Best Practice

Development & Testing

You should use the **Delayed Application Key** for any initial development and functional testing. Historical data is made available via **https://historicdata.betfair.com/#/home** for strategy modelling & analysis.

Only apply for Live Application Key access once you are ready to start transacting on the Exchange using your Live Application Key.

Please see the **Personal Betting Access Overview** for more details regarding the difference between Delayed an Live Application Keys.

Session Management

Use Login to create a new session and Keep Alive to extend the session beyond the session expiry time. A single session can and should be used across multiple API calls/threads simultaneously.

You should ensure that you handle the INVALID_SESSION_TOKEN error within your code by creating a new session token via the API login method.

General Tips

- Make the minimal number of transactions/changes possible when transacting.
- Observe the Market Data Limits when making requests to listMarketCatalogue, listRunnerBook, listMarketBook and listMarketProfitandLoss
- Always prefer leaving an order in place rather than cancelling/re-placing it – stay at the front of the queue to be matched!
- Use the Stream API instead of polling wherever possible, particularly if you are running a high frequency trading application.
- Log as much as possible to aid queries/problem investigation (especially connectionid when using the Stream API)
- Make use of betting enhancements

API Status

Use the API status page http://status.developer.betfair.com/ to check the health of the API.

The API Status:

- Measures response latency and error rate against a number of operations every second
- Automatically toggles the status page if certain thresholds are breached

Please check the API status before contacting Developer Support regarding API problems.

Expect: 100 - Continue Header

Sending this header will result in the error: **"The remote server returned an error: (417) Expectation Failed."**

You should be aware that if using the .Net Framework you will need to set the relevant property in the ServicePointManager which then prevents the "Expect" header from being added:

System.Net.ServicePointManager.Expect100Continue = false;

Enabling HTTP Compression

HTTP compression is a capability built into both web servers and web clients to reduce the number of bytes transmitted in an HTTP response. This makes better use of available bandwidth and increases performance while reducing download time. When enabled, HTTP protocol data is compressed before it is sent from the server. Clients capable of receiving compressed HTTP data announce that they support compression in the HTTP header. Almost all modern web browsers support HTTP Compression by default.

The Betfair API uses HTTP to handle communication between API clients and servers. Therefore, the JSON messages can be compressed using the same HTTP compression used by web browsers. Custom API applications may need some modification before they can take advantage of this feature. Specifically, they need to send an additional HTTP header to indicate they support receipt of compressed responses from the API. In addition, some environments require you to explicitly decompress the response.

We would therefore recommend that all Betfair API request are sent with the '**Accept-Encoding: gzip, deflate'** request header.

HTTP Persistent Connection

We recommend that **Connection: keep-alive** header is set for all requests to guarantee a persistent connection and therefore reducing latency. **Please note**: Idle keep-alive connection to the API endpoints are closed every 3 minutes.

Other Performance Tips

Additional advice regarding optimizing HTTPClient performance can be found via **Connection management**