IG TRADING API – CODEX CHEAT SHEET
==================================

BASE URLS

Demo: https://demo-api.ig.com/gateway/deal
Live: https://api.ig.com/gateway/deal

COMMON HEADERS

X-IG-API-KEY: <your_api_key>
CST: <client_session_token>   # v1/v2 sessions
X-SECURITY-TOKEN: <account_session_token>
Authorization: Bearer <oauth_access_token>   # v3 sessions
IG-ACCOUNT-ID: <account_id>   # when using OAuth
Content-Type: application/json
Accept: application/json; charset=UTF-8
Version: <integer_api_version>   # defaults to 1 if omitted

AUTHENTICATION / SESSION

POST /session   # create session, returns CST/X-SECURITY-TOKEN or OAuth tokens
GET /session   # get session details; can request CST/X-SECURITY-TOKEN when using OAuth
PUT /session   # switch active account, optionally set default
DELETE /session   # log out and invalidate tokens
GET /session/encryptionKey # get AES key + timestamp for encrypting password before login
POST /session/refresh-token # refresh OAuth access token and get new refresh token

ACCOUNT

GET /accounts   # list all accounts for logged-in user
GET /accounts/preferences   # get preferences for active account
PUT /accounts/preferences   # update preferences for active account

Activity & transaction history
GET /history/activity   # activity history (latest, supports paging/filters)
GET /history/activity/{fromDate}/{toDate}   # activity history for date range
GET /history/activity/{lastPeriod}   # activity history for relative period (e.g. 1D, 1W)
GET /history/transactions   # transaction history (default recent window)
GET /history/transactions/{type}/{from}/{to}   # transactions by type and date range
GET /history/transactions/{type}/{lastPeriod}   # transactions by type and recent period

DEALING (POSITIONS & WORKING ORDERS)

Deal confirmations
GET /confirms/{dealReference}   # get asynchronous deal result by reference

Positions
GET   /positions   # list all open positions on active account
GET   /positions/{dealId}   # get details for specific open position
POST   /positions/otc   # open OTC position (create trade)
DELETE /positions/otc   # close one or more OTC positions
PUT   /positions/otc/{dealId}   # update existing OTC position

Working orders
GET   /working-orders   # list all open working orders on active
account
POST   /working-orders/otc   # create OTC working order
DELETE /working-orders/otc/{dealId}   # delete working order
PUT   /working-orders/otc/{dealId}   # update working order

Repeat dealing
GET /repeat-dealing-window   # get repeat-dealing window status for account

MARKETS & PRICES

GET /categories   # list top-level instrument categories
available
GET /categories/{categoryId}/instruments   # list instruments in a given category

(Deprecated navigation – prefer categories)
GET /market-navigation   # list top-level nodes in navigation tree
GET /market-navigation/{nodeId}   # list child nodes for a given tree node

Market details
GET /markets   # get details for multiple markets (by epics
query)
GET /markets/{epic}   # get details for a single market (multiple
versions)
GET /markets?searchTerm={term}   # search markets by text

Historical prices
GET /prices/{epic}   # default recent prices (e.g. last minutes)
GET /prices/{epic}/{resolution}/{numPoints}   # OHLCV series for N points
GET /prices/{epic}/{resolution}/{start}/{end}   # OHLCV series for explicit date range
GET /prices/{epic}/{resolution}?startdate=…&enddate=…
                        # same as above with query params

WATCHLISTS

```
GET   /watchlists                    # list all watchlists on active account
POST  /watchlists                     # create a new watchlist
GET   /watchlists/{watchlistId}         # get specific watchlist and its markets
PUT   /watchlists/{watchlistId}         # add a market to watchlist (via epic)
DELETE /watchlists/{watchlistId}          # delete entire watchlist
DELETE /watchlists/{watchlistId}/{epic}       # remove a market from watchlist
```

CLIENT SENTIMENT

```
GET /client-sentiment                 # sentiment for one or more markets
GET /client-sentiment/{marketId}         # sentiment for given market
GET /client-sentiment/related/{marketId}     # related markets + sentiment
```

INDICATIVE COSTS & CHARGES

```
POST /indicativecostsandcharges/open       # estimate costs/charges for opening order
POST /indicativecostsandcharges/close       # estimate costs/charges for closing
POST /indicativecostsandcharges/edit        # estimate costs/charges for editing order
GET  /indicativecostsandcharges/history/from/{from}/to/{to}
                               # costs/charges history over date range
GET  /indicativecostsandcharges/durablemedium/{quoteRef}
                               # download generated quote as PDF
```

GENERAL / APPLICATION MANAGEMENT

```
GET /operations/application               # list apps linked to this account/API key
PUT /operations/application               # update application details/limits
PUT /operations/application/disable         # disable current API key (can re-enable via
web UI)
```

STREAMING API – CORE PATTERNS (LIGHTSTREAMER)
===

CONNECTION

Lightstreamer endpoint: provided in REST /session response
User (identifier): active IG account ID
Password: "CST-<CST>|XST-<X-SECURITY-TOKEN>"   # built from REST session tokens

NOTE: Streaming currently requires CST/X-SECURITY-TOKEN even if you use OAuth for REST.
     Use GET /session?fetchSessionTokens=true to obtain them when on OAuth.

PRICE STREAMS

Primary price subscription (recommended)
Item: PRICE:{accountId}:{epic}
Mode: MERGE
Fields (subset): BIDPRICE1-5, ASKPRICE1-5, BIDSIZE1-5, ASKSIZE1-5,
          HIGH, LOW, MID_OPEN, TIMESTAMP, CURRENCY*, etc.

Deprecated market subscription (older pattern)
Item: MARKET:{epic}
Mode: MERGE
Fields (subset): BID, OFFER, MID_OPEN, HIGH, LOW, CHANGE, CHANGE_PCT,
          MARKET_STATE, MARKET_DELAY, UPDATE_TIME, etc.

ACCOUNT STREAMS

Item: ACCOUNT:{accountId}
Mode: MERGE
Fields (subset): PNL, FUNDS, MARGIN, AVAILABLE_TO_DEAL,
          EQUITY, EQUITY_USED, DEPOSIT, etc.

TRADE STREAMS

Item: TRADE:{accountId}
Mode: DISTINCT

Key sub-objects:
- CONFIRMS # trade confirmations
- OPU   # open position updates
- WOU   # working order updates

Fields across these include:
direction (BUY/SELL), epic, size, level, limitLevel, stopLevel,
dealId, dealReference, status, dealStatus, expiry, currency,
guaranteedStop, trailingStopDistance, trailingStep, etc.

CHART (TICK) STREAMS

Item: CHART:{epic}:TICK
Mode: DISTINCT
Fields (subset): BID, OFR, LTP, LTV, TTV, UTM,
          DAY_OPEN_MID, DAY_NET_CHG_MID, DAY_PERC_CHG_MID,
          DAY_HIGH, DAY_LOW

CHART (CANDLE) STREAMS

```
----------------------
Item:  CHART:{epic}:{scale}
Mode:  MERGE
scale: SECOND, 1MINUTE, 5MINUTE, HOUR

Fields (subset):
LTV, TTV, UTM,
DAY_OPEN_MID, DAY_NET_CHG_MID, DAY_PERC_CHG_MID,
DAY_HIGH, DAY_LOW,
OFR_OPEN/HIGH/LOW/CLOSE,
BID_OPEN/HIGH/LOW/CLOSE,
LTP_OPEN/HIGH/LOW/CLOSE,
CONS_END (1 when candle ends),
CONS_TICK_COUNT (number of ticks in candle)

TYPICAL WORKFLOW (HIGH LEVEL)
-----------------------------
1. POST /session with API key + credentials to obtain CST/X-SECURITY-TOKEN (or OAuth tokens).
2. For OAuth: use Authorization: Bearer <access_token> and IG-ACCOUNT-ID on REST calls.
3. For streaming: obtain Lightstreamer endpoint from /session, then connect using
   accountId + "CST-...|XST-..." as password.
4. Use REST for one-off operations (search markets, place trades, query history, etc.).
5. Use Streaming for live prices, account updates, and trade confirmations.
```