Non-Interactive (bot) login

The non-interactive login method for the Betfair API requires that you create and upload a self-signed certificate which will be used, alongside your username and password to authenticate your credentials and generate a session token.

For the purposes of this guide, we have used openssl to generate this client, details of which can be found at **http://www.openssl.org/**

2 Step Authentication With Non Interactive Login

Using **2 Step Authentication** to secure your account for website logins will have no impact on your use of the non-interactive login method and vice versa.

Getting Started

There are a couple of steps required before we can actually log in:

1. Create a self-signed certificate
2. Link the certificate to your Betfair account

Creating a Self Signed Certificate

API-NG requires that a 1024-bit or 2048-bit RSA certificate be used. There are various tutorials available on the Internet but be aware that the certificate needs to be for client authentication (most tutorials only cover server authentication).

Create a public/private RSA key pair using openssl

```
openssl genrsa -out client-2048.key 2048
```

Update or Create the openssl configuration file (openssl.cnf) for OpenSSL to override some of the default settings:

```
[ ssl_client ]
basicConstraints = CA:FALSE
nsCertType = client
keyUsage = digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth
```

In Windows, the config file is located in the installation directory of OpenSSL

In Linux distributions, the config file is located at /usr/lib/ssl/openssl.cnf or /etc/ssl/openssl.cnf

Create a certificate signing request (CSR).

```
openssl req -new -config openssl.cnf -key client-2048.key -out client-2048.csr


Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:London
Locality Name (eg, city) []:London
Organization Name (eg, company) [Internet Widgits Pty Ltd]:yourcompany.com
Organizational Unit Name (eg, section) []:Security Team
Common Name (e.g. server FQDN or YOUR name) []:Test API-NG Certificate
Email Address []:my.name@mydomain.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Self-sign the certificate request to create a certificate

```
openssl x509 -req -days 365 -in client-2048.csr -signkey client-2048.key -out client-2048.crt -extfile openssl.
cnf -extensions ssl_client
```

In Windows, using any text editor, copy the contents of the .crt file and the .key file into a new file. Save this new file as client-2048.pem.

Linking the Certificate to Your Betfair Account

The previous steps should have created the following files:

File name	Description
client-2048.key	The private key. This file is needed in order to use the certificate and should be protected and shouldn't be shared with anyone.
client-2048.csr	A certificate signing request. This file is no longer needed and can be deleted.
client-2048.crt	The certificate. This file is not sensitive in security terms and can be shared with anyone.

Before you login using the certificate, it must be attached to your Betfair account, as follows:

1. Log in to your Betfair account through betfair.comPaste the following URL into the address bar of your browser
2. Navigate to https://myaccount.betfair.com/accountdetails/mysecurity?showAPI=1 **- Note:** Please use https://myaccount.betfair.it/accountdetails/mysecurity?showAPI=1 for the **Italian Exchange or the endpoint relevant to your own jusristiction. See the URL Definition section for more details**
3. Scroll to the section titled "**Automated Betting Program Access**" and click **'Edit'**
4. Click on "Browse" and then locate and select the file client-2048.crt (client-2048.pem if applicable) created above.
5. Click on the "**Upload Certificate**" button.

Scroll down to the "**Automated Betting Program Access**" section if required and the certificate details should be shown. You should now be able to log in to your Betfair account using the API-NG endpoint.

Note on File Formats

Some systems require that client certificates are in a different format to the ones we've created. The two most common formats are (a) PEM format key and certificate in a single file and (b) PKCS#12 format file. .NET applications require a PKCS#12 format file.

To create a PEM format file that contains both the private key and the certificate you can use the following command:

Linux

```
cat client-2048.crt client-2048.key > client-2048.pem
```

Create the PKCS#12 format using crt and key

```
openssl pkcs12 -export -in client-2048.crt -inkey client-2048.key -out client-2048.p12
```

Don't circulate the key, PEM file or PCKS#12 format files as these files are security sensitive

Details of a Login Request

A login request can now be made as follows:

1. Submit a HTTP “POST” request to: https://identitysso-cert.betfair.com/api/certlogin
2. As part of the SSL connection, the certificate created previously must be supplied.
3. Include a custom Header called “X-Application” with a value that identifies your application. The value is not validated and is only used to help with troubleshooting and diagnosing any problems.
4. Ensure the POST’s Content-Type is “application/x-www-form-urlencoded” rather than MIME attachment encoded.
5. As part of the POST body include two parameters “username” and “password” which should have the relevant username/password for your account.

Certificate Login Interface Details

URL Definition

Certificate Endpoint
`https://identitysso-cert.betfair.com/api/certlogin`

This endpoint is also available under the following jurisdictions

Please use the below if your country of residence is in one of the list jurisdictions.

Jurisdiction	Endpoint
Italy	`https://identitysso-cert.betfair.it`
Spain	`https://identitysso-cert.betfair.es`
Romania	`https://identitysso-cert.betfair.ro`
Sweden	`https://identitysso-cert.betfair.se`

Please note: Danish residents cannot use the Non-Interactive (bot) login method due to the NEMID requirement which is only supported by the Interactive Login - Desktop Application method

Request headers

- **X-Application** - You must set the X-Application header to your application key.

Request Parameters

- **username** (mandatory) - The username of the user logging in.
- **password** (mandatory) - The password of the user logging in.

Please note: The username and password values should be encoded when making the login request. All method names are case sensitive, this includes login, keepAlive and logout.

Response

The response returned is a json string. If the response is successful then the loginStatus key will contain SUCCESS, for example:

```
{
  sessionToken: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx;
  loginStatus: SUCCESS;
}
```

Should a failure or exception be returned, the response will be structured as below and loginStatus will contain a failure reason:

```
{
  loginStatus: INVALID_USERNAME_OR_PASSWORD;
}
```

The possible failure and exceptional return codes are:

error	Description
INVALID_USERNAME_OR_PASSWORD	the username or password are invalid
ACCOUNT_NOW_LOCKED	the account was just locked
ACCOUNT_ALREADY_LOCKED	the account is already locked
PENDING_AUTH	pending authentication
TELBET_TERMS_CONDITIONS_NA	Telbet terms and conditions rejected
DUPLICATE_CARDS	duplicate cards
SECURITY_QUESTION_WRONG_3X	the user has entered wrong the security answer 3 times
KYC_SUSPEND	KYC suspended
SUSPENDED	the account is suspended
CLOSED	the account is closed
SELF_EXCLUDED	the account has been self-excluded
INVALID_CONNECTIVITY_TO_REGULATOR_DK	the DK regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
NOT_AUTHORIZED_BY_REGULATOR_DK	the user identified by the given credentials is not authorized in the DK's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the DK's jurisdiction.
INVALID_CONNECTIVITY_TO_REGULATOR_IT	the IT regulator cannot be accessed due to some internal problems in the system behind or in at regulator; timeout cases included.
NOT_AUTHORIZED_BY_REGULATOR_IT	the user identified by the given credentials is not authorized in the IT's jurisdictions due to the regulators' policies. Ex: the user for which this session should be created is not allowed to act(play, bet) in the IT's jurisdiction.
SECURITY_RESTRICTED_LOCATION	the account is restricted due to security concerns
BETTING_RESTRICTED_LOCATION	the account is accessed from a location where betting is restricted
TRADING_MASTER	Trading Master Account
TRADING_MASTER_SUSPENDED	Suspended Trading Master Account
AGENT_CLIENT_MASTER	Agent Client Master
AGENT_CLIENT_MASTER_SUSPENDED	Suspended Agent Client Master
DANISH_AUTHORIZATION_REQUIRED	Danish authorization required
SPAIN_MIGRATION_REQUIRED	Spain migration required
DENMARK_MIGRATION_REQUIRED	Denmark migration required
SPANISH_TERMS_ACCEPTANCE_REQUIRED	The latest Spanish terms and conditions version must be accepted. You must login to the website to accept the new conditions.
ITALIAN_CONTRACT_ACCEPTANCE_REQUIRED	The latest Italian contract version must be accepted. You must login to the website to accept the new conditions.

CERT_AUTH_REQUIRED	Certificate required or certificate present but could not authenticate with it
CHANGE_PASSWORD_REQUIRED	Change password required
PERSONAL_MESSAGE_REQUIRED	Personal message required for the user
INTERNATIONAL_TERMS_ACCEPTANCE_REQUIRED	The latest international terms and conditions must be accepted prior to logging in.
EMAIL_LOGIN_NOT_ALLOWED	This account has not opted in to log in with the email
MULTIPLE_USERS_WITH_SAME_CREDENTIAL	There is more than one account with the same credential
ACCOUNT_PENDING_PASSWORD_CHANGE	The account must undergo password recovery to reactivate via https://identitysso.betfair.com/view/recoverpassword
TEMPORARY_BAN_TOO_MANY_REQUESTS	The limit for successful login requests per minute has been exceeded. New login attempts will be banned for 20 minutes
ITALIAN_PROFILING_ACCEPTANCE_REQUIRED	You must login to the website to accept the new conditions
AUTHORIZED_ONLY_FOR_DOMAIN_RO	You are attempting to login to the Betfair Romania domain with a non .ro account.
AUTHORIZED_ONLY_FOR_DOMAIN_SE	You are attempting to login to the Betfair Swedish domain with a non .se account.
SWEDEN_NATIONAL_IDENTIFIER_REQUIRED	You must provided your Swedish National identifier via Betfair.se before proceeding.
SWEDEN_BANK_ID_VERIFICATION_REQUIRED	You must provided your Swedish bank id via Betfair.se before proceeding.
ACTIONS_REQUIRED	You must login to https://www.betfair.com to provide missing information.

Sample curl command to quickly check the certificate based login

```
curl -q -k --cert client-2048.crt --key client-2048.key https://identitysso-cert.betfair.com/api/certlogin -d
"username=testuser&password=testpassword" -H "X-Application: curlCommandLineTest"

Response should be

{"sessionToken":"Zx8i4oigut5nc+l4L8qFb0DSxG+mwLn2t0AMGFxjrMJI=","loginStatus":"SUCCESS"}
```

Sample Code for Non-Interactive Login

Sample C# code using PKCS#12 key store

Please see code sample via https://github.com/betfair/API-NG-sample-code/tree/master/loginCode/Non-interactive-cSharp

Sample Java code using Apache http client library and PKCS#12 key store

Java API-NG login

```
package com.test.aping.client;



import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.entity.UrlEncodedFormEntity;
```

```
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.ClientConnectionManager;
import org.apache.http.conn.scheme.Scheme;
import org.apache.http.conn.ssl.SSLSocketFactory;
import org.apache.http.conn.ssl.StrictHostnameVerifier;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

import javax.net.ssl.KeyManager;
import javax.net.ssl.KeyManagerFactory;
import javax.net.ssl.SSLContext;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.SecureRandom;
import java.util.ArrayList;
import java.util.List;


public class HttpClientSSO {

    private static int port = 443;

    public static void main(String[] args) throws Exception {

        DefaultHttpClient httpClient = new DefaultHttpClient();

        try {
            SSLContext ctx = SSLContext.getInstance("TLS");
            KeyManager[] keyManagers = getKeyManagers("pkcs12", new FileInputStream(new File("C:
\\sslcerts\\client-2048.p12")), "password");
            ctx.init(keyManagers, null, new SecureRandom());
            SSLSocketFactory factory = new SSLSocketFactory(ctx, new StrictHostnameVerifier());

            ClientConnectionManager manager = httpClient.getConnectionManager();
            manager.getSchemeRegistry().register(new Scheme("https", port, factory));
            HttpPost httpPost = new HttpPost("https://identitysso-cert.betfair.com/api/certlogin");
            List<NameValuePair> nvps = new ArrayList<NameValuePair>();
            nvps.add(new BasicNameValuePair("username", "testuser"));
            nvps.add(new BasicNameValuePair("password", "testpassword"));

            httpPost.setEntity(new UrlEncodedFormEntity(nvps));



            httpPost.setHeader("X-Application","appkey");


            System.out.println("executing request" + httpPost.getRequestLine());

            HttpResponse response = httpClient.execute(httpPost);
            HttpEntity entity = response.getEntity();

            System.out.println("----------------------------------------");
            System.out.println(response.getStatusLine());
            if (entity != null) {
                String responseString = EntityUtils.toString(entity);
                            //extract the session token from responsestring
                            System.out.println("responseString" + responseString);
            }

        } finally {
            httpClient.getConnectionManager().shutdown();
        }
    }



    private static KeyManager[] getKeyManagers(String keyStoreType, InputStream keyStoreFile, String
```

```
keyStorePassword) throws Exception {
        KeyStore keyStore = KeyStore.getInstance(keyStoreType);
        keyStore.load(keyStoreFile, keyStorePassword.toCharArray());
        KeyManagerFactory kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
        kmf.init(keyStore, keyStorePassword.toCharArray());
        return kmf.getKeyManagers();
    }
}
```

Sample Python code

```
#!/usr/bin/env python

import requests

#openssl x509 -x509toreq -in certificate.crt -out CSR.csr -signkey privateKey.key


payload = 'username=myusername&password=password'
headers = {'X-Application': 'SomeKey', 'Content-Type': 'application/x-www-form-urlencoded'}

resp = requests.post('https://identitysso-cert.betfair.com/api/certlogin', data=payload, cert=('client-2048.
crt', 'client-2048.key'), headers=headers)

if resp.status_code == 200:
  resp_json = resp.json()
  print resp_json['loginStatus']
  print resp_json['sessionToken']
else:
  print "Request failed."
```