OpenAPI definition

Overview

Tags

Market Data Replay

APIs for streaming pre-recorded market data. Provides APIs to create session, start, stop, rewind, jump, forward, and set replay speed.

Paths

PUT `/mktdata/session/stop/{sessionId}` Stop replay session.

Stop streaming events from replay session. Current state is saved. Call start to resume streaming from saved state. It is a no-op for stopped sessions.

Parameters

Type	Name	Description	Schema
path	**sessionId** *required*	Session Id (UUID)	string

Responses

Code	Description	Links
400	Bad request *Content* /	No Links
200	Successfully stopped *Content* /	No Links

PUT `/mktdata/session/start/{sessionId}` Start replay session.

Start streaming events from replay session. A stopped replay session will start from previous saved state. It is a no-op for running sessions.

Parameters

Type	Name	Description	Schema
path	**sessionId** *required*	Session Id (UUID)	string

Responses

Code	Description	Links
400	Bad request *Content* /	No Links
200	Successfully started *Content* /	No Links

PUT `/mktdata/session/speed/{sessionId}/{speed}` Set replay speed.

Set replay session speed. Default speed is 1.0 (real-time). Value less than 1.0 will slow down replay. Value greater than 1.0 will speed up replay.

Parameters

Type	Name	Description	Schema
path	**sessionId** *required*	Session Id (UUID)	string
path	**speed** *required*	Must be positive (> 0.0). Default is 1.0. Value less than 1.0 will slow down replay. Value greater than 1.0 will speed up replay.	number (double)

Responses

Code	Description	Links
400	Bad request *Content* /	No Links
200	Successfully set replay speed *Content* /	No Links

PUT `/mktdata/session/rewind/{sessionId}` Rewind session to beginning.

Rewind session to beginning. Running replay sessions will resume streaming from beginning.

Parameters

Type	Name	Description	Schema
path	**sessionId** *required*	Session Id (UUID)	string

Responses

Code	Description	Links
400	Bad request *Content* /	No Links
200	Successfully rewound *Content* /	No Links

PUT `/mktdata/session/jump/{sessionId}/{eventId}` Jump to session event.

Jump to specific event in the replay session. Running sessions will resume streaming from specified event.

Parameters

Type	Name	Description	Schema
path	**eventId** *required*	Event Id. Must be positive (> 0)	integer (int32)
path	**sessionId** *required*	Session Id (UUID)	string

Responses

Code	Description	Links
400	Bad request *Content* /	No Links

Code	Description	Links
200	Successfully jumped to event *Content* /	No Links

PUT `/mktdata/session/forward/{sessionId}/{skipCount}` Forward replay session by number of events.

Jump forward by skipping specified number of events. If number of skip events is greater than remaining events, session will stop (does not wrap around).

Parameters

Type	Name	Description	Schema
path	**sessionId** *required*	Session Id (UUID)	string
path	**skipCount** *required*	Number of events to skip. Must be positive (> 0)	integer (int32)

Responses

Code	Description	Links
400	Bad request *Content* /	No Links
200	Successfully forwarded session *Content* /	No Links

POST `/mktdata/session` Create replay session

Create new replay session. Make sure to subscribe to session by calling stream.

Responses

Code	Description	Links
200	Successfully created session *Content* /	No Links

GET `/mktdata/session/subscribe/{sessionId}` Subscribe to replay session events.

Subscribe to replay session events. After subscription, replay session must be started to actually begin streaming.

Parameters

Type	Name	Description	Schema
path	**sessionId** *required*	Session Id (UUID)	string

Responses

Code	Description	Links
400	Bad request *Content* **text/event-stream**	No Links
200	Successfully subscribed *Content* **text/event-stream**	No Links

Components

Schemas

MarketDataEvent

Properties

Name	Description	Schema
id *optional*		integer (int32)
timestamp *optional*		integer (int64)
event *optional*		string
price1 *optional*		number (double)
shares1 *optional*		integer (int32)
xchg1 *optional*		string

Name	Description	Schema
price2 *optional*		number (double)
shares2 *optional*		integer (int32)
xchg2 *optional*		string