**PLAYTECH  
Integration Documentation**

**API INTEGRATION**

**Integration API server**

https://kioskpublicapi.<*cloudlocation>*.com

Do not forget to add the entity key to the HTTP header. In this example the Top Level Entity key must be used together with the certificate, otherwise the transaction will not work.

Accept:text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8Cache-Control: max-age=0Connection: keep-aliveKeep-Alive:timeout=5, max=100Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3Accept-Language:es-ES,es;q=0.8Pragma: X\_ENTITY\_KEY:<*entity key>*

Sample of entity key:

*3cf272869754310eca13e63f53f333c241874bd4736e69f90b647008b0fb4f843e81bea5e8f378aed6cff881699fb85011eaa2cc8cc8f12c1c2126de609df5692*

**REM: Please refer to Appendix A for the value of your brand’s virtual database, cloud location, brand code top level entity and entity key.**

**Sample Code to Call API - PHP**

<?php

$path = dirname(*\_\_FILE\_\_*);

$url= " https://kioskpublicapi.<cloudlocation>.com/player/info/playername/<name>";

$entity\_key= "<your\_entity\_key>";

$header = array();

$header[] = "Accept:text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8";

$header[] = "Cache-Control: max-age=0";

$header[] = "Connection: keep-alive";

$header[] = "Keep-Alive:timeout=5, max=100";

$header[] = "Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3";

$header[] = "Accept-Language:es-ES,es;q=0.8";

$header[] = "Pragma: ";

$header[] = "X\_ENTITY\_KEY: " . $entity\_key;

$tuCurl= curl\_init();

curl\_setopt($tuCurl, *CURLOPT\_URL*, $url);

curl\_setopt($tuCurl, *CURLOPT\_PORT* , 443);

curl\_setopt($tuCurl, *CURLOPT\_VERBOSE*, 0);

curl\_setopt($tuCurl, *CURLOPT\_HTTPHEADER*, $header);

curl\_setopt($tuCurl, *CURLOPT\_TIMEOUT*, 60000 );

curl\_setopt($tuCurl, *CURLOPT\_SSL\_VERIFYPEER*, 0);

curl\_setopt($tuCurl, *CURLOPT\_SSL\_VERIFYHOST*, 0);

curl\_setopt($tuCurl, *CURLOPT\_SSLCERT*, $path . '</api/ssl.pem>');

curl\_setopt($tuCurl, *CURLOPT\_RETURNTRANSFER*, 1);

curl\_setopt($tuCurl, *CURLOPT\_SSLKEY*, $path . '</api/ssl.key>');

$exec = curl\_exec($tuCurl);

curl\_close($tuCurl);

$data = json\_decode($exec, *TRUE*);

echo"<pre>";

print\_r($data);

echo"</pre>";

?>

**Sample Code to Call API – C#**

private void callAPI()

{

string rURL = "https://kioskpublicapi.<cloudlocation>.com/player/info/playername/<name>";

string data = "playername=" + <username>;

byte[] dataStream = Encoding.UTF8.GetBytes(data);

HttpWebRequest Request = (HttpWebRequest)WebRequest.Create(rURL);

HttpWebResponse Response = null;

Request.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8";

Request.Headers.Add("Cache-Control", "max-age=0");

Request.KeepAlive = true;

Request.Headers.Add("Keep-Alive", "timeout=5, max=100");

Request.Headers.Add("Accept-Charset", "ISO-8859-1,utf-8;q=0.7,\*;q=0.3");

Request.Headers.Add("Accept-Language", "es-ES,es;q=0.8");

Request.Headers.Add("Pragma", "");

Request.Headers.Add("X\_ENTITY\_KEY", <your\_entity\_key>);

Request.Method = "POST";

Request.ContentType = "application/x-www-form-urlencoded";

Request.ClientCertificates.Add(new X509Certificate2(<SSLCert.p12\_path>, <password>, X509KeyStorageFlags.MachineKeySet));

ServicePointManager.ServerCertificateValidationCallback = CertificateValidationCallBack;

Request.ContentLength = dataStream.Length;

Stream newStream = Request.GetRequestStream();

// Send the data.

newStream.Write(dataStream, 0, dataStream.Length);

newStream.Close();

Response = (HttpWebResponse)Request.GetResponse();

StreamReader reader = new StreamReader(Response.GetResponseStream());

String retData = reader.ReadToEnd();

JObject jObject = JObject.Parse(retData);

JToken jresult = null;

IDictionary<string, JToken> dictionary = jObject;

if (dictionary.ContainsKey("result"))

{

jresult = jObject["result"];

if (jObject.Count > 0)

{

string playername = jresult["PLAYERNAME"].ToString();

string kioskname = jresult["KIOSKNAME"].ToString();

string kioskadminname = jresult["KIOSKADMINNAME"].ToString();

string isfrozen = jresult["FROZEN"].ToString() == "1" ? "YES" : "NO";

}

}

if (dictionary.ContainsKey("error"))

{

jresult = jObject["error"];

}

retData = null;

jObject = null;

}

private static IEnumerable<JToken> AllChildren(JToken json)

{

foreach (var c in json.Children())

{

yield return c;

foreach (var cc in AllChildren(c))

{

yield return cc;

}

}

}

public class TrustAllCertificatePolicy : System.Net.ICertificatePolicy

{

public TrustAllCertificatePolicy()

{ }

public bool CheckValidationResult(ServicePoint sp,

System.Security.Cryptography.X509Certificates.

X509Certificate cert, WebRequest req, int problem)

{

return true;

}

}

private static bool CertificateValidationCallBack(

object sender,

System.Security.Cryptography.X509Certificates.X509Certificate certificate,

System.Security.Cryptography.X509Certificates.X509Chain chain,

System.Net.Security.SslPolicyErrors sslPolicyErrors)

{

// If the certificate is a valid, signed certificate, return true.

if (sslPolicyErrors == System.Net.Security.SslPolicyErrors.None)

{

return true;

}

// If there are errors in the certificate chain, look at each error to determine the cause.

if ((sslPolicyErrors & System.Net.Security.SslPolicyErrors.RemoteCertificateChainErrors) != 0)

{

if (chain != null && chain.ChainStatus != null)

{

foreach (System.Security.Cryptography.X509Certificates.X509ChainStatus status in chain.ChainStatus)

{

if ((certificate.Subject == certificate.Issuer) &&

(status.Status == System.Security.Cryptography.X509Certificates.X509ChainStatusFlags.UntrustedRoot))

{

// Self-signed certificates with an untrusted root are valid.

continue;

}

else

{

if (status.Status != System.Security.Cryptography.X509Certificates.X509ChainStatusFlags.NoError)

{

// If there are any other errors in the certificate chain, the certificate is invalid,

// so the method returns false.

return false;

}

}

}

}

return true;

}

else

{

// In all other cases, return false.

return false;

}

}

**Sample Code to Call API – VB .net**

Public Function getAPI() As String

Dim URL = "https://kioskpublicapi.*<cloudlocation>*.com/player/info/playername/<name>"

Dim entity\_key As String = "<*your\_entity\_key*>"

Dim Request As HttpWebRequest = DirectCast(WebRequest.Create(URL), HttpWebRequest)

Dim Response As HttpWebResponse = Nothing

Request.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8"

Request.Headers.Add("Cache-Control", "max-age=0")

Request.KeepAlive = True

Request.Headers.Add("Keep-Alive", "timeout=5, max=100")

Request.Headers.Add("Accept-Charset", "ISO-8859-1,utf-8;q=0.7,\*;q=0.3")

Request.Headers.Add("Accept-Language", "es-ES,es;q=0.8")

Request.Headers.Add("Pragma", "")

Request.Headers.Add("X\_ENTITY\_KEY", entity\_key)

Request.Method = "POST"

Request.ContentType = "application/x-www-form-urlencoded"

Dim cert As String = HttpContext.Current.Server.MapPath("<*SSLCERT.p12\_path*>")

Dim password As System.Security.SecureString = convertToSecureString("<*cert\_password*>")

Request.ClientCertificates.Add(New X509Certificate2(cert, password, X509KeyStorageFlags.MachineKeySet))

ServicePointManager.ServerCertificateValidationCallback = AddressOf CertificateValidationCallBack

Response = DirectCast(Request.GetResponse(), HttpWebResponse)

Dim reader As New StreamReader(Response.GetResponseStream())

Dim retData As String = reader.ReadToEnd()

Dim j As Object = New JavaScriptSerializer().Deserialize(Of Object)(retData)

'Example

For Each item As Dictionary(Of String, Object) In j("result")

Dim s As String = item("PLAYERNAME")

'For Each val As String In item.Values

' Dim s As String = val(0)

'Next

Next

reader.Close()

Return retData

End Function

Public Function convertToSecureString(ByVal strPassword As String) As SecureString

Dim secureStr = New SecureString()

If strPassword.Length > 0 Then

For Each c As String In strPassword.ToCharArray()

secureStr.AppendChar(c)

Next

End If

Return secureStr

End Function

Private Shared Function CertificateValidationCallBack(ByVal sender As Object, ByVal certificate As System.Security.Cryptography.X509Certificates.X509Certificate, ByVal chain As System.Security.Cryptography.X509Certificates.X509Chain, ByVal sslPolicyErrors As System.Net.Security.SslPolicyErrors) As Boolean

If sslPolicyErrors = System.Net.Security.SslPolicyErrors.None Then

Return True

End If

If (sslPolicyErrors And System.Net.Security.SslPolicyErrors.RemoteCertificateChainErrors) <> 0 Then

If chain IsNot Nothing AndAlso chain.ChainStatus IsNot Nothing Then

For Each status As System.Security.Cryptography.X509Certificates.X509ChainStatus In chain.ChainStatus

If (certificate.Subject = certificate.Issuer) AndAlso (status.Status = System.Security.Cryptography.X509Certificates.X509ChainStatusFlags.UntrustedRoot) Then

Continue For

Else

If status.Status <> System.Security.Cryptography.X509Certificates.X509ChainStatusFlags.NoError Then

Return False

End If

End If

Next

End If

Return True

Else

' In all other cases, return false.

Return False

End If

End Function

**Sample Code to Call API – JAVA**

package javatestapi;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.net.MalformedURLException;

import java.net.URL;

import java.security.KeyStore;

import java.security.KeyStoreException;

import java.security.NoSuchAlgorithmException;

import java.security.SecureRandom;

import java.security.cert.CertificateException;

import java.security.cert.X509Certificate;

import javax.net.ssl.HostnameVerifier;

import javax.net.ssl.HttpsURLConnection;

import javax.net.ssl.KeyManager;

import javax.net.ssl.KeyManagerFactory;

import javax.net.ssl.SSLContext;

import javax.net.ssl.SSLSession;

import javax.net.ssl.TrustManager;

import javax.net.ssl.X509TrustManager;

import org.apache.commons.io.IOUtils;

public class JavaTestAPI {

public static void main(String[] args) {

CallAPI();

}

private static void CallAPI()

{

try {

KeyStore ks = KeyStore.getInstance("PKCS12");

//get certificate file from test/resources as InputFileStream & load to existing keystore

URL fileURL = new File("<*Cert.p12*>").toURI().toURL();

File file = new File(fileURL.getFile());

FileInputStream fis = new FileInputStream(file);

ks.load(fis, "<*cert\_password*>".toCharArray());

//Create KeyManagerFactory using loaded keystore

KeyManagerFactory kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());

kmf.init(ks, "<cert\_*password*>".toCharArray());

KeyManager[] kms = kmf.getKeyManagers();

//Crete TrustManager to bypass trusted certificate check

TrustManager[] trustAllCerts = new TrustManager[] {

new X509TrustManager() {

public java.security.cert.X509Certificate[] getAcceptedIssuers() {

return null;

}

public void checkClientTrusted(X509Certificate[] certs, String authType) { }

public void checkServerTrusted(X509Certificate[] certs, String authType) { }

}

};

//Hostname verification bypass method

HostnameVerifier allHostsValid = new HostnameVerifier() {

public boolean verify(String hostname, SSLSession session) {

return true;

}

};

//Set connection properties to use bypass certificate/hostname check methods

SSLContext sslContext = null;

sslContext = SSLContext.getInstance("TLS");

sslContext.init(kms, trustAllCerts, new SecureRandom());

HttpsURLConnection.setDefaultHostnameVerifier(allHostsValid);

HttpsURLConnection.setDefaultSSLSocketFactory(sslContext.getSocketFactory());

//Send API call together with entity key for validation

HttpsURLConnection connection = (HttpsURLConnection) new URL("https://kioskpublicapi.<*cloudlocation*>.com/player/info/playername/<name>").openConnection();

connection.setRequestProperty("X\_ENTITY\_KEY",

"<*your\_entity\_key*>");

InputStream response = connection.getInputStream();

String resp = IOUtils.toString(response);

System.out.println(resp);

connection.disconnect();

} catch (Exception e) { e.printStackTrace(); }

}

}

**Structure Management**

https://kioskpublicapi.<*cloudlocation>*.com/entity

**Get Entity List**

https://kioskpublicapi.<*cloudlocation>*.com/entity/list

|  |  |  |
| --- | --- | --- |
| **Optional Parameters:** | **Value** | **Example** |
| adminname | (*alphanumeric*) | *ADMINUSER1* |
| kioskname | (*alphanumeric*) | *KIOSKNAME1* |
| entityname | (*alphanumeric*) | *ENTITYNAME1* |
| childmaxlevel | (0-99) | *5* |
| showbalance | (1 / 0) | *1* |
| includefrozen | (1 / 0) | *1* |

Example:

https://kioskpublicapi.*<cloudlocation>*.com/entity/list/entityname/*ENTITYNAME1*/includefrozen/*1*

API Response:

{

"result": [

{

"entityname": "ENTITYNAME1",

"parententityname": "ENTITYPARENT",

"kioskname": "KIOSKNAME1"

}

]

}

**Get Entity Structure**

https://kioskpublicapi.*<cloudlocation>*.com/entity/structure

|  |  |  |
| --- | --- | --- |
| **Optional Parameters:** | **Value** | **Example** |
| adminname | (*alphanumeric*) | *ADMINUSER1* |
| kioskname | (*alphanumeric*) | *KIOSKENTITY1* |
| entityname | (*alphanumeric*) | *ENTITYNAME1* |
| childmaxlevel | (0-99) | *5* |
| showbalance | (1 / 0) | *1* |
| includefrozen | (1 / 0) | *1* |

Example:

https://kioskpublicapi.*<cloudlocation>*.com/entity/structure/entityname/*ENTITYNAME1*/includefrozen/*1*

API Response:

{

"result": {

"entity": {

"ENTITYNAME": "ENTITYNAME1",

"FROZEN": "0"

},

"kiosk": {

"KIOSKNAME": null,

"FROZEN": null

},

"admins": [

{

"ADMINNAME": "ENTITYNAME1",

"DEPOSITBALANCE": "25000",

"ISDEPOSITOWNER": "0",

"ISDEPOSITMANAGER": "1",

"BONUSBALANCE": "0",

"ISBONUSOWNER": "0",

"ISBONUSMANAGER": "0",

"FROZEN": "0"

}

],

"childs": [

{

"entity": {

"ENTITYNAME": "ENTITYNAME1-01",

"FROZEN": "0"

},

"kiosk": {

"KIOSKNAME": "ENTITYNAME1-01",

"FROZEN": "0"

},

"admins": [

{

"ADMINNAME": "ENTITYNAME1-01",

"DEPOSITBALANCE": "9150",

"ISDEPOSITOWNER": "0",

"ISDEPOSITMANAGER": "1",

"BONUSBALANCE": "0",

"ISBONUSOWNER": "0",

"ISBONUSMANAGER": "0",

"FROZEN": "0"

}

],

"childs": []

},

{

"entity": {

"ENTITYNAME": "ENTITYNAME1-02",

"FROZEN": "0"

},

"kiosk": {

"KIOSKNAME": "ENTITYNAME1-02",

"FROZEN": "0"

},

"admins": [

{

"ADMINNAME": "ENTITYNAME1-02",

"DEPOSITBALANCE": ".13",

"ISDEPOSITOWNER": "0",

"ISDEPOSITMANAGER": "1",

"BONUSBALANCE": "0",

"ISBONUSOWNER": "0",

"ISBONUSMANAGER": "0",

"FROZEN": "0"

}

],

"childs": []

]

}

}

**Create New Sub-entity**

https://kioskpublicapi.*<cloudlocation>*.com/entity/create

Example:

https://kioskpublicapi.*<cloudlocation>*.com/entity/create/entityname/*ENTITYNAME\_SUB*/entityname/*ENTITYNAME1*

{

"result": {

"result": "Business entity has been successfully created"

}

}  
*Recommended API Call for Creating Admin:*BaseURI/entity/create/entityname/*<NEW\_ENTITY\_NAME>*/parententityname/*<PARENT\_ENTITY\_NAME>*

**Create Admin Account Inside a Sub-entity**

https://kioskpublicapi.*<cloudlocation>*.com/admin/create

Example:

https://kioskpublicapi.*<cloudlocation>*.com/admin/create/adminname/*ADMINUSER1*/entityname/*ENTITYNAME1*/countrycode/*CN*/password/*qwerty*

API Response:

{

"result": {

"result": "Kiosk admin has been successfully created",

"password": "qwerty"

}

}  
*Recommended API Call for Creating Admin:*BaseURI/admin/create/adminname/*<NEW\_ADMIN\_NAME>*/entityname/*<PARENT\_ENTITY\_NAME>*/countrycode/*<ACCORDING\_TO\_CURRENCY>*/languagecode/*EN*/password/*<your\_password>*/deposit\_limit/*0*/bonus\_limit/*0*/deposit\_limit\_timeperiod/*24*/bonus\_limit\_timeperiod/*24*/is\_deposit\_manager/*1*

**Create Kiosk in an Entity**

https://kioskpublicapi.*<cloudlocation>*.com/kiosk/create

Example:

https://kioskpublicapi.*<cloudlocation>*.com/admin/create/adminname/*ADMINUSER1*/entityname/*ENTITYNAME1*/countrycode/*CN*/password/*qwerty*

API Response:

{

"result": {

"result": "Kiosk has been successfully created"

}

}  
*Recommended API Call for Creating Kiosk in an Entity:*BaseURI/kiosk/create/kioskname/<*NEW\_KIOSK\_NAME>*/entityname/<*PARENT\_ENTITY\_NAME>*/countrycode/<*ACCORDING\_TO\_CURRENCY>*/languagecode/*EN*/frozen/*0*/player\_username\_minlength/*5*/player\_username\_maxlength/*32*/player\_username\_prefix/<*BRANDCODE\_>*/player\_password\_minlength/*6*/player\_password\_maxlength/*40*/def\_viplevel/*1*/passwordchange/*0*

**Increase The Sub-entity Admin’s Balance**

https://kioskpublicapi.*<cloudlocation>*.com/admin/increasebalance

Example:

https://kioskpublicapi.*<cloudlocation>*.com/admin/increasebalance/fromadminname/*ADMINUSER1*/toadminname/*ADMINUSER101*/depositbalanceamount/100

API Response:

{

"result": {

"result": "100 credits successfully added to Deposit balance"

}

}

**Player Management**

https://kioskpublicapi.*<cloudlocation>*.com/player

|  |  |  |
| --- | --- | --- |
| **Mandatory Parameters:** | **Value** | **Example** |
| kioskname | (*alphanumeric*) | *KIOSKNAME1* |
| adminname | (*alphanumeric*) | *ADMINUSER1* |
| playername | (*alphanumeric*) | *BC-PLAYERUSER1* |

Example:

**Get Player List**

https://kioskpublicapi.*<cloudlocation>*.com/player/list/kioskname/*KIOSKNAME1*/adminname/*ADMINUSER1*/page/*1*/perPage/*200*

|  |  |  |
| --- | --- | --- |
| **Mandatory Parameters:** | **Value** | **Example** |
| page | (*integer*) | *1* |
| perPage | (*integer*) | *200* |

API Response:

{

"result": [

{

"PLAYERNAME": "BC-PLAYERUSER1",

"SIGNUPDATE": "2015-03-27",

"VIPLEVEL": "1",

"KIOSKNAME": "DEV1",

"ADMINNAME": "ADMINUSER1",

"FROZEN": "0",

"LASTLOGINDATE": null,

"BALANCE": "0",

"BONUSBALANCE": "0",

"IP": null

}

],

"total": {

"TotalCount": "1",

"TotalBalance": 0,

"TotalBonusBalance": 0

},

"pagination": {

"currentPage": 1,

"totalPages": 1,

"itemsPerPage": "200"

}

}

**Get Player Info**

https://kioskpublicapi.*<cloudlocation>*.com/player/info/playername/*BC-PLAYERUSER1*

API Response:

{

"result": {

"ACCOUNTBUSINESSPHASE": "online",

"ACCOUNTMIGRATED": "false",

"ADDRESS": "NA",

"ADVERTISER": "default73",

"ADVERTISERTYPE": "Affiliate",

"AGEVERIFICATIONSTATUS": "unknown",

"BALANCE": 0,

"BANNERID": "-",

"BIRTHDATE": "1950-01-01",

"BONUSBALANCE": 0,

"CASINOCODE": "1769",

"CASINONAME": "playforreal88",

"CASINONICKNAME": 0,

"CHECKIDDOCUMENT": "false",

"CITY": "NA",

"CLIENTSKIN": "playforreal88",

"CLIENTTYPE": "casino",

"CODE": "12962786",

"COMPPOINTS": 0,

"COUNTRYCODE": "MY",

"CREFERER": 0,

"CURRENCY": "MYR",

"CURRENCYCODE": "MYR",

"CURRENTBET": 0,

"CURRENTBONUSBET": 0,

"CUSTOM01": "PLAY",

"CUSTOM02": "CASINO",

"DISABLEGAMING": "false",

"DISABLEHELDFUNDS": 0,

"EMAIL": "bc-playeruser1@tleluckystonemyr.com",

"FAX": "11111",

"FIRSTNAME": "NA",

"FROZEN": 0,

"GENDER": "M",

"INTERNAL": 0,

"KIOSKADMINCODE": "2057501",

"KIOSKCODE": "1842121",

"LANGUAGECODE": "EN",

"LASTNAME": "NA",

"MARKASADVERTISER": 0,

"MARKASBONUSSEEKER": 0,

"MARKDUPCHECKFAILED": 0,

"MARKEMAILVERIFIED": 0,

"MAXBALANCE": 0,

"NETWORKNICKNAMES": 0,

"NOBONUS": 0,

"PASSWORD": "7F2DF787C643C333376A2FEBD113CB58229B77C7",

"PASSWORDCHANGE": 0,

"PENDINGBONUSBALANCE": 0,

"PHONE": "11111",

"PLAYERCODE": "12962786",

"PLAYERDORMANCY": "false",

"RESERVEDBALANCE": 0,

"RISKPROFILECODE": "3",

"SERIAL": "73120CARD22ER22098531",

"SEX": "M",

"SIGNUPCLIENTPLATFORM": "download",

"SIGNUPCLIENTVERSION": "-",

"SIGNUPDATE": "2015-03-27 14:51:31",

"SIGNUPDELIVERYPLATFORM": "null",

"SIGNUPDEVICEBROWSER": "null",

"SIGNUPKIOSKCODE": "1842121",

"SIGNUPOSNAME": "null",

"SIGNUPOSVERSION": "null",

"STATE": "NA",

"SUSPENDED": 0,

"TOTALCOMPPOINTS": 0,

"TRACKINGID": 0,

"PLAYERNAME": "BC-PLAYERUSER1",

"VIPLEVEL": "1",

"WANTMAIL": 1,

"ZIP": "NA",

"ISINGAME": 0,

"KIOSKNAME": "DEV1",

"KIOSKADMINNAME": "KIOSKNAME1",

"LASTLOGINDATE": false,

"LOCKEDMINUTES": 0

}

}

**Create Player in a Kiosk**

https://kioskpublicapi.*<cloudlocation>*.com/player/create

Example:

https://kioskpublicapi.*<cloudlocation>*.com/player/create/playername/*BC-PLAYERUSER1*/adminname/*ADMINUSER1*/kioskname/*KIOSKENTITY1*/countrycode/*CN*/viplevel/*1*/languagecode/*EN*/password/*qwerty*/

API Response:

{

"result": {

"result": "New player has been created",

"playername": "BC-PLAYERUSER1",

"password": "qwerty",

"executiontime": {

"webapi": "82.901 ms",

"skywind": "0.6089 ms"

}

}

}

*Recommended API Call for Creating Player:*BaseURI/player/create/playername/*<NEW\_PLAYER\_NAME>*/adminname/*<PARENT\_ADMIN\_NAME>*/kioskname/*<PARENT\_KIOSK\_NAME>*/firstname/*NA*/lastname/*NA*/countrycode/*<ACCORDING\_TO\_CURRENCY>*/viplevel/*1*/languagecode/*EN*/password/*<NEW\_PASSWORD>*

**\* Condition of creating player’s username**

**- MUST BE capital letter.**

**- Only A-Z, 0-9, dot (.), comma (,), underscore (\_), hypen (-) and plus (+) are acceptable.**

**- Started with prefix, “<BRANDCODE>-<DESIREDNAME>”. For E.g.: WBET-ROBERT1988**

(you might receive many failure response if “brandcode” is not included.)

**Check Balance**

https://kioskpublicapi.*<cloudlocation>*.com/player/balance/playername/*BC-PLAYERUSER1*

API Response:

{

"result": {

"currencycode": "MYR",

"balance": "0",

"bonusbalance": "0",

"playername": "BC-PLAYERUSER1"

}

}

**Deposit**

https://kioskpublicapi.*<cloudlocation>*.com/player/deposit/playername/*BC-PLAYERUSER1*/amount/*100*/adminname/*ADMINUSER1*

|  |  |  |
| --- | --- | --- |
| **Mandatory Parameters:** | **Value** | **Example** |
| amount | (*integer*) | *100* |

API Response:

{

"result": {

"amount": "100",

"currentplayerbalance": "283.71",

"executiontime": "2631.148 ms",

"externaltransactionid": null,

"instantcashtype": "local",

"kiosktransactionid": "160852998",

"kiosktransactiontime": "2015-04-09 04:15:32",

"ptinternaltransactionid": "1022249419",

"result": "Deposit OK"

}

}

**Withdrawal**

https://kioskpublicapi.*<cloudlocation>*.com/player/withdraw/playername/*BC-PLAYERUSER1*/amount/*100*/adminname/*ADMINUSER1*/externaltranid/*<BRANDCODE>-123456*

|  |  |  |
| --- | --- | --- |
| **Optional Parameters:** | **Value** | **Example** |
| externaltranid | (*alphanumeric*) | *bc-2789100* |

API Response:

{

"result": {

"amount": "100",

"currentplayerbalance": "183.71",

"executiontime": "1948.996 ms",

"externaltransactionid": "bc-2789100",

"instantcashtype": "local",

"kiosktransactionid": "160853602",

"kiosktransactiontime": "2015-04-09 04:18:32",

"ptinternaltransactionid": "1022251530",

"result": "Withdraw OK"

}

}

**Logout (kick out) player**

https://kioskpublicapi.*<cloudlocation>*.com/player/logout/playername/*BC-PLAYERUSER1*

API Response:

{

"result": {

"result": "Logout request has been successfully sent"

}

}

**Check Player Online**

https://kioskpublicapi.*<cloudlocation>*.com/player/online/playername/*BC-PLAYERUSER1*

API Response:

{

"result": {

"result": 0

}

}

**Reset Failed-login**

https://kioskpublicapi.*<cloudlocation>*.com/player/resetfailedlogin/playername/*BC-PLAYERUSER1*

API Response:

{

"result": {

"result": "Failed login attempts count has been reset"

}

}

**Player Games** *(support Gzip compression)*

**Game History (Bet Transactions) – Option 1 – PlayerGames**

https://kioskpublicapi.*<cloudlocation>*.com/customreport/getdata/reportname/PlayerGames/startdate/*2015-03-03%2013:55:00*/enddate/*2015-03-03%2014:00:00*/frozen/*all*/playername/*BC-PLAYERUSER1/*perPage/*5*/gametype/*Slot Machines*/

|  |  |  |
| --- | --- | --- |
| **Mandatory Parameters:** | **Value** | **Example** |
| startdate | yyyy-mm-dd%20hh:mm:ss | *2015-03-03%2013:55:00* |
| enddate | yyyy-mm-dd%20hh:mm:ss | *2015-03-03%2014:00:00* |
| frozen | (yes / no / all) | *all* |

|  |  |  |
| --- | --- | --- |
| **Optional Parameters:** | **Value** | **Example** |
| timeperiod | (now / yesterday / weekstart / monthstart / yearstart) |  |
| playername | (*alphanumeric*) | *BC-PLAYERUSER1* |
| showinfo | (1 / 0) | *1* |
| gametype | (Card Games / Fixed Odds / GTS Games / Keno / Live Games / Live Hi-Lo Game / Mahjong / Mini games / None / Pachinko / PlaytechTV / Progressive Slot Machines / Progressive Video Pokers / Scratchcards / Sidegames / Slot Machines / Table Games / Tournaments / VF Games / VIP Live Games / Video Pokers) | *Slot Machines* |
| page | (*integer*) | *1* |
| perPage | (*integer*) | *5* |

API Response:

{

"result": [

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1791",

“GAMECODE”: “13229679841",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

"BET": "0",

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.22",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:46",

"LIVENETWORK": null,

"RNUM": "1"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1790",

“GAMECODE”: “13229667134",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.22",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:44",

"LIVENETWORK": null,

"RNUM": "2"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1789",

“GAMECODE”: “13229657358",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.27",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:39",

"LIVENETWORK": null,

"RNUM": "3"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1788",

“GAMECODE”: “13229646851",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": ".1",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.32",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:33",

"LIVENETWORK": null,

"RNUM": "4"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1787",

“GAMECODE”: “13229636509",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.27",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:27",

"LIVENETWORK": null,

"RNUM": "5"

}

],

"pagination": {

"currentPage": 1,

"totalPages": 1409,

"itemsPerPage": "5",

"totalCount": 7043

}

}

**Game History (Bet Transactions) – Option 2 – GameFlow**

https://kioskpublicapi.*<cloudlocation>*.com/game/flow/ startdate/*2015-03-03%2013:55:00*/enddate/*2015-03-03%2014:00:00*/perPage/*5*/page/1/showinfo/1

|  |  |  |
| --- | --- | --- |
| **Mandatory Parameters:** | **Value** | **Example** |
| *<none>* | *<none>* | *<none>* |

|  |  |  |
| --- | --- | --- |
| **Optional Parameters:** | **Value** | **Example** |
| startdate | yyyy-mm-dd%20hh:mm:ss | *2015-03-03%2013:55:00* |
| enddate | yyyy-mm-dd%20hh:mm:ss | *2015-03-03%2014:00:00* |
| showinfo | (1 / 0) | *1* |
| page | (*integer*) | *1* |
| perPage | (*integer*) | *5* |

API Response:

{

"result": [

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1791",

“GAMECODE”: “13229679841",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

"BET": "0",

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.22",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:46",

"LIVENETWORK": null,

"RNUM": "1"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1790",

“GAMECODE”: “13229667134",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.22",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:44",

"LIVENETWORK": null,

"RNUM": "2"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1789",

“GAMECODE”: “13229657358",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.27",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:39",

"LIVENETWORK": null,

"RNUM": "3"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1788",

“GAMECODE”: “13229646851",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": ".1",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.32",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:33",

"LIVENETWORK": null,

"RNUM": "4"

},

{

"PLAYERNAME": "BC-PLAYERUSER1",

"WINDOWCODE": "5",

"GAMEID": "1787",

“GAMECODE”: “13229636509",

"GAMETYPE": "Slot Machines",

"GAMENAME": "Amazon Wild (ashamw)",

“SESSIONID”: “278995149",

“BET”: “.5”,

"WIN": "0",

"PROGRESSIVEBET": "0",

"PROGRESSIVEWIN": "0",

"BALANCE": "1047.27",

"CURRENTBET": "0",

“GAMEDATE”: “2015-03-03 13:55:27",

"LIVENETWORK": null,

"RNUM": "5"

}

],

"pagination": {

"currentPage": 1,

"totalPages": 1409,

"itemsPerPage": "5",

"totalCount": 7043

}

}

**Sample Code with GZIP Compression Enabled - PHP**

<?php

ini\_set("zlib.output\_compression", 1);

ob\_start();

/\*\*\* Configuration \*\*\*/

// GZIP only valid for data within 48 hours period

$startDate = urlencode("<start\_date>"); // required paramater E.g.: 2018-12-04 17:35:00

$endDate = urlencode("<end\_date>"); // required parameter E.g.: 2018-12-04 17:37:00

$domain = "<cloudlocation>"; // endpoint's domain

$entitykey = "<your\_entity\_key>"; // entity key provided

$keyfilePath = "<directory\_path>"; // key file directory E.g.: D:\\directory\\

$keyfilename = "<ssl.key>";

$pemfilename = "<ssl.pem>";

$localZipFileName = "<file.gz>"; // file name of your zip file downloaded. E.g.: result.gz

/\*\*\* End Configuration \*\*\*/

$url= "https://kioskpublicapi.$domain.com/customreport/getdata/reportname/PlayerGames/frozen/all/showinfo/1/startdate/$startDate/enddate/$endDate/";

$header = array();

$header[] = "Accept:text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8";

$header[] = "Cache-Control: max-age=0";

$header[] = "Connection: keep-alive";

$header[] = "Keep-Alive:timeout=5, max=100";

$header[] = "Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3";

$header[] = "Accept-Language:es-ES,es;q=0.8";

$header[] = "Accept-Encoding: gzip,deflate";

$header[] = "Pragma: ";

$header[] = "X\_ENTITY\_KEY: " . $entitykey;

$tuCurl= curl\_init();

$fp = fopen (dirname(\_\_FILE\_\_) . '/' . $localZipFileName, 'w+');

curl\_setopt($tuCurl, CURLOPT\_URL, $url);

curl\_setopt($tuCurl, CURLOPT\_PORT , 443);

curl\_setopt($tuCurl, CURLOPT\_VERBOSE, 0);

curl\_setopt($tuCurl, CURLOPT\_HTTPHEADER, $header);

curl\_setopt($tuCurl, CURLOPT\_TIMEOUT, 60000 );

curl\_setopt($tuCurl, CURLOPT\_SSL\_VERIFYPEER, 0);

curl\_setopt($tuCurl, CURLOPT\_SSL\_VERIFYHOST, 0);

curl\_setopt($tuCurl, CURLOPT\_SSLCERT,dirname($keyfilePath) . $keyfilename);

curl\_setopt($tuCurl, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($tuCurl, CURLOPT\_SSLKEY, dirname($keyfilePath) . $pemfilename);

curl\_setopt($tuCurl, CURLOPT\_FOLLOWLOCATION, true);

curl\_setopt($tuCurl, CURLOPT\_FILE, $fp);

$exec = curl\_exec($tuCurl);

curl\_close($tuCurl);

fclose($fp);

?>

**Sample Code with GZIP Compression Enabled - C#**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Net;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

namespace GZIP

{

class Program

{

static void Main(string[] args)

{

/\*\*\* Configuration \*\*\*/

// GZIP only valid for data within 48 hours period

var startDate = "<start\_date>"; // required paramater E.g.: 2018-12-04 17:35:00

var endDate = "<end\_date>"; // required parameter E.g.: 2018-12-04 17:37:00

var domain = "<cloudlocation>"; // endpoint's domain

var entitykey = "<your\_entity\_key>"; // entity key provided

var keyfilePassword = "<cert\_password>"; // certificate's password

var keyfilePath = "<\\file.p12>"; // key file name with directory E.g.: D:\\directory\\key.p12

var localZipPath = "<\\file.gz>"; // file path of your zip file downloaded. E.g.: D:\\GZdirectory\\result.gz;

/\*\*\* End Configuration \*\*\*/

var endpointUrl = string.Format("https://kioskpublicapi.{0}.com/customreport/getdata/reportname/PlayerGames/frozen/all/showinfo/1/startdate/{1}/enddate/{2}",domain,startDate,endDate);

HttpWebRequest Request = (HttpWebRequest)WebRequest.Create(endpointUrl);

HttpWebResponse Response = null;

Request.Accept = "text/html,application/xhtml+xml,application/xml;q=0.9, \*;q=0.8";

Request.Headers.Add("Cache-Control", "max-age=0");

Request.KeepAlive = true;

Request.Headers.Add("Keep-Alive", "timeout=5000, max=100");

Request.Headers.Add("Accept-Charset", "ISO -8859-1,utf-8;q=0.7,\*;q=0.3");

Request.Headers.Add("Accept-Language", "es-ES,es;q=0.8");

Request.Headers.Add("Accept-Encoding: gzip,deflate");

Request.Headers.Add("Pragma", "");

Request.Headers.Add("X\_ENTITY\_KEY", entitykey);

Request.Method = "POST";

string APICert = keyfilePath;

Request.ContentType = "application/x-www-form-urlencoded";

Request.ClientCertificates.Add(new X509Certificate2(@APICert, keyfilePassword, X509KeyStorageFlags.MachineKeySet));

ServicePointManager.ServerCertificateValidationCallback += (sender, certificate, chain, sslPolicyErrors) => true;

byte[] dataStream = Encoding.UTF8.GetBytes("");

Request.ContentLength = dataStream.Length;

Stream reqStreamData = Request.GetRequestStream();

reqStreamData.Write(dataStream, 0, dataStream.Length);

if (reqStreamData != null) reqStreamData.Close();

byte[] b = null;

Response = (HttpWebResponse)Request.GetResponse();

using (Stream stream = Response.GetResponseStream())

using (MemoryStream ms = new MemoryStream())

{

int count = 0;

do

{

byte[] buf = new byte[1024];

count = stream.Read(buf, 0, 1024);

ms.Write(buf, 0, count);

} while (stream.CanRead && count > 0);

b = ms.ToArray();

var xguid = Guid.NewGuid();

using (var fileStream = File.Create(localZipPath))

{

ms.Seek(0, SeekOrigin.Begin);

ms.CopyTo(fileStream);

}

}

}

}

}

**Sample Code with GZIP Compression Enabled - JAVA**

import java.io.\*;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.net.MalformedURLException;

import java.net.URL;

import java.security.KeyStore;

import java.security.KeyStoreException;

import java.security.NoSuchAlgorithmException;

import java.security.SecureRandom;

import java.security.cert.CertificateException;

import java.security.cert.X509Certificate;

import javax.net.ssl.HostnameVerifier;

import javax.net.ssl.HttpsURLConnection;

import javax.net.ssl.KeyManager;

import javax.net.ssl.KeyManagerFactory;

import javax.net.ssl.SSLContext;

import javax.net.ssl.SSLSession;

import javax.net.ssl.TrustManager;

import javax.net.ssl.X509TrustManager;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.DataOutputStream;

import java.util.\*;

import java.net.URLEncoder;

import java.security.cert.Certificate;

import java.io.\*;

import javax.net.ssl.SSLPeerUnverifiedException;

import java.util.zip.GZIPInputStream;

import java.util.zip.GZIPOutputStream;

public class gzip {

public static void main(String[] args){

String keyfilePath = "<\\dir\\key.p12>";

String keyfilePassword = "<key\_password>";

String domain = "<cloudlocation>";

String entitykey = "<your\_entity\_key>";

String localZipPath = "<\\dir\\file.gz>";

String startDate = "<start\_date>"; // required paramater E.g.: 2018-12-04 17:35:00

String endDate = "<end\_date>"; // required parameter E.g.: 2018-12-04 17:37:00

try {

String endpointUrl = "https://kioskpublicapi." + domain + ".com/customreport/getdata/reportname/PlayerGames/frozen/all/showinfo/1/startdate/" + URLEncoder.encode(startDate, "UTF-8") + "/enddate/" + URLEncoder.encode(endDate, "UTF-8") + "/";

KeyStore ks = KeyStore.getInstance("PKCS12");

URL fileURL = new File(keyfilePath).toURI().toURL();

File file = new File(fileURL.getFile());

FileInputStream fis = new FileInputStream(file);

ks.load(fis, keyfilePassword.toCharArray());

//Create KeyManagerFactory using loaded keystore

KeyManagerFactory kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());

kmf.init(ks, keyfilePassword.toCharArray());

KeyManager[] kms = kmf.getKeyManagers();

//Crete TrustManager to bypass trusted certificate check

TrustManager[] trustAllCerts = new TrustManager[] {

new X509TrustManager() {

public java.security.cert.X509Certificate[] getAcceptedIssuers() { return null; }

public void checkClientTrusted(X509Certificate[] certs, String authType) { }

public void checkServerTrusted(X509Certificate[] certs, String authType) { }

}

};

//Hostname verification bypass method

HostnameVerifier allHostsValid = new HostnameVerifier() {

public boolean verify(String hostname, SSLSession session) { return true;}

};

//Set connection properties to use bypass certificate/hostname check methods

SSLContext sslContext = null;

sslContext = SSLContext.getInstance("TLS");

sslContext.init(kms, trustAllCerts, new SecureRandom());

HttpsURLConnection.setDefaultHostnameVerifier(allHostsValid);

HttpsURLConnection.setDefaultSSLSocketFactory(sslContext.getSocketFactory());

HttpsURLConnection.setDefaultHostnameVerifier(new HostnameVerifier() {

public boolean verify(String hostname, SSLSession session) {

if (hostname.equals("localhost")) { return true; }

return false;

}

});

//Send API call together with entity key for validation

HttpsURLConnection connection = (HttpsURLConnection) new URL(endpointUrl).openConnection();

connection.setRequestProperty("X\_ENTITY\_KEY", entitykey);

connection.setRequestMethod("POST");

connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");

connection.setRequestProperty("Connection", "Keep-Alive");

connection.setRequestProperty("Content-length", String.valueOf(""));

connection.setRequestProperty("Accept-Encoding" ,"gzip,deflate");

connection.setDoOutput(true);

int responsecode = connection.getResponseCode();

if (responsecode == 200){

java.io.BufferedInputStream in = new BufferedInputStream(connection.getInputStream());

GZIPInputStream gzin = new GZIPInputStream(in);

File destFile = new File(localZipPath);

java.io.FileOutputStream fos = new FileOutputStream(destFile);

GZIPOutputStream gz = new GZIPOutputStream(fos);

byte data[] = new byte[65536];

int gsize = 0;

while((gsize = gzin.read(data)) != -1) {

gz.write(data, 0, gsize);

}

gz.close();

}

else { System.out.println(connection.getResponseMessage());}

connection.disconnect();

} catch (Exception e) { e.printStackTrace(); }

}

}

**Complete API functions**

Please refer to document “**Playtech API Guide.htm**” in “**API Functions & Back Office.zip**”.

**Player Login**

**Flash Integration**

In order for the login function to work the customer must place a specific file in their website server on a specific path. This file is called integrationRedirect.html and it is part of the login functionality. This file must be located on the same domain and URL as from where the login request is made. For example, if the login function is called from www.yourdomain.com then the integrationRedirect.html file must be located at www.*yourdomain.com*/integration/integrationRedirect.html (integrationRedirect.html can be found in “***Flash Game Integration.zip***”) and included the JavaScript in your login page as below:

<script type="text/javascript" src=" https://login.***<virtualdatabase>***.com/jswrapper/integration.js.php?casino=***<virtualdatabase>***"></script>

Below is the sample JavaScript functions that is used for logging in and logging out of a player (on Flash platform) from the games.

<scripttype="text/javascript">

iapiSetCallout('Login', callloutLogin);

iapiSetCallout('Logout', calloutLogout);

function login(realMode) {

iapiLogin("<*USERNAME*>", "<*password*>", realMode, "en");

}

function logout(allSessions, realMode) {

iapiLogout(allSessions, realMode);

}

functioncalloutLogin(response) {

if (response.errorCode) {

alert("Login failed, " + response.errorText);

}

else {

window.location = "/path/to/lobby/page";

}

}

functioncalloutLogout(response) {

if (response.errorCode) {

alert("Logout failed, " + response.errorCode);

}

else {

alert("Logout OK");

}

}

</script>

//For login function

login(1);

//For logout function

logout(1,1);

**HTML5 Integration**

Please make sure www.*yourdomain.com*/integration/integrationRedirect.html is in place (integrationRedirect.html can be found in “***Flash Game Integration.zip***”).

<script type="text/javascript" src="https://login.<*mobilehub*>.com/jswrapper/integration.js.php?casino=<*virtualdatabase*>"></script>

<script type="text/javascript">

var mobiledomain = "<*mobilehub*>";

var systemidvar = "<*systemID*>";

</script>

<script type="text/javascript">

function loading() {

document.getElementById('superform').style.display = 'none';

document.getElementById('loginbutton').style.visibility = 'hidden';

document.getElementById('loadingimage').style.display = 'block';

login()

}

iapiSetCallout('Login', calloutLogin);

function login() {

iapiSetClientPlatform("mobile&deliveryPlatform=HTML5");

var realMode = 1;

iapiLogin(document.getElementById("loginform").username.value, document.getElementById("loginform").password.value, realMode, "en");

}

function calloutLogin(response) {

if (response.errorCode) {

alert("Login failed. " + response.playerMessage + " Error code: " + response.errorCode);

document.getElementById('superform').style.display = 'block';

document.getElementById('loginbutton').style.visibility = 'visible';

document.getElementById('loadingimage').style.display = 'none';

}

else {

window.location = "lobby.html?username=" + document.getElementById("loginform").username.value;

}

}

function getUrlVars() {

var vars = {};

var parts = window.location.href.replace(/[?&]+([^=&]+)=([^&]\*)/gi, function(m,key,value) {

vars[key] = value;

});

return vars;

}

iapiSetCallout('GetTemporaryAuthenticationToken', calloutGetTemporaryAuthenticationToken);

function askTempandLaunchGame(game) {

currentgame = game;

var realMode = 1;

iapiRequestTemporaryToken(realMode, systemidvar, 'GamePlay');

}

function launchMobileClient(temptoken) {

var clientUrl = 'https://hub.' + mobiledomain + '/igaming/' + '?gameId=' + currentgame + '&real=1' + '&username=' + getUrlVars()["username"] + '&lang=' + document.getElementById("languageselect").gamelanguage.value + '&tempToken=' + temptoken + '&lobby=' + location.href.substring(0,location.href.lastIndexOf('/')+1) + 'lobby.html' + '&support=' + location.href.substring(0,location.href.lastIndexOf('/')+1) + 'support.html' + '&logout=' + location.href.substring(0,location.href.lastIndexOf('/')+1) + 'logout.html' + '&deposit=' + location.href.substring(0,location.href.lastIndexOf('/')+1) + 'deposit.html';

document.location = clientUrl;

}

function calloutGetTemporaryAuthenticationToken(response) {

if (response.errorCode) {

alert("Token failed. " + response.playerMessage + " Error code: " + response.errorCode);

}

else {

launchMobileClient(response.sessionToken.sessionToken);

}

}

</script>

</head>

**Launching Games**

**Play for Real**

You should be able to open the game with the Flash game URL after login to the game successfully as below.

**Flash:** https://cache.download.banner.***<virtualdatabase>****.com*/casinoclient.html?language=***<languagecode>***&game=***<gamecode>***

Example:

https://cache.download.banner.winyourfuture88.com/casinoclient.html?language=en&game=gos

**HTML5:**

https://hub.**<*mobilehub*>**.com/igaming/?gameId=***<game\_ID>***&real=1&username=***<player\_username>***&lang=***<language\_code>***&tempToken=***<temp\_token>***&lobby=***<lobby.html>***&support=***<support.html>***&logout=***<logout.html>***&deposit=***<cashier.html>***

Example:

https://hub.wf179888.com/igaming/?gameId=gos&real=1&username=PLY0102PP&lang=en&tempToken=Q0PJa69Tx\_j5XYtaBO126utQwDCw0KCAYKDY&lobby=htcmd:close&support=htcmd:close&logout=htcmd:close&deposit=htcmd:close

**Play for Fun**

It is an offline mode, for DEMO PURPOSES ONLY. Players does not require to login. All the transactions will not be recorded in system. Closing of the current game window and re-launch it shall refresh the balance amount.

Live games and progressive bets are not supported in Play for Fun mode.

**Flash:** https://cache.download.banner.***<virtualdatabase>****.com*/casinoclient.html?language=***<languagecode>***&game=***<gamecode>***&mode=offline

**HTML5:**

*N/A*

Please refer to the “**Game List - <platform>.xlsx**” for *gamecode*; **Appendix B** for *currencycode* and *languagecode*.

**Jackpot Ticker**

For code sample, please refer to “**Ticker Sample Code.html**”.

Reference:

“**Game Codes for the Jackpot Ticker System.pdf**”

“**Playtech Progressive Jackpot Ticker Specifications.pdf”**

**Marketing Materials**

Download Link 1: (Google Drive)

<https://drive.google.com/folderview?id=0BwmprfFPefbZWXdhZm9BYks5VUE&usp=sharing>

Download Link 2: （alternative for partners based in China Mainland）

[https://mega.nz/#F!IwRWjL4J!yEToM1viQpNJCqy0\_QBOuA](https://mega.nz/%23F!IwRWjL4J!yEToM1viQpNJCqy0_QBOuA)

*在* download link 2 *(Mega.nz) 的压缩档都已经过加密，密码：*ptru1es!

**APPENDIX A**

Server, Entity, & Brand Info

|  |  |  |  |
| --- | --- | --- | --- |
| **cloudlocation** | **virtualdatabase** | **mobilehub** | **systemID** |
|  |  |  |  |
| luckydragon88 | *winforfun88* | *ld176988* | *424* |

|  |  |  |
| --- | --- | --- |
| **TLE** | **TLA** | **Entity Keys** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Brandcode / Prefix:**

**\* Condition of creating player’s username**

**- MUST BE capital letter.**

**- Only A-Z, 0-9, dot (.), comma (,), underscore (\_), hypen (-) and plus (+) are acceptable.**

**- Started with prefix, “<BRANDCODE>-<DESIREDNAME>”. For E.g.: WBET-ROBERT1988**

(you might receive many failure responses if “brandcode” is not included.)

**APPENDIX B**

Language, Country & Currency Code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Corresponding Language** | **Language Code** |  | **Corresponding Language** | **Language Code** |
| Afrikaans | AF |  | **Italian** | **IT** |
| Arabic (Literary) | AR |  | **Japanese** | **JA** |
| Bulgarian | BG |  | **Korean** | **KO** |
| **Chinese (Traditional)** | **CH** |  | Norwegian | NO |
| **Chinese (Simplified)** | **ZH-CN** |  | **Polish** | **PL** |
| **Czech** | **CS** |  | **Portuguese (Brazil)** | **PT-BR** |
| Danish | DA |  | Romanian | RO |
| Dutch | NL |  | **Russian** | **RU** |
| **English** | **EN** |  | Serbian | SR |
| Estonian | ET |  | Slovak | SK |
| Finnish | FI |  | **Spanish** | **ES** |
| **French** | **FR** |  | Swedish | SV |
| German | DE |  | **Thai** | **TH** |
| **Greek** | **EL** |  | **Turkish** | **TR** |
| Hungarian | HU |  |  |  |

**\*** Flash platforms supported language are highlighted in **Blue**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Country** | | **Currency** | | |
| **Name** | **Code** | **Name** | **Code** | **Symbol** |
| Afghanistan | AF | Afghani | AFN | ؋ |
| Albania | AL | Lek | ALL | Lek |
| Australia | AU | Dollar | AUD | $ |
| Bangladesh | BD | Taka | BDT |  |
| Bhutan | BT | Ngultrum | BTN |  |
| Brunei | BN | Dollar | BND | $ |
| Cambodia | KH | Riels | KHR | ៛ |
| China | CN | Yuan Renminbi | CNY | ¥ |
| India | IN | Rupee | INR | ₹ |
| Indonesia | ID | Rupiah | IDR | Rp |
| Iran | IR | Rial | IRR | ﷼ |
| Iraq | IQ | Dinar | IQD |  |
| Israel | IL | Shekel | ILS | ₪ |
| Japan | JP | Yen | JPY | ¥ |
| Macao | MO | Pataca | MOP | MOP |
| Malaysia | MY | Ringgit | MYR | RM |
| Myanmar | MM | Kyat | MMK | K |
| North Korea | KP | Won | KPW | ₩ |
| Pakistan | PK | Rupee | PKR | ₨ |
| South Korea | KR | Won | KRW | ₩ |
| Sri Lanka | LK | Rupee | LKR | ₨ |
| Taiwan | TW | Dollar | TWD | NT$ |
| Thailand | TH | Baht | THB | ฿ |
| Turkey | TR | Lira | TRY | YTL |
| Ukraine | UA | Hryvnia | UAH | ₴ |
| Vietnam | VN | Dong | VND | ₫ |

**----------------------- END OF DOCUMENTS -----------------------**