INTERACTIVE BROKERS

CP API for Institutions

User Documentation

Version 5.0 – July 14, 2022

Contents

CP API

1. Introduction

The purpose is to provide third parties with a web-based interface that supports:

- User authentication.
- Market data subscriptions.
- Chart subscription.
- All additional functionality supported on CP Web API as a proxy only

2. Implementation

This API uses window.postMessage that allows third party to get data from IB Server.

- Third party website loads a page from IB site inside an iframe
- The API implements a bi-directional postMessage cross-domain communication
- The postMessage message source must be validated on both sides

Requirements:

- Browser must support window.postMessage
- Institution application must be implemented as single page application.
- Websocket support

3. Supported Calls and their Responses

The communication between the iframe and the parent is made using PostMessage messages.

Both incoming and outgoing messages are JSON objects. The data JSON Object will contain ACTION property that identifies which package should handle the message. We use the following ACTION Values: MD for market data, and CHART for chart data. The output package could also contain error codes in an ERRORS property (array type).

Examples are provided with some of the calls in later sections.

4. Third party calls to IB

Prerequisites:

Each Institution needs to:

- Have a white branded ID (string) – that is validated and known to IB.
- Know how long the server should keep the session (minutes, integer between 1 and 30).
- Have an iframe in the DOM (to display the login form).
- Have a selected language for the login form (available: en, cn, de, es, fr, it, jp, nl, ru, tw).
- Have Client version: 1.3 or 1.4 (if you need additional support for orders/trades).

Requests

Third party makes a call to IB API, and that API invokes call to backend that generates response that is sent back to caller.

The sample implementation of the communication of third party with IB server is by using the module wtapi.js (look in appendix for the location of file). This module wtapi.js can be accessed with the methods of window.WTAPI object. This module is standalone, does not require any additional library to work.

The communication is asynchronous and implemented with PostMessage messages.

4.1 Initialization

1. Load the IB page with GET parameters from the IB server to the iframe. The parameters are the following:

WB_ID <string> : white branded ID

lang<string> : (optional) display language of the login form Ex:

https://api.ibkr.com/sso/cpapi/?lang=en&WB_ID=Asterix&CLIENT_VERSION=1.4

Once the iframe loads, it sends a message to parent:

{ method: 'loaded'}

If IB server is down or getting restarted, no response is sent back to parent.

2. Next an init call is made to iframe:

{ method: 'init', wb_id: <string>, max_timeout: <number>, client_version_id: <number> }

Response:

{ method: 'initialized'}

If the INIT call is made successfully, the frame URL and whitebranded ID are validated and the login page within the iframe is shown.

If INIT is not successful, one of the following messages is sent back to parent:
{ method: 'message', message: 'Initialization failed', ERRORS: <Array> }

Potential Error Codes
WTE099
WTE018
WTE019
WTE023

4.2 Authentication

Authentication is done within the iframe. If login succeeds, the following messages are sent back to the parent. 2 Signature messages are sent along with accounts message.

Response sent to parent:

{ method: 'message', message: 'Logged in' }

{ method: 'setPing', timeout: 60000 }

{ method: 'data', message:

{"BACKEND_SESSION_ID":"581c0b12.00000291","ERRORS":[],"ACCOUNTS":"DU91474","MESSAGE_TYPE":"FORWARD","ID":"d53d96e0-0095-4ef0-bc33-2b7117d6ca03","USER":"qapap908","CURRENT_TIME":1478294565714}

{"SIGNATURE_TYPE":"USER_SESSION","MESSAGE_TYPE":"FORWARD","ID":"":"d53d96e0-0095-4ef0-bc33-2b7117d6ca03","USER":"qapap908","RESULT":true,"SIGNATURE":"OBi7YFp3lBY1d+jMvGg7NwJtsJpNVYqyD67WbQR+wAfRzHvwLVGTEd3EeIaITw3tOtTMzzKuQR4/BJKp1qcLryGPHo5r3sJcgnNEXOqal9g8P/plNpjkgLVaVd0uDgp0aiis+EXW2Z38LKhOEP/3vdGvR4Lvin9jjtmEts724aUnMfVg9ZiNK/dbWd1ADgXgid/f1fkRekDDbDmGphYaQ3jEv0bSkCfg6d8rc/jW+gDAiBGMbPnu9Jxvx1vjbT4vkGPL9PkcpgEJHgFKDXbw6IlNDJ6sHvX41WXZuv/6o49DpgXyITQQhgPwIf7HbKs5uAA4IJSoNEfHVB+sHyv/hnONZIE1vNPjwewtRC84Sob1y7pblYY3eUusfzy756y63qTVBfOoLynUEj12pihs6zvghZd26kMGE1TexO+hs4EkepY1Fnq5jzR0IA5cp0dBXpehGiXbdXp1FwEFFDjWlNUlDpLSgIwcy4SKy6xstPviiasZktlUKbQP1ruTXw4ANcUXLcdggbhABhXWjqsFz8WN9M65hHTCx2CpM3gZn2uLRXz4gg2aRUL4ilXlgMRKi2ILDgWSxVn7R80Q1lhRalt1jmmUBjNY7pQJvGsSOQ3+66i4hiedctWcVugDHsF3IsxVgrp04PmIUjd4XNYKfuhrFsJeQVjCYHAfxduC+bo="}

{"SIGNATURE_TYPE":"USER_ACCOUNTS","MESSAGE_TYPE":"FORWARD","ID":"":"d53d96e0-0095-4ef0-bc33-2b7117d6ca03","USER":"qapap908","RESULT":true,"SIGNATURE":"REJSCdTG23xmiPOgACV8uyoeq3vIiTnGo/3EygQ/C3R1FAiXn3e59tAEm6FER15LikWjNw9IXLcsEdiRsKbc+MHZ6iFsU7UrPY/ZORdWL0KkXEqu1swo2ZE4KaT3O0Dujnh+GFGSQcNdYP2I7lSWXS5442Bub06pShXuGOmC1Dd9FdmwKWCrR1kTwm9RsylhLB32OHaNtdaxMYgULPbDH8WVUjAYheRZ389F9yHoSqSRlzSQUdZIch66/747vABMuFzaAYqxxbM0LL0k1fpXpifuFAAqOTMjwtZB6HmSbpgwH45NRD5b7WRrwyFcm8HLpa2P2isc6lCk/AxY42Ay8IQUvluFzSmE3N+4vCbiWf9IiB2jcovZoZb0J/twm2E98J9ryP76IvgdULdMBDciy7J728dSoHr0SOPmBkjGWHrF8sLrKpVFm6YJ3t6UrOhMboOiPIIUe/m/e3WAUn48NOYaOpwiKjjkmBfi5H05Z+UtMMLZErt7w4vRdgbfEb+IgsLX75HBSUwcOhyHP0/Tm/DljFRKNYhTJ2yPE+oKqiC2M8iP8XNChurxFQKqIgT2CaIqCRSAd48C6hBEZqacX4CUGbeKDaptnRwkaIopzlqIQ+NWVLLlPulV3aIOS81W8+oq73dNa6/aXnR8fEoIKk9acvM7Sjy1NbSPumnYw+Q="}

If user authentication fails, user cannot proceed beyond login page.

4.3 Ping

A ping message needs to be sent to the iframe every 1 minute (defined as timeout below). The ping should start after a successful login and stops when the logout() method is called.

If ping stops and the max_timeout defined in Section 4.1 is reached, the server will clean out the session and no market data or chart data will be sent from the server.

After successful login, the iframe sends a ping message as below:

```
{ method: 'setPing', timeout: 60000 }
```

After that the parent needs to send a ping message to iframe, at the timeout interval defined above.

The parent frame has to send the following message periodically:

```
{ method: 'ping' }
```

4.4 Competition

If the user is logged into any other IB trading application like TWS or CP or with the same username and tries to login into CP API, there will be a competition message. This competition callback function is called with a message 'competition'. The callback function should display a popup message (with the native confirm or a custom one like jQuery Dialog) asking the user to continue the session or stop it.

Competition message:
{ method: 'competition', message: <string> }

If the user wants to continue the current session, the app should call:
{ method: 'keepSession' }
To kill the current session, call the method:
{ method: 'logout' }

4.5 Market Data Subscribe

To subscribe to market data the application should call the 'mdSubscribe' method with a conid-exchange-pairs parameter. This parameter can contain one or more conid-exchange pairs separated by a semicolon character. Ex: 8314:SMART;43645865:SMART

Input below:

{ method: 'mdSubscribe', md: <string> }

After a successful subscribe the data callback function will be called. For market data, MD packages contain static data describing the contract properties, prices, and other market data attributes The data callback function will call the correct package based on the ACTION property in the message. See parseMarketDataPackage() function in client.js for detailed info.

DATA (MD):
{"ACTION":"MD","MD":[{"last":"132.30","symbol":"IBM","high":"132.39","low":"12
7.97","market_value":"1.26M","average_price":"125.00","change_price":"-
2.73","change_percent":-
2.02,"bid_price":"132.31","ask_size":"5","ask_price":"132.34","volume":"8.14M","bid_s
ize":"4","sec_type":"STK","expiry_type":"0","company_name":"INTL BUSINESS
MACHINES
CORP","last_size":"1","bbo_exchange":"a6","contract_description_1":"IBM","listing_ex
change":"NYSE","close_price":"8/10","dividend_yield":"5.0%","open_price":"130.50","
unformatted_volume":8140000.5,"conid":8314,"conidEx":"8314@SMART","exchange":
"SMART"}],"MESSAGE_TYPE":"FORWARD"}

DATA (MD):
{"ACTION":"MD","MD":[{"bid_price":"132.27","ask_price":"132.30","conid":8314,"c
onidEx":"8314@SMART","exchange":"SMART"}],"MESSAGE_TYPE":"FORWARD"
}

DATA (MD):
{"ACTION":"MD","MD":[{"volume":"8.15M","unformatted_volume":8149999.5,"coni
d":8314,"conidEx":"8314@SMART","exchange":"SMART"}],"MESSAGE_TYPE":"F
ORWARD"}

DATA (MD): {"ACTION":"MD","MD":[{"last":"132.23","change_price":"-
2.80","change_percent":-
2.07,"last_size":"2","conid":8314,"conidEx":"8314@SMART","exchange":"SMART"}],
"MESSAGE_TYPE":"FORWARD"}

If you subscribe to an incorrect conid-exchange pair, you get an error like below:

{method: "message", message: "Market Data subscription failed",
ERRORS:["WTE020"]}

Potential Error Codes
WTE008
WTE020
WTE028

Data fields returned after subscription in MD Action is:MD Object key value pair of:

con_id

exchange

change_percent

change_price

bid_size

bid_price

ask_size

ask_price

volume (in K, M, B)

high

low

market_value

average_price

symbol

company_name

contract_description_1: contains the symbol in most cases

last_price

yield_last

mark_price

is_delayed(true/false) : If user doesn’t have market data permission for this conid, exchange pair; but we have delayed market data available then this flag is true.

open

close

open_price

close_price

dividend_yield

4.6 Market Data Unsubscribe

To unsubscribe market data the app should call 'mdUnsubscribe' method with a conid-exchange-pairs parameter. This parameter can contain one or more conid-exchange pairs separated by a semicolon character. Ex: 8314:SMART;43645865:SMART

Input below:

{ method: 'mdSubscribe', md: <string> }

After un-subscribe, market data stops ticking for the un-subscribed conid, exchange pair. Only unsubscribe for conid-exchange pair, to which you have subscribed earlier.

If you unsubscribe to incorrect conid-exchange pair, you get an error like below:

{method: "message", message: "Cancelling Market Data subscription failed", ERRORS:["WTE020"]}

Potential Error Codes
WTE008
WTE020

4.7 Chart Data Subscribe

To subscribe to chart data the application should call 'chartSubscribe' method with three parameters. The first parameter contains one (and only one) conid-exchange-pair. The second parameter is the length of the chart. This value can be selected from a predefined list (1d, 1w, 2w, 1m, 6m, 1y, 5y). The last one is a boolean for “Outside Regular Trading Hours”.

Input below:

{ method: 'chartSubscribe', md: <string> , time: <string> , orth: <boolean> }

After a successful Chart subscribe the data callback function called the chart packages is invoked. The CHART type packages contain the data in following format.

The chart data is sent to parent frame every 1 minute.

{ method: 'data', message: {"RESULT":true,"MESSAGE_TYPE":"BOTH","SESSION":"d53d96e0-0095-4ef0-bc33-

2b7117d6ca03","data_points":[{"open":147.37,"time":1442237430000,"volume":607,"high":147.37,"low":147,"close":147.2},{"open":147.21,"time":1442237490000,"volume":19,"high":147.31,"low":147.18,"close":147.31},{"open":147.14,"time":1442237550000,"volume":81,"high":147.31,"low":147.06,"close":147.14},{"open":146.35,"time":1442238150000,"volume":53,"high":146.38,"low":146.24,"close":146.38},{"open":146.29,"time":1442238210000,"volume":23,"high":146.29,"low":146.14,"close":146.14},{"open":145.46,"time":1442257830000,"volume":18,"high":145.48,"low":145.44,"close":145.46},{"open":145.47,"time":1442257890000,"volume":71,"high":145.54,"low":145.47,"close":145.53}],"start_time":"2015091413:30:30","ACTION":"CHART"}}

If your chart subscribe fails, you get an error like below:

{ method: 'data', message:
{"RESULT":false,"MESSAGE_TYPE":"BOTH","SESSION":"6008e42c-6e22-4c08-97a9-
a092c6201517","ERRORS":["WTE021"],"ACTION":"CHART_UNSUBSCRIBE"}}

Potential Error Codes
WTE008
WTE021
WTE022
WTE028

4.8 Chart Data Unsubscribe

To unsubscribe chart data the app should call 'chartUnsubscribe' method with a one (and only one) conid-exchange-pair. Once unsubscribe is called, the chart details stop being sent to parent every 1 minute.

Input:

{ method: 'chartUnsubscribe', md: <string> }

If chart unsubscribe fails, you will a get message like the below:

{ method: 'data', message:
{"RESULT":false,"MESSAGE_TYPE":"BOTH","SESSION":"6008e42c-6e22-4c08-97a9-
a092c6201517","ERRORS":["WTE021"],"ACTION":"CHART_UNSUBSCRIBE"}}

Potential Error Codes
WTE008
WTE021

WTE028

4.9 Proxy all Client Portal Web API calls

If the client application is using version 1.4, then they can request all calls available here: https://www.interactivebrokers.com/api/doc.html using the cpapi application too.

Example requests below.

Request message:

{ method: "cpwebapiRequest", httpMethod: "GET", url: "/iserver/contract/8314/info" }

Response message:

{ method: "data", message:
{"type":"CPWEBAPI","request":{"method":"GET","url":"/iserver/contract/8314/info"},"

response":{"statusCode":200,"body":{"cfi_code":"","symbol":"IBM","cusip":null,"expiry_full":null,"con_id":8314,"maturity_date":null,"industry":"Computers","instrument_type":"STK","trading_class":"IBM","valid_exchanges":"ZERO,AMEX,NYSE,CBOE,PHLX,ISE,CHX,ARCA,ISLAND,DRCTEDGE,BEX,BATS,EDGEA,CSFBALGO,JEFFALGO,BYX,IEX,EDGX,FOXRIVER,PEARL,NYSENAT,LTSE,MEMX,PSX","allow_sell_long":false,"is_zero_commission_security":true,"local_symbol":"IBM","contract_clarification_type":null,"classifier":null,"currency":"USD","text":null,"underlying_con_id":0,"r_t_h":true,"multiplier":null,"underlying_issuer":null,"contract_month":null,"company_name":"INTL BUSINESS MACHINES CORP","exchange":"ZERO","category":"Computer Services"}}}}

Request message:
{ method: "cpwebapiRequest", httpMethod: "GET", url: "/portfolio/accounts }

Response message:

{ method: "data", message:
{"type":"CPWEBAPI","request":{"method":"GET","url":"/portfolio/accounts"},"response":{"statusCode":200,"body":[{"id":"DU91474","accountId":"DU91474","accountVan":"DU91474","accountTitle":"","displayName":"DU91474","accountAlias":null,"accountStatus":1280894400000,"currency":"USD","type":"DEMO","tradingType":"PMRGN","ibEntity":"IBLLC-
US","faclient":false,"clearingStatus":"O","covestor":false,"parent":{"mmc":[],"accountId":"","isMParent":false,"isMChild":false,"isMultiplex":false},"desc":"DU91474"}]}}}

For additional help with the Client Portal Web API documentation and end points available, please reach out to api@interactivebrokers.com

5. Unsolicited Messages from IB

All market data messages are sent to the parent once they arrive. On subscription of a market data message, the server pushes every tick to the client, and the parent does not need to poll for it. If websocket is enabled for the browser and it is regular trading hours, market data ticks ever 250 ms. If browser does not support websocket or if websocket is disconnected for any reason, market data is sent to the client every three seconds.

{ method: 'data',
message:{"MESSAGE_TYPE":"FORWARD","ACTION":"MD","MD":[{"symbol":"IB
KR","conidex":"43645865","company_name":"INTERACTIVE BROKERS GRO-CL
A","expiry_type":"0","sec_type":"STK","listing_exchange":"NASDAQ.NMS","contract
_description_1":"IBKR","exchange":"SMART","conid":43645865}]}}
{ method: 'data',
message:{"MESSAGE_TYPE":"FORWARD","ACTION":"MD","MD":[{"change_price
":"-
1.90","bid_size":"3","last_price":"145.47","is_delayed":false,"open_price":"147.37","cha
nge_percent":"-
1.29%","market_value":"1,018","conid":8314,"mark_price":"145.47","bid_price":"145.4
7","ask_price":"145.48","last_trading_day":"20150911","volume":"1.61M","close_price"
:"147.37","dividend_yield":"3.6%","high":"147.37","low":"145.41","exchange":"SMAR
T","ask_size":"8"}]}}

Additionally, the Chart message is sent to parent frame every 1 minute by IB Server.

The following messages are sent from the server if there are any application problems. The client should either try to reconnect or show login window again.

WT Error	What to do
WTE002	Show the login window
WTE003	Show the login window
WTE006	Let user decide to end current session or disconnect other session; call force initialization accordingly
WTE008	Show the login window

6. Signed data and its verification from calling program

Once a user has logged in, there will be three messages sent to the client. The data will be formatted like the below:

DATA 1:

{"BACKEND_SESSION_ID":"581c0b12.00000280","ERRORS":[],"ACCOUNTS":"DU91474","MESSAGE_TYPE":"FORWARD","ID":"":"d53d96e0-0095-4ef0-bc33-2b7117d6ca03","USER":"qapap908","CURRENT_TIME":1478290612110,"RESULT":true}

DATA 2:

{"SIGNATURE_TYPE":"USER_SESSION","MESSAGE_TYPE":"FORWARD","ID":"":"d53d96e0-0095-4ef0-bc33-2b7117d6ca03","USER":"qapap908","RESULT":true,"SIGNATURE":"OBi7YFp3lBY1d+jMvGg7NwJtsJpNVYqyD67WbQR+wAfRzHvwLVGTEd3EeIaITw3tOtTMzzKuQR4/BJKp1qcLryGPHo5r3sJcgnNEXOqal9g8P/plNpjkgLVaVd0uDgp0aiis+EXW2Z38LKhOEP/3vdGvR4Lvin9jjtmEts724aUnMfVg9ZiNK/dbWd1ADgXgid/f1fkRekDDbDmGphYaQ3jEv0bSkCfg6d8rc/jW+gDAiBGMbPnu9Jxvx1vjbT4vkGPL9PkcpgEJHgFKDXbw6IlNDJ6sHvX41WXZuv/6o49DpgXyITQQhgPwIf7HbKs5uAA4IJSoNEfHVB+sHyv/hnONZIE1vNPjwewtRC84Sob1y7pblYY3eUusfzy756y63qTVBfOoLynUEj12pihs6zvghZd26kMGE1TexO+hs4EkepY1Fnq5jzR0IA5cp0dBXpehGiXbdXp1FwEFFDjWlNUlDpLSgIwcy4SKy6xstPviiasZktlUKbQP1ruTXw4ANcUXLcdggbhABhXWjqsFz8WN9M65hHTCx2CpM3gZn2uLRXz4gg2aRUL4ilXlgMRKi2ILDgWSxVn7R80Q1lhRalt1jmmUBjNY7pQJvGsSOQ3+66i4hiedctWcVugDHsF3IsxVgrp04PmIUjd4XNYKfuhrFsJeQVjCYHAfxduC+bo="}

DATA 3:

{"SIGNATURE_TYPE":"USER_ACCOUNTS","MESSAGE_TYPE":"FORWARD","ID":"":"d53d96e0-0095-4ef0-bc33-2b7117d6ca03","USER":"qapap908","RESULT":true,"SIGNATURE":"REJSCdTG23xmiPOgACV8uyoeq3vIiTnGo/3EygQ/C3R1FAiXn3e59tAEm6FER15LikWjNw9IXLcsEdiRsKbc+MHZ6iFsU7UrPY/ZORdWL0KkXEqu1swo2ZE4KaT3O0Dujnh+GFGSQcNdYP2I7lSWXS5442Bub06pShXuGOmC1Dd9FdmwKWCrR1kTwm9RsylhLB32OHaNtdaxMYgULPbDH8WVUjAYheRZ389F9yHoSqSRlzSQUdZIch66/747vABMuFzaAYqxxbM0LL0k1fpXpifuFAAqOTMjwtZB6HmSbpgwH45NRD5b7WRrwyFcm8HLpa2P2isc6lCk/AxY42Ay8IQUvluFzSmE3N+4vCbiWf9IiB2jcovZoZb0J/twm2E98J9ryP76IvgdULdMBDciy7J728dSoHr0SOPmBkjGWHrF8sLrKpVFm6YJ3t6UrOhMboOiPIIUe/m/e3WAUn48NOYaOpwiKjjkmBfi5H05Z+UtMMLZErt7w4vRdgbfEb+IgsLX75HBSUwcOhyH

P0/Tm/DljFRKNYhTJ2yPE+oKqiC2M8iP8XNChurxFQKqIgT2CaIqCRSAd48C6hBEZqacX4CUGbeKDaptnRwkaIopzlqIQ+NWVLLlPulV3aIOS81W8+oq73dNa6/aXnR8fEoIKk9acvM7Sjy1NbSPumnYw+Q="}

DATA2 is the **username|backend_session_id|time** which has been signed

DATA3 is the **username|accounts|time** which has been signed

When the calling application gets the data, they will take the Signature and verify it against the unsigned data using the downloaded certificate.

The public certificate to be used for verifying the data can be downloaded from:

https://api.ibkr.com/sso/cpapi/downloadCertificate

The steps to verify (to be done by the calling application):

1. Download and save the public certificate from IB (at a certain file location on your end). This doesn't need to be done every day. It can be once, and then every time the key changes (IB will notify application team).
2. Ensure that the current_time is within a few minutes of the application's server time. This ensures that no one is playing back old messages.
3. Run Verify (sample java code attached). If Verify is true and 2 is true, then you can proceed knowing that the user has logged into the IB system in the last few minutes.
4. Send the backend_session_id on orders for additional security.

7. ERROR Codes

All error codes with descriptions. The last column indicates if the message can be seen in the local login page, is sent to parent, or both.

WT Error	Description	Where
WTE001	Not Supported Message	Both
WTE002	Generic Error	Both
WTE003	Disconnected	Both
WTE004	Unable to Connect	Both
WTE005	Invalid username or password	Local
WTE006	Competition: user connected on another machine	Both

WT Error	Description	Where
WTE007	Unable to connect to api	Both
WTE008	Not connected to server	Both
WTE009	Restricted IP	Local
WTE010	User not specified	Local
WTE011	User locked out	Local
WTE012	Password expired	Local
WTE013	Session Token AuthenticationFailed	Local
WTE014	Session Token AuthenticationPassed	Local
WTE015	AUTHENTICATED	Local
WTE016	NEED_AUTHENTICATION	Local
WTE017	Guaranteed Dollar user not supported	Local
WTE018	Invalid Whitebranded user	Parent
WTE019	Invalid Parent domain	Parent
WTE020	Invalid subscription	Parent
WTE021	Invalid chart request	Parent
WTE022	Chart Result failed	Parent
WTE023	Invalid Timeout	Parent
WTE024	Invalid contract	Parent
WTE025	Invalid Login Message	Both
WTE028	Invalid access	Parent
WTE099	Min Version	Parent

8. Security Checks

Only valid white branded users and valid domains are allowed to use this application. Applications will need to register their test and prod domains with IB (this will be discussed during setup).

Also check with IB for the correct iframe domain to be used for PostMessage validations on the parent side. This could be different for test and production environments.

9. Style Sheet for Each Institution

A Custom CSS style sheet to alter the look of the login page can be used for each institution. These styles sheets will be hosted by IB, separately from the application. CP API will contain the default configuration but will allow different style sheets per white branding ID to be loaded after the default style sheet. This style sheet for the whitebranded user will be served from a different location so that style sheet changes will not require a release.

The custom style sheet can overwrite the colors, background colors, font family, font size, etc. Recommended selectors to use are body, input.text, button.submit, h1, .title. Institutions will have to test their modified style sheet on every major supported browser before sending to IB. IB will not be held responsible for the broken layout of a login page after incorporating the custom style sheet.

Style sheets to be incorporated for each white branded user (if any) will be confirmed during setup.

10. Translations

Institutions can be provided with the Error codes for translations (as needed). The process followed would be the same as for other IB applications and would be through CLAMS. Users could choose from a list of languages that we offer for translation and translated text, error codes would be shown on the login page.

Please contact IB Sales Engineering **(**salesengineering@interactivebrokers.com**)** for additional translation needs.

11. Required from Institutions

Institutions must provide the following information to IB before they start:

- Is functionality provided using Mobile App and/or Browser based app?
- Is application to be used for Authentication, market data, or both?
- Provide a white branded ID (string)
- Tell us how long server should keep the session alive (in minutes, integer between 1 and 30).
- Provide the Test and Prod domains from which you will call cpapi
- Number of users who will use the application in production, by region
- Number of users during peak time
- Additional translation needs, if any

Please contact IB Sales Engineering (salesengineering@interactivebrokers.com) with the above information and confirm that white-branded users are set up for testing before proceeding with development/testing.

Appendix

The demo url is:

https://cdcdyn.interactivebrokers.com/sso/cpapi/frame/

The sample wtapi.js can be found at:

https://cdcdyn.interactivebrokers.com/sso/cpapi/frame/wtapi.js

The sample client.js can be found at:

https://cdcdyn.interactivebrokers.com/sso/cpapi/frame/client.js

Please copy these 2 files (wtapi.js and client.js) and customize as needed in your application.

Sample Java program to verify signature:


VerifySignature.java