [0] cb: (data: any) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";  
  
const ws = new WebSocketApi({  
  url: "wss://api.1inch.dev/fusion/ws",  
  network: NetworkEnum.ETHEREUM,  
  authKey: "YOU-API-KEY",  
});  
  
ws.onMessage((data) => {  
  console.log("message received", data);  
});
```

## onClose

Description: subscribe to close event

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";  
  
const ws = new WebSocketApi({  
  url: "wss://api.1inch.dev/fusion/ws",  
  network: NetworkEnum.ETHEREUM,  
  authKey: "YOU-API-KEY",  
});  
  
ws.onClose(() => {  
  console.log("connection is closed");  
});
```

## onError

Description: subscribe to error event

Arguments:

- [0] cb: (error: any) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.onError((error) => {
  console.log("error is received", error);
});
```

Order namespace

## onOrder

Description: subscribe to order events

Arguments:

- [0] cb: (data: OrderEventType) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.order.onOrder((data) => {
  if (data.event === "order_created") {
    // do something
  }
  if (data.event === "order_invalid") {
    // do something
  }
});
```

## onOrderCreated

Description: subscribe to order\_created events

Arguments:

- [0] cb: (data: OrderCreatedEvent) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.order.onOrderCreated((data) => {
  // do something
});
```

## onOrderInvalid

Description: subscribe to order\_invalid events

Arguments:

- [0] cb: (data: OrderInvalidEvent) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.order.onOrderInvalid((data) => {
  // do something
});
```

## onOrderBalanceOrAllowanceChange

Description: subscribe to order\_balance\_or\_allowance\_change events

Arguments:

- [0] cb: (data: OrderBalanceOrAllowanceChangeEvent) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
```

```

url: "wss://api.1inch.dev/fusion/ws",
network: NetworkEnum.ETHEREUM,
authKey: "YOU-API-KEY",
});

ws.order.onOrderBalanceOrAllowanceChange((data) => {
  // do something
});

```

## onOrderFilled

Description: subscribe to order\_filled events

Arguments:

- [0] cb: (data: OrderFilledEvent) => void

Example:

```

import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.order.onOrderFilled((data) => {
  // do something
});

```

## onOrderFilledPartially

Description: subscribe to order\_filled\_partially events

Arguments:

- [0] cb: (data: OrderFilledPartiallyEvent) => void

Example:

```

import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.order.onOrderFilledPartially((data) => {

```



```
// do something  
});
```

Rpc namespace

## onPong

Description: subscribe to ping response

Arguments:

- [0] cb: (data: string) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";  
  
const ws = new WebSocketApi({  
  url: "wss://api.1inch.dev/fusion/ws",  
  network: NetworkEnum.ETHEREUM,  
  authKey: "YOU-API-KEY",  
});  
  
ws.rpc.onPong((data) => {  
  // do something  
});
```

## ping

Description: ping healthcheck

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";  
  
const ws = new WebSocketApi({  
  url: "wss://api.1inch.dev/fusion/ws",  
  network: NetworkEnum.ETHEREUM,  
  authKey: "YOU-API-KEY",  
});  
  
ws.rpc.ping();
```

## getAllowedMethods

Description: get the list of allowed methods

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.rpc.getAllowedMethods();
```

## onGetAllowedMethods

Description: subscribe to get allowed methods response

Arguments:

- [0] cb: (data: RpcMethod[]) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.rpc.onGetAllowedMethods((data) => {
  // do something
});
```

## getActiveOrders

Description: get the list of active orders

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.rpc.getActiveOrders();
```

## onGetActiveOrders

Description: subscribe to get active orders events

Arguments:

- [0] cb: (data: PaginationOutput<ActiveOrder>) => void

Example:

```
import { WebSocketApi, NetworkEnum } from "@1inch/fusion-sdk";

const ws = new WebSocketApi({
  url: "wss://api.1inch.dev/fusion/ws",
  network: NetworkEnum.ETHEREUM,
  authKey: "YOU-API-KEY",
});

ws.rpc.onGetActiveOrders((data) => {
  // do something
});
```

## Types

### OrderEventType

```
type Event<K extends string, T> = { event: K; data: T };
```

```
type OrderEventType =
  | OrderCreatedEvent
  | OrderInvalidEvent
  | OrderBalanceOrAllowanceChangeEvent
  | OrderFilledEvent
  | OrderFilledPartiallyEvent;
```

```
type OrderCreatedEvent = Event<
  "order_created",
  {
    orderHash: string;
    signature: string;
    order: LimitOrderV3Struct;
    deadline: string;
    auctionStartDate: string;
    auctionEndDate: string;
    remainingMakerAmount: string;
  }
>;
```

```
type OrderBalanceOrAllowanceChangeEvent = Event<
  "order_balance_or_allowance_change",
  {
    orderHash: string;
    remainingMakerAmount: string;
    balance: string;
    allowance: string;
```

```
}  
>;  
  
type OrderInvalidEvent = Event<  
  "order_invalid",  
  {  
    orderHash: string;  
  }  
>;  
  
type OrderFilledEvent = Event<"order_filled", { orderHash: string }>;  
  
type OrderFilledPartiallyEvent = Event<  
  "order_filled_partially",  
  { orderHash: string; remainingMakerAmount: string }  
>;
```

## RpcMethod

```
type RpcMethod = "getAllowedMethods" | "ping";
```

Previous

[< Auction Suffix](#)

Next

[Introduction >](#)