Playtech Integration Docs

Version 2.3

2020-05

Fix logs

Version	Date	Content	Person
1.0	2020.2	Init	Angus
2.0	2020.4	remove Kiosk Management	Angus
2.1	2020.4	added player/checktransaction	Angus
2.2	2020.4	added success and fail response examples	Angus
2.2	2020.4	Translations	Doremi
2.3	2020.5	Update game launch code	Angus

2.0 Important Notice		
Deleted API	Replacements	Comments
kiosk/create	entity/create	Params have been merged
kiosk/info	entity/info	Params have been merged
kiosk/list	entity/list	Params have been merged
kiosk/update	entity/update	Params have been merged

Table of Content

1. Authorization

Entity Key	Provided by us, please add it to request header as X_ENTITY_KEY
API Certificates	Provided by us, please refer to appendix 6.3 6.4 6.5

2. Player Management

2.1. Create Player

This API is used to create single player.

- Request URL

POST baseURI/player/create

- Help URL

POST baseURI/help/player/create

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player name, prefix included
adminname	string	Yes	1-100	Admin Name
entityname	string	No	1-100	Entity Name
password	string	No	-	Password
custom02	string	No	2-200	Unique code for player, usually set it same as Prefix

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
result	string	New player has been created	-
playername	string	-	Player username
password	string	-	Player password

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.2. Player Balance

This API is used to get player balance

- Request URL

POST baseURI/player/balance

- Help URL

POST baseURI/help/player/balance

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player username，Prefix included

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
playername	string	-	Player username
bonusbalance	float	-	Player bonus balance
balance	float	-	Player balance
currencycode	string	-	ISO currency code

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.3. Change Password

This API is used to change player password.

- Request URL

POST baseURI/player/changepassword

- Help URL

POST baseURI/help/player/changepassword

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player username，Prefix included
password	string	No	-	New Password

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
playername	string	-	Player username

password	string	-	New Password

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.4. Deposit

This API is used to deposit for one player.

- Request URL

POST baseURI/player/deposit

- Help URL

POST baseURI/help/player/deposit

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player username，Prefix included
amount	float	Yes	-	Deposit amount
adminname	string	Yes	1-100	Admin name
externaltranid	string	No	1-100	External transaction ID

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
amount	float	-	Deposit amount
currentplayerbalance	float	-	Player current balance
kiosktransactiontime	date	-	Transaction time
kiosktransactionid	integer	-	Transaction ID
ptinternaltransactionid	integer	-	Internal transaction ID
externaltransactionid	integer	-	External transaction ID
instantcashtype	string	local|api	-
executiontime	string	-	Execution time in millisecond
result	string	Deposit OK	-

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.5. Withdraw

This API is used to withdraw for one player.

- Request URL

POST baseURI/player/withdraw

- Help URL

POST baseURI/help/player/withdraw

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length/Value	Description
playername	string	Yes	5-32	Player username，Prefix included
amount	float	Yes	-	Amount to Withdraw
adminname	string	Yes	1-100	Admin name
isForce	boolean	否	1|0	Force withdraw
externaltranid	string	否	1-100	External transaction ID
losebonus	boolean	否	1|0	In case 'true', Payments will decline all bonuses that are marked to be 'Declined upon withdraw request'.

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
amount	float	-	Withdraw amount
currentplayerbalance	float	-	Player current balance
kiosktransactiontime	date	-	Transaction time
kiosktransactionid	integer	-	Transaction ID
ptinternaltransactionid	integer	-	Internal transaction ID
externaltransactionid	integer	-	External transaction ID
instantcashtype	string	local|api	-
executiontime	string	-	Execution time in millisecond
result	string	Withdraw OK	-

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.6. Frozen Player

This API is used to frozen player; no action permitted after being frozen.

- Request URL

POST baseURI/player/freeze

- Help URL

POST baseURI/help/player/freeze

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player username，Prefix included

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
playername	string	-	Player username
frozen	boolean	1|0	Being frozen or not

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.7. Unfrozen Player

This API is used to unfrozen player.

- Request URL

POST baseURI/player/unfreeze

- Help URL

POST baseURI/help/player/unfreeze

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player username，Prefix included

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
playername	string	-	Player username
frozen	boolean	1|0	Being frozen or not

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.8. Player Status

This API is used to check player status.

- Request URL

POST baseURI/player/online

- Help URL

POST baseURI/help/player/online

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
playername	string	Yes	5-32	Player username，Prefix included

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
result	boolean	1|0	Online / offline status

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.9. Logout Player

This API is used to logout player.

- Request URL

POST baseURI/player/logout

- Help URL

POST baseURI/help/player/logout

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description

playername	string	Yes	5-32	Player username，Prefix included

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
result	string	Logout request has been successfully sent	Result

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.10. Transaction Record

This API is used to check player transaction record

- Request URL

POST baseURI/player/transactions

- Help URL

POST baseURI/help/player/transactions

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length/Value	Description
playername	string	Yes	5-32	playername
last	integer	No	-	Amount of rows to show
type	string	No	deposit|withdraw	Transaction type
status	string	No	approved|declined	Transaction status
transactionbiggerthan	float	No	-	Transactions with amount greater than this value
transactionsmallerthan	float	No	-	Transactions with amount less than this value
remoteip	string	No	-	Player IP
showmethod	boolean	No	1|0	Show method
showcomments	boolean	No	1|0	Show comments
page	integer	No	-	Page
perPage	integer	No	-	Perpage

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
ENTITYNAME	string	-	Entity Name
KIOSKNAME	string	-	Kiosk Name
PLAYERNAME	string	-	Player Name
TRANSACTIONTIME	date	-	Transaction time
AMOUNT	float	-	Amount
TYPE	string	deposit|withdraw	Type
STATUS	string	approved|declined	Status
REMOTEIP	string	-	IP
EXTERNALTRANSACTIONID	string	-	External transaction ID
KIOSKTRANSACTIONID	string	-	Transaction Id
COMMENTS	string	-	Comments
METHOD	string	-	Method

total	array	-	Total list

Name	Type	Value	Description
TotalCount	integer	-	Total count

pagination	array	-	The list of fields of the response, see the table below.

Name	Type	Value	Description
currentPage	integer	-	current Page
totalPages	integer	-	total Pages
itemsPerPage	integer	-	items PerPage

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

2.11. Transaction State

This API is used to check player transaction state

- Request URL

POST baseURI/player/checktransaction

- Help URL

POST baseURI/help/player/checktransaction

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
externaltransactionid	string	Yes	-	External transaction

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
status	string	approved|declined| missing|waiting|notallowed	Status
externaltransactionid	integer	-	External transaction
executiontime	string	-	Execution time in millisecond
amount	float	-	amount
currentplayerbalance	float	-	Current player balance
kiosktransactiontime	date	-	Transaction time
kiosktransactionid	string	-	Transaction ID
ptinternaltransactionid	string	-	Internal transaction ID
ip	string	-	IP

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

3. Entity Management

3.1. Entity Information

This API is used to get entity information.

- Request URL

POST baseURI/entity/info

- Help URL

POST baseURI/help/entity/info

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length/Value	Description
entityname	string	Yes	1-100	Entity Name
with3RDPData	boolean	No	1|0	Show 3RDP container data

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
casino	string	-	Casino Name
parententityname	string	-	Parent business entity name
entityname	string	-	Entity Name
frozen	boolean	1|0	Frozen
newplayercustomfieldname	string	custom02-20	Custom field name for new player
newplayercustomfieldvalue	string	-	Custom field value for new player
currency	string	-	currency
revenueshare	float	-	-
payablebalance	float	-	-
freeSpinBalance	integer	-	-
description	string	-	Description
entitykey	string	-	Business entity key
restrictedcountry	string	-	Restricted country code
rolling_commission	array	-	-
language	string	-	language
country	string	-	country
city	string	-	city
address	string	-	address
phone	string	-	phone
email	string	-	email
state	string	-	state
zip	string	-	zip
mobile	string	-	mobile
fax	string	-	fax
askpasswordonaction	boolean	1|0	-
cashoutreceiptenabled	boolean	1|0	-
confirmdepositaction	boolean	1|0	-
confirmnewplayeraction	boolean	1|0	-
playerusernameminlength	integer	-	Minimum length of the player username
playerusernamemaxlength	integer	-	Maximum length of the player username
playerusernameprefix	string	-	Prefix of the players usernames
playerusernameonlynumbers	boolean	1|0	Allow only numbers in player usernames
playerpasswordminlength	integer	-	Minimum length of the player password
playerpasswordmaxlength	integer	-	Maximum length of the player password
playerpwdfromtoplevelent	boolean	1|0	-
defaultplayerpassword	string	-	Kiosk specific hardcoded player password

unfriendlyplayerpasswords	boolean	1|0	Use only numbers when generating player passwords
defviplevel	integer	-	Default VIP level for new players
passwordchange	boolean	1-0	Player should change password on login

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

3.2. Get Entity Broken Game

This API is used to get entity broken game.

- Request URL

POST baseURI/entity/brokengames

- Help URL

POST baseURI/help/entity/brokengames

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length/Value	Description
entityname	string	No	1-100	entityname
playername	string	No	5-32	Player username，Prefix included
brokengametype	string	No	-	Broken game type
status	string	No	reverted|completed|waiting	status
last	integer	No	-	Amount of rows to show
clienttype	string	No	-	Client type
game	string	No	-	Game Name
betbiggerthan	float	No	-	Games with bets greater than this value
betsmallerthan	float	No	-	Games with bets less than this value
remoteip	string	No	-	IP

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
PLAYERNAME	string	-	Player username
GAMEDATE	date	-	Game type
CLIENTTYPE	string	-	Client type
PLATFORM	string	-	Platform
GAMENAME	string	-	Game Name
SHORTNAME	string	-	Game short name
BET	float	-	Player's bet
REMOTEIP	string	-	IP
BROCKENGAMETYPE	string	-	Type of broken game
BROCKENGAMECODE	string	unique	Code of broken game
STATUS	string	reverted|completed|waiting	Brocken game status
FINISHEDGAMECODE	string	-	Finished game code
FINISHEDGAMEDATE	string	-	Finished game date

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg
errorcode	integer	-	Error code，please refer to Help URL

4. Game Record

4.1. Game Record (Game Flow)

This API is used to get game record, maximum delay is 30 seconds.

- Request URL

POST baseURI/game/flow

- Help URL

POST baseURI/help/game/flow

- Request Params (application/x-www-form-urlencoded)

Name	Type	Mandatory	Length	Description
exitgame	boolean	No	1|0	(0 1) is exit game events have to be shown
showdetailedinfo	boolean	No	1|0	Encoded game details information

showbonustype	boolean	No	1|0	4 - Free spin; 5 - Golden chips
excludezero	boolean	No	1|0	(0 1) is games with zero bet and win have to be shown
progressiveonly	boolean	No	1|0	(0 1) is only progressive games have to be shown
startdate	date	No	-	Report start date in yyyy-mm-dd hh:ii:ss format
enddate	date	No	-	Report end date in yyyy-mm-dd hh:ii:ss format
clientinfo	boolean	No	1|0	(0 1) is client type and platform have to be shown
page	integer	No	-	page
perPage	integer	No	-	perPage

- Success Response （JSON）

Name	Type	Value	Description
result	array	-	The list of fields of the response

Name	Type	Value	Description
PLAYERNAME	string	-	Player username
WINDOWCODE	integer	-	Window code
GAMEID	integer	-	In session game id
GAMECODE	integer	Unique	Code of the game
GAMETYPE	string	-	Game type
GAMENAME	string	-	Name of the game
SESSIONID	integer	-	Gaming session id
BET	float	-	Player bet
WIN	float	-	Player win
PROGRESSIVEBET	float	-	Progressive share part of the bet
PROGRESSIVEWIN	float	-	Progressive win
BALANCE	float	-	Player balance
CURRENTBET	float	-	Current bet
GAMEDATE	date		Date and time of the game
LIVENETWORK	string		Live network name

pagination	array	-	The list of fields of the response, see the table below.

Name	Type	Value	Description
currentPage	integer	-	Current Page
totalPages	integer	-	Total Pages
itemsPerPage	integer	-	Items Per Page

- Fail Response （JSON）

Name	Type	Value	Description
error	string	-	Error msg

errorcode	integer	-	Error code，please refer to Help URL

5. Open Game

Get open game URL should achieve in front-end not via API
Request parameter list：

parameter	Value	Description
username	-	Player PT Username should be capital letter with prefix.
password	-	Player password
game	-	Gmae code
lang	-	Language please use appendix 6.1
client	ngm_desktop ngm_mobile live_desk live_mob	live game use live, other games use ngm.
mode	real offline	real(real money) offline(play for fun)

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample Code for Game Launching</title>
</head>

<body>
    <!-- We do not support testing games via localhost -->
    <!-- Please host this html file on your server and assign one domain with it,
    remeber that domain needs to be whitelisted otherwise you will get error code 6 -->
    <!-- You can try testing games with fun mode.it doesnot require login.-->
    <div>
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" placeholder="Must be capital letters"><br><br>
        <label for="password">Password:</label>
        <input type="text" id="password" name="password"><br><br>
        <label for="lang">Language:</label>
```

```
    <input type="text" id="lang" name="lang" placeholder="en,zh-CN,...."><br><br>
    <label for="client">Client:</label>
    <input type="text" id="client" name="client" placeholder="ngm_desktop,ngm_mobile,live_desk,live_mob"
      size="50"><br><br>
    <label for="mode">Mode:</label>
    <input type="text" id="mode" name="mode" placeholder="real,offline"><br><br>
    <label for="game">Game Code:</label>
    <input type="text" id="game" name="game"><br><br>
    <button onclick="login()">Login and Launch</button>
  </div>
  <script>
    function login() {
      // Get variables
      let username = document.getElementById("username").value;
      let password = document.getElementById("password").value;
      let lang = document.getElementById("lang").value;
      let mode = document.getElementById("mode").value;

      if (mode == 'real') {
        iapiSetClientType('casino');
        iapiSetClientPlatform('web');
        iapiLogin(username, password, 1, lang);
      } else {
        // mode is offline, which does not require login. NOTE: only supports client with ngm_desktop and
ngm_mobile
        launchGameWithFunMode();
      }
    }

    function launchGame() {
      // Get variables
      let client = document.getElementById("client").value;
      let mode = document.getElementById("mode").value;
      let game = document.getElementById("game").value;
      let lang = document.getElementById("lang").value;
      let real = (mode == 'real') ? 1 : 0;
      // Optional Variables
      let lobbyUrl = '';
      let logoutUrl = '';
```

```
        let supportUrl = '';
        let depositUrl = '';

        // Slots,Table Games and other non-live games
        if (client == 'ngm_desktop' || client == 'ngm_mobile') {
            iapiSetClientParams(client, 'language=' + lang + '&real=' + real + '&lobby=' + lobbyUrl + '&logout=' +
logoutUrl + '&deposit=' + depositUrl + '&support=' + supportUrl);
            iapiLaunchClient(client, game, mode, '_self');
        }

        // Live Games
        if (client == 'live_desk' || client == 'live_mob') {
            iapiSetClientParams(client, '&launch_alias=' + game + '&language=' + lang + '&real=' + real + '&lobby='
+ lobbyUrl + '&logout=' + logoutUrl + '&deposit=' + depositUrl + '&support=' + supportUrl);
            iapiLaunchClient(client, null, mode, '_self');
        }
    }

    function launchGameWithFunMode() {
        // Get variables
        let client = document.getElementById("client").value;
        let game = document.getElementById("game").value;
        let lang = document.getElementById("lang").value;
        let mode = document.getElementById("mode").value;

        if (client == 'ngm_desktop' || client == 'ngm_mobile') {
            iapiSetClientParams(client, 'language=' + lang + '&real=0');
            iapiLaunchClient(client, game, mode, '_self');
        }
    }

    function calloutLogin(response) {
        if (response.errorCode) {
            // Login failed
            if (response.errorCode == 48) {
                alert('Login failed, error: ' + response.errorCode + ' playerMessage: ' +
response.actions.PlayerActionShowMessage[0].message);
            } else {
                alert('Login failed, error: ' + response.errorCode + ' playerMessage: ' + response.playerMessage);
```

```
            }
        } else {
            // Login success
            launchGame();
        }
    }
</script>
<script>
    // Load JS file
    let script = document.createElement('script');
    script.setAttribute('src', 'https://login-am.hotspin88.com/jswrapper/hotspin88am/integration.js');
    document.head.appendChild(script);
    // Set up callback after JS file is loaded
    script.onload = () => {
        iapiSetCallout('Login', calloutLogin);
    }
</script>
</body>

</html>
```

6. Appendix

6.1. Language List

代码	语言
BG	Bulgarian
CH	Traditional Chinese
CS	Czech
DA	Danish
DE	German
EL	Greek
EN	English
ES	Spanish
ES-MX	Spanish (Mexico)
ET	Estonian
FI	Suomi
FR	French
IT	Italian
JA	Japanese
KO	Korean
MS	Malay

NL	Dutch
PL	Polish
PT	Portuguese
RO	Romanian
RU	Russian
SK	Slovak
SV	Swedish
TH	Thai
ZH-CN	Simplified Chinese

6.2. API List

URL	Description
Admin Management	
admin/accesslist	Get admin access list
admin/checkpassword	Check admin password
admin/decreasebalance	Admin withdraw
admin/decreasebonusbalance	Admin withdraw(bonus)
admin/fullaccesslist	Check admin permission list
admin/increasebalance	Admin deposit
admin/info	Check admin information
admin/list	Get admin list
admin/login	Admin Login
admin/update	Update admin
Report Management	
customreport/getallreports	Get available report
customreport/getdata	Get report data
customreport/getdecorators	
customreport/getform	Get report description
customreport/getreportweight	
Entity Management	
entity/brokengames	Entity incomplete game
entity/cachedrop	Clean entity caching
entity/create	Create entity
entity/freeze	Lock entity
entity/generatekey	Generate entity keyentity/info
entity/info	Get entity information
entity/list	Get entity list
entity/listblockedips	Get entity locked IP
entity/peak	Get entity player peak information
entity/structure	Entity structure
entity/unblockips	Unlock entity locked IP
entity/unfreeze	Unlock entity
entity/update	Update entity
Player Management	
player/balance	Player balance
player/bonustemplates	Player bonus example

player/bonustransactions	Player bonus record
player/brokengames	Player incomplete game
player/cancelbonus	Cancel player bonus
player/changepassword	Reset player password
player/checktoken	Check player token
player/checktransaction	Check player transaction record
player/clearcommission	Clean player commission
player/compensation	Player compensation
player/create	Create player
player/deposit	Player deposit
player/freeze	Lock Player
player/games	Player game record
player/getactivebonuses	Get bonus
player/givebonus	Give player bonus
player/info	Player information
player/list	Player list
player/logerror	Player error record
player/logout	Player log out
player/massclearcommission	Player bulk clear commission
player/masscreate	Player bulk create
player/massdeposit	Player bulk deposit
player/massfreeze	Player bulk lock
player/masslogout	Player bulk logout
player/massmove	Player bulk move
player/masspaycommission	Player bulk pay commission
player/massupdate	Player bulk update
player/masswithdraw	Player bulk withdraw
player/move	Move player
player/online	Check player state
player/paycommission	Pay commission
player/redeembonus	Redeem bonus
player/resetfailedlogin	Reset fail loginplayer/revertbrokengame
player/revertbrokengame	Revert broken game
player/serials	Get player serials
player/sessions	Player sessions
player/transactions	Player Transaction record
player/unfreeze	Unlock Player
player/update	Update Player
player/validplayername	Check Player Username
player/withdraw	Player withdraw
player/withdrawall	Withdraw all
Game Management	
game/flow	Game record 1
game/stream	Game record 2

6.3. Encryption script – PHP

```php
<?php
$path = dirname(__FILE__);

    $url= "<baseURI>";
    $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);

    print_r($data);
?>
```

6.4. Encryption script – C#

```
private void callAPI () {
    string rURL = "baseURI";

    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;
  }
}
```

6.5. Encryption script – JAVA

```
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 {

   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("baseURI").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(); }
  }


}
```