| FHI Casino integration manual  Version 0.2 |
| --- |

09 / 2014

## Purpose and Overview

### Document purpose

The purpose of this document is to describe the requirements and process of integrating casino games to a 3rd party system.

### Overview of topics

This document will cover the below topics:

* Contact details, integration servers and websites
* Client graphics and customization
* Integration API and functions
* Game limits and VIP levels
* Additional documents

## Contact details, integration servers and websites

### Contact details form

In order for us to serve you better please fill in the below contact details form.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | E-mail | Skype | Position/Role |
| Support | support@akateksolutions.com |  | Akatek 24/7 Support Team |
| Martin Roots | martin@akateksolutions.com |  | Akatek Technical Consultant |
| Gladys Mendoza | gladys@akateksolutions.com |  | Account Executive and Customer Support Manager |
| Catherine Wang | catherine@akateksolutions.com | catcatty89 | Akatek Account Executive |
| Support | Please fill in | Fill in | Please fill in |
| PM | Project manager, fill in | Fill in | Please fill in |
| Operations | Operations contact, fill in | Fill in | Please fill in |
| Developer | Developer contract, fill in | Fill in | Please fill in |

### Customer integration system details

It is required to document in our system all IP addresses for Customer side websites for flash platform to work. In addition, IP addresses of customer offices and integration servers might be needed to be white listed by us for the games to work in restricted countries. Please fill in the below form with the required details.

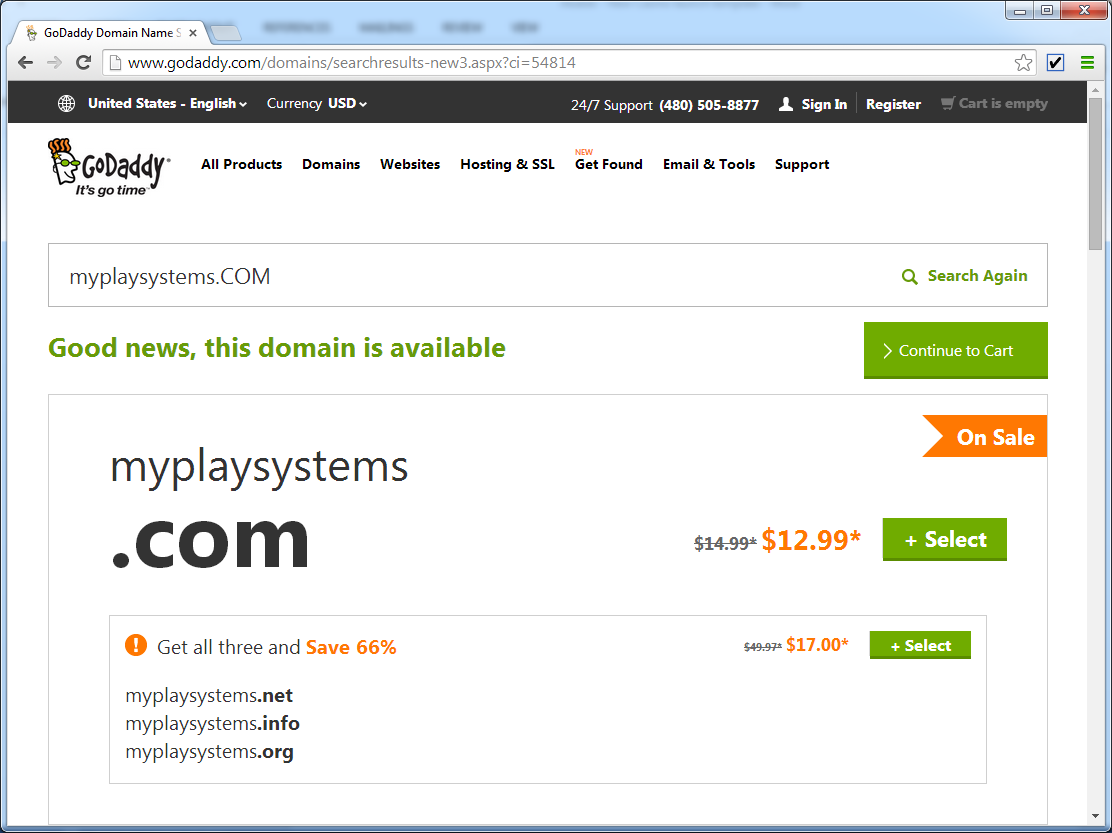
【please just fill up the Doc. Called “FHI First Kick-off Meeting”】

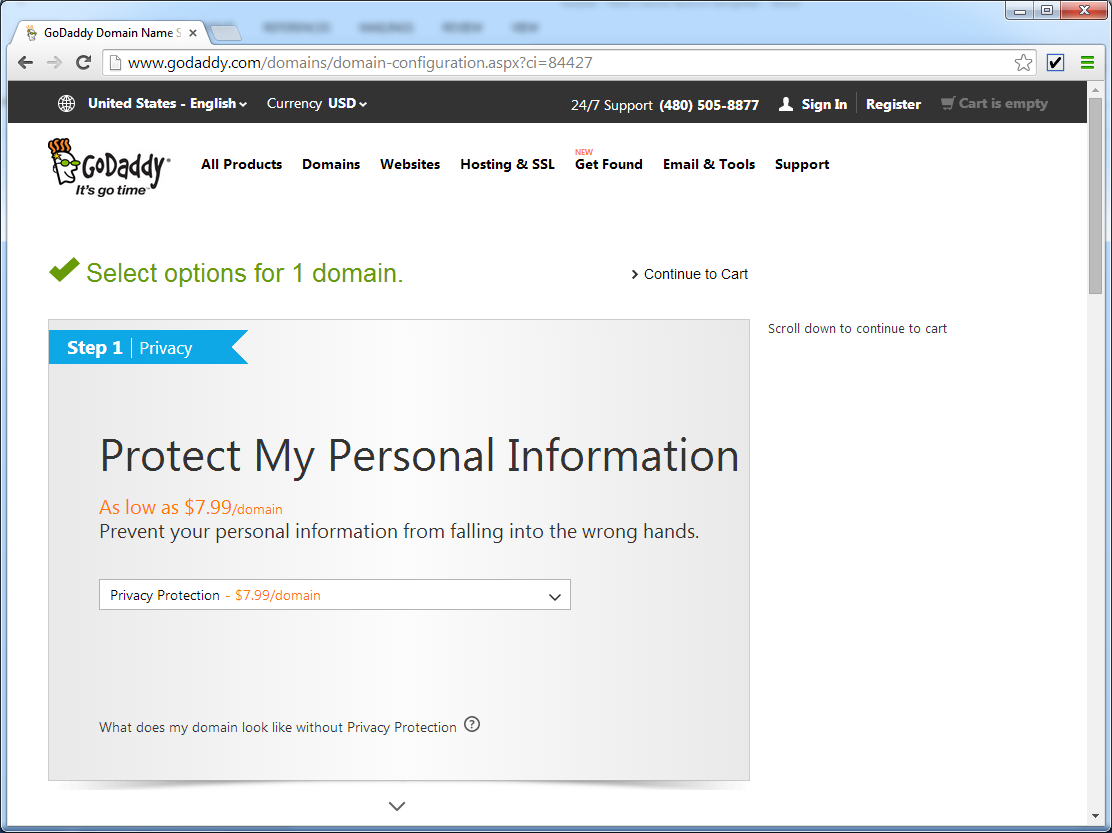
## Domain name and SSL certificate requirements

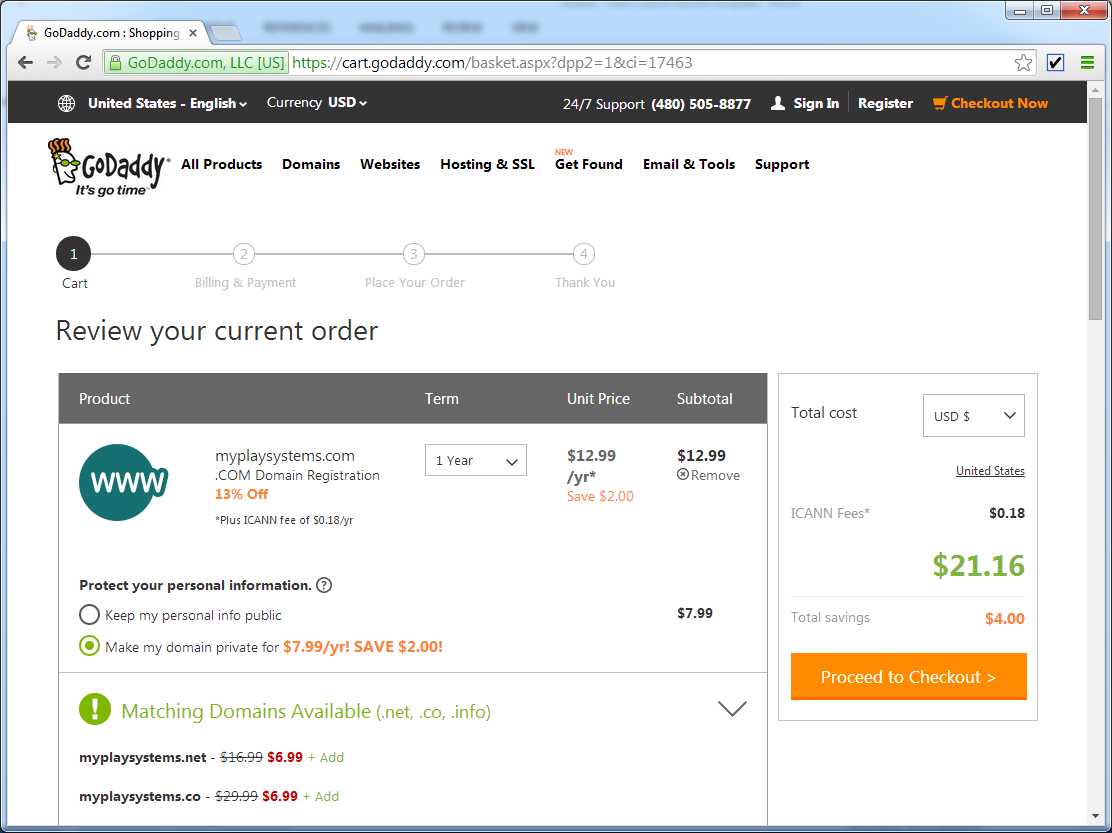
We will provide you with a download link for the Windows and Android casino client using your own domain, we will also use this domain to grant you access to our existing admin system. In order for us to be able to do so it is required that you purchase and delegate to our domain servers the management of a new domain on top of what our engineers will configure your services that connect to our system.

### Purchase a new domain with domain privacy

We recommend that you purchase a new domain from Go Daddy with Domain Privacy: <http://www.godaddy.com/>



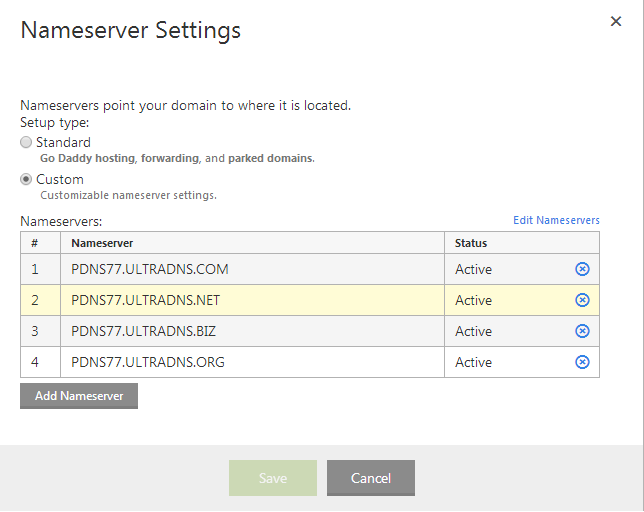




### Purchase a new domain with domain privacy

Once the domain with privacy is purchased, delegation of the domain DNS servers must be done to our UltraDNS managed account. Please change the DNS servers for the domain to the below:

* pdns77.ultradns.com
* pdns77.ultradns.net
* pdns77.ultradns.biz
* pdns77.ultradns.org



**Provide host SSL certificate for Kiosk Admin access**

**We would like you to purchase an SSL certificate for the sub domain Kiosk.FHI.club. We will use this certificate to secure access to your Admin that we will create for you.**

**Please provide us the below details for me to generate the CSR file.**

**City:**

**State:**

**Country:**

### Purchase an SSL certificate for kiosk.yourdomain.com

We require you to purchase the SSL certificate to secure access to our existing admin system. We will provide you the CSR and an Apace+SSL certificate must be purchased and sent to us. We recommend purchasing the certificate from InstantSSL: <http://www.instantssl.com/ssl-certificate-products/ssl/ssl-certificate-essentialssl.html>

https://www.namecheap.com/

## Client graphics and customization

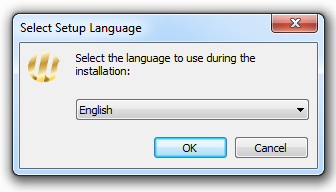
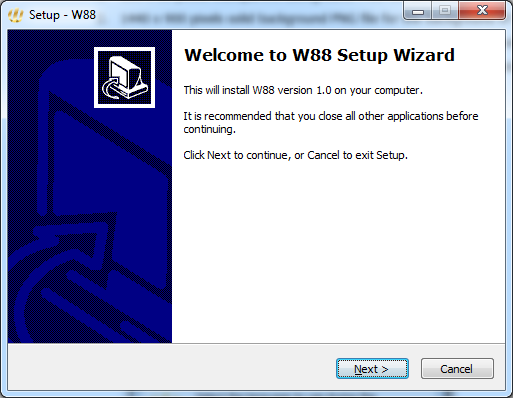
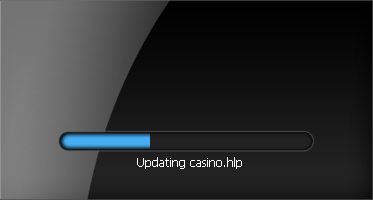
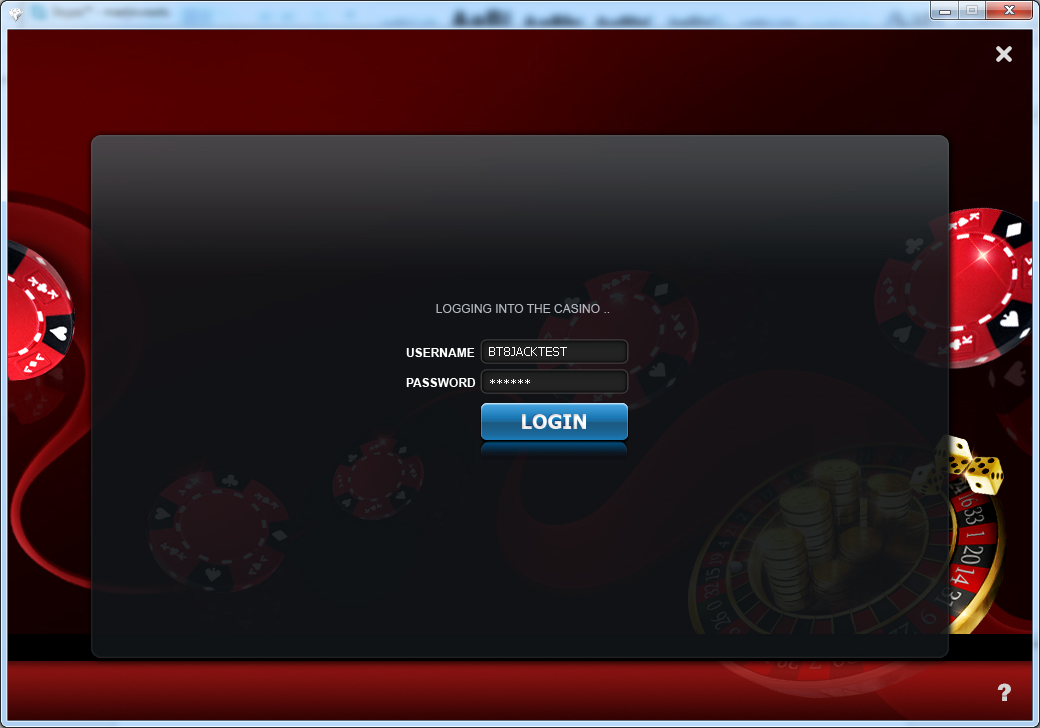
We are offering the possibility to customize the below in our download client:

1. Customers logo in the download client lobby
2. Customers background in the download client lobby
3. Customers logo as the desktop icon
4. Customers logo in the casino installer icon
5. Customers logo in the casino installer menu
6. Customers logo in the casino installer wizard

In order for us to facilitate the graphics we will need to receive from the customer the below images:

1. 230 x 86 pixels transparent background PNG file for the logo in the lobby
2. 1440 x 900 pixels solid background PNG file for the background of the lobby
3. 48 x 48 pixels transparent or solid background PNG file for the desktop icon
4. 48 x 48 pixels transparent or solid background PNG file for the installer icon
5. 165 x 315 pixels solid background PNG file for the installer menu
6. 55 x 58 pixels solid background PNG file for the installer wizard

Please see below the examples of the images:

1. Icon  
     
   
2. Installer language selection  
     
   
3. Installer with the customizable wizard image  
     
   
4. Customized desktop icon  
     
   
5. Default, non-customizable client download screen  
     
   
6. Default, non-customizable client login screen  
     
   
7. Customized logo (green highlight) and background (red highlight) in the casino lobby  
     
   

## Data integration and reporting functions

We are using an HTTP based integration API system where all integration API calls are sent to our system from the customer’s environment over SSL with using private SSL keys. Our system utilizes a multi-level security model and automated internal security mechanisms to enhance the integrity and validity of our environment.

The private SSL keys for the integration API will be provided to you separately upon the completion of activating the integration and management accounts.

### Integration API server

**Integration server URL:**  https://kioskpublicapi.mightypanda88.com/

The API calls must be done with the private SSL certificate issued to the licensee and the licensee must use the assigned Entity Key in the HTTP header to make the API calls.

### Integration API functions

The full list of functions and parameters will be provided to you by your account executive.

**Create player:** player/create/playername/FHICNY001USER/adminname/FHICNY001/kioskname/FHICNY001/

**Get player info:** player/info/playername/FHICNY001USER

**Update player:** player/update/playername/FHICNY001USER/password/test11

**Is player online:** player/online/playername/FHICNY001USER

**Deposit:** player/deposit/playername/FHICNY001USER/amount/1/adminname/FHICNY001/externaltranid/D123456

**Withdraw:** player/withdraw/playername/FHICNY001USER/amount/1/adminname/FHICNY001/externaltranid/W123456

**Check transaction:** player/checktransaction/externaltransactionid/W123456

**Logout Player:** player/logout/playername/FHICNY001USER

**Reset failed login attempts:** player/resetFailedLogin/playername/ FHICNY001USER

**Get players:** player/list/kioskadmin/FHICNY001

**Generate key:** entity/generatekey/entityname/FHICNY001

**This is the key:**

**663d1d180aab766e5d73ea84da670684c31cc1af2f46674cd63c4b64d0f77625c99a2ac3ba456759d88bd096b3794ce16650f63e3431a11f4a15f1dd19a4ba37**

TLE KEY: 52290b49153224bdebc54964c53ae0c145d0faabc749bed33b581ac1481134ea4ccd39eb39f2e64cce43d838db24660a6b0013ed33d28783b456c599d19f2cc9

### 

### Admin system access details

**Kiosk management URL:**  kiosk.myogph.net

**Kiosk admin user name:** FHICNYTLA

**Kiosk admin password:** test11

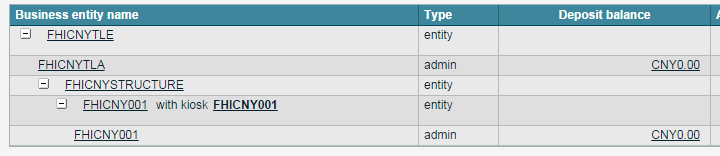
## Business structure management

The management of business structures in our system is the responsibility of the licensee. We will only create the below for the licensee:  
  
1. Top Level Entity that will hold all licensee data

2. Top Level Admin that will hold all the licensee funds

3. A sub-entity to hold all of the B2B sites for the licensee

Please see below for your structure, should the name of the Licensee be WBGSOFTCNYTLE and the currency will be CNY, the structure created by for the licensee will look like this:



FHICNYTLE will be the Top Level Entity for the CNY currency structure for customer FHI. This entity will hold all of the licensee’s data for CNY currency and will be set up by us.

FHICNYTLA will be the Top Level Admin for the CNY currency structure for customer FHI. This admin will hold all of the licensee’s depositing power, we will only fund this account and we will not fund/touch any other admin accounts in the system.

### Displaying the business structure over API call

To see the structure using the Kiosk API call the below call should be made:

***Example API input code:***

URL: https://kioskpublicapi.mightypanda88.com/entity/structure

Do not forget to add the entity key to the HTTP header. In this example the Top Level Entity key is used together with the certificate thus all of the data in the structure is exposed.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

**The API will return JSON as response showing the business structure.**

{

"result": {

"entity": {

"ENTITYNAME": "FHICNYTLE",

"FROZEN": "0"

},

"kiosk": {

"KIOSKNAME": null,

"FROZEN": null

},

"admins": [

{

"ADMINNAME": "FHICNYTLA",

"DEPOSITBALANCE": "0",

"ISDEPOSITOWNER": "0",

"ISDEPOSITMANAGER": "0",

"BONUSBALANCE": "0",

"ISBONUSOWNER": "0",

"ISBONUSMANAGER": "0",

"FROZEN": "0"

}

],

"childs": [

{

"entity": {

"ENTITYNAME": "FHICNYSTRUCTURE",

"FROZEN": "0"

},

"kiosk": {

"KIOSKNAME": null,

"FROZEN": null

},

"admins": [],

"childs": [

{

"entity": {

"ENTITYNAME": "FHICNY001",

"FROZEN": "0"

},

"kiosk": {

"KIOSKNAME": "FHICNY001",

"FROZEN": "0"

},

"admins": [

{

"ADMINNAME": "FHICNY001",

"DEPOSITBALANCE": "0",

"ISDEPOSITOWNER": "0",

"ISDEPOSITMANAGER": "0",

"BONUSBALANCE": "0",

"ISBONUSOWNER": "0",

"ISBONUSMANAGER": "0",

"FROZEN": "0"

}

],

"childs": []

}

]

}

]

}

}

### Creating a new sub-structure

The API offers the licensee the possibility to create sub-structures, each of the sub-structures should belong to a specific B2B customer of the licensee. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To create a new sub-entity using the Kiosk API the below call should be made:

***Example API input code:***

URL:https://kioskpublicapi.mightypanda88.com/entity/create/entityname/FHICNY001/parententityname/FHICNYSTRUCTURE

Do not forget to add the entity key to the HTTP header. In this example the Top Level Entity key is used together with the certificate thus all of the data in the structure is exposed.

*Accept:text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8,Cache-Control: max-age=0,Connection: keep-alive,Keep-Alive:timeout=5, max=100,Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3,Accept-Language:es-ES,es;q=0.8,Pragma: ,X\_ENTITY\_KEY: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the sub-entity is created:***

{

"result": "Business entity is successful created"

}

### Generating entity key for Kiosk API access for a sub-entity

In order for the owner of the sub-entity to have secure API access they need to be given the Kiosk API integration SSL certificate generated by us and the licensee must generate the entity key for the owner of the sub-entity.

To create a new sub-entity key using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/entity/generatekey/name/FHICNY001*

Do not forget to add the entity key to the HTTP header. In this example the Top Level Entity key is used together with the certificate thus all of the data in the structure is exposed.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the sub-entity key is created:***

{

"key": "663d1d180aab766e5d73ea84da670684c31cc1af2f46674cd63c4b64d0f77625c99a2ac3ba456759d88bd096b3794ce16650f63e3431a11f4a15f1dd19a4ba37"

}

### Show the sub-entity structure

It is recommended to verify that the sub-entity key will work. In order to do this the licensee must issue any API call with the new sub-entity key in the HTTP header instead of the Top Level Entity key in the HTTP header.

To verify a new sub-entity key as functional the Kiosk API the below call should be made:

***Example API input code:***

*URL: https://kioskpublicapi.mightypanda88.com/entity/structure/entityname/FHICNY001*

Do not forget to add the new sub-entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate thus only the sub-entity data is exposed.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the EMPERORCNY sub-entity structure only***

*{*

*"result": {*

*"entity": {*

*"ENTITYNAME": "FHICNY001",*

*"FROZEN": "0"*

*},*

*"kiosk": {*

*"KIOSKNAME": "FHICNY001",*

*"FROZEN": "0"*

*},*

*"admins": [*

*{*

*"ADMINNAME": "FHICNY001",*

*"DEPOSITBALANCE": "0",*

*"ISDEPOSITOWNER": "0",*

*"ISDEPOSITMANAGER": "0",*

*"BONUSBALANCE": "0",*

*"ISBONUSOWNER": "0",*

*"ISBONUSMANAGER": "0",*

*"FROZEN": "0"*

*}*

*],*

*"childs": []*

*}*

*}*

### Create an admin account inside a sub-entity

The admin accounts in our system are used to hold the funds for different entities and sub-entities. The admin accounts are also direct owners and managers of the player accounts inside sub-entities and must be present in every sub-entity created. A number of parameters are mandatory to create a kiosk admin due to the need to have backwards compatibility with the old Kiosk Admin interfaces.

To create a new admin using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/admin/create/adminname/FHICNY001/entityname/ FHICNY001/countrycode/CN/password/test11*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate thus only the sub-entity data is exposed.

*Accept:text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8,Cache-Control: max-age=0,Connection: keep-alive,Keep-Alive:timeout=5, max=100,Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3,Accept-Language:es-ES,es;q=0.8,Pragma: ,X\_ENTITY\_KEY:* *083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the admin as created:***

{

"result": "Kiosk admin is successfully created",

"password": "test11"

}

### Show the list of admins inside the structure

To see the structure using the Kiosk API call the below call should be made:

***Example API input code:***

*URL: https://kioskpublicapi.mightypanda88.com/entity/structure/*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate thus only the sub-entity data is exposed.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the entity structure:***

*{*

*"result": {*

*"entity": {*

*"ENTITYNAME": "FHICNYTLE",*

*"FROZEN": "0"*

*},*

*"kiosk": {*

*"KIOSKNAME": null,*

*"FROZEN": null*

*},*

*"admins": [*

*{*

*"ADMINNAME": "FHICNYTLA",*

*"DEPOSITBALANCE": "0",*

*"ISDEPOSITOWNER": "0",*

*"ISDEPOSITMANAGER": "0",*

*"BONUSBALANCE": "0",*

*"ISBONUSOWNER": "0",*

*"ISBONUSMANAGER": "0",*

*"FROZEN": "0"*

*}*

*],*

*"childs": [*

*{*

*"entity": {*

*"ENTITYNAME": "FHICNYSTRUCTURE",*

*"FROZEN": "0"*

*},*

*"kiosk": {*

*"KIOSKNAME": null,*

*"FROZEN": null*

*},*

*"admins": [],*

*"childs": [*

*{*

*"entity": {*

*"ENTITYNAME": "FHICNY001",*

*"FROZEN": "0"*

*},*

*"kiosk": {*

*"KIOSKNAME": "FHICNY001",*

*"FROZEN": "0"*

*},*

*"admins": [*

*{*

*"ADMINNAME": "FHICNY001",*

*"DEPOSITBALANCE": "0",*

*"ISDEPOSITOWNER": "0",*

*"ISDEPOSITMANAGER": "0",*

*"BONUSBALANCE": "0",*

*"ISBONUSOWNER": "0",*

*"ISBONUSMANAGER": "0",*

*"FROZEN": "0"*

*}*

*],*

*"childs": []*

*}*

*]*

*}*

*]*

*}*

*}*

NB! If the same Kiosk API function is used with the TLE entity key, the whole structure and not only the sub-structure will be shown.

### Create a kiosk inside a sub-entity

The kiosks in our system are used to hold the players inside specific sub-entities. A number of parameters are mandatory to create a kiosk due to the need to have backwards compatibility with the old Kiosk Admin interfaces.

To create a new kiosk using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/kiosk/create/kioskname/FHICNY001/entityname/ FHICNY001/languagecode/countrycode/CN/*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate thus only the sub-entity data is exposed.

*Accept:text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8,Cache-Control: max-age=0,Connection: keep-alive,Keep-Alive:timeout=5, max=100,Accept-Charset:ISO-8859-1,utf-8;q=0.7,\*;q=0.3,Accept-Language:es-ES,es;q=0.8,Pragma: ,X\_ENTITY\_KEY: :083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the kiosk is created:***

{

"result": "Kiosk is successful created"

}

To verify the sub-structure as holding the sub-entity, admin and kiosk you can use again the entity/structure/ function together with the sub-entity key.

### Increase the sub-entity admin deposit balance

For the sub-entity to be able to give depositing balance for the players the admin inside the sub-entity needs to have depots balance. Deposit balance must be transferred by the licensee from the top level admin to the appropriate sub-entity admin.

To make a transfer from one admin to another using the Kiosk API the below call should be made:

***Example API input code:***

*URL:* [*https://kioskpublicapi.mightypanda88.com/admin/increaseBalance/fromadminname/FHICNYTLA/toadminname/FHICNY001/depositbalanceamount/1/*](https://kioskpublicapi.mightypanda88.com/admin/increaseBalance/fromadminname/FHICNYTLA/toadminname/FHICNY001/depositbalanceamount/1/)

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the deposit:***

{ "result": "Deposit balance increased by 1" }

To verify that the sub-entity admin account now has balance the licensee can use again the entity/structure/ function together with the sub-entity key or if the B2B customer of the licensee wants to see the balance of their admin the same function can be used together with the sub-entity key.

### Decrease the sub-entity admin deposit balance

There are occasions where the deposit balance of the sub-entity admins also needs to be decreased. Deposit balance must be transferred by the licensee from the top level admin to the appropriate sub-entity admin.

To make a transfer from one admin to another using the Kiosk API the below call should be made:

***Example API input code:***

*URL: https://kioskpublicapi.mightypanda88.com/admin/decreasebalance/fromadminname/FHICNY001/toadminname/FHICNYTLA/depositbalanceamount/1*

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the withdraw:***

{ "result": "Deposit balance decreased by 1" }

To verify that the sub-entity admin account now has balance the licensee can use again the entity/structure/ function together with the sub-entity key or if the B2B customer of the licensee wants to see the balance of their admin the same function can be used together with the sub-entity key.

## Player data synchronization

Player data synchronization is required as a basis of all other integration parts for our platform. Our integration system requires that a valid user name and password exists in our platform database to authenticate users into the system and the keep the mandatory user information up to date.

### Creating a new Player account

The API offers the licensees B2B customers the possibility to create players inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To create a new player using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/player/create/playername/FHICNY001USER/adminname/FHICNY001/kioskname/FHICNY001/*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player account creation must be done with the sub-entity certificate only and the player must be placed in the sub-entity ADMIN/KIOSK.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the player is created:***

{

"playername": "FHICNY001USER",

"password": "humrriua"

}

### Reading player account information

The API offers the licensees B2B customers the possibility to read player information from inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To read player information using the Kiosk API the below call should be made:

***Example API input code:***

[*https://kioskpublicapi.mightypanda88.com/player/info/playername/*](https://kioskpublicapi.redhorse88.com/player/info/playername/VBETCNYTEST)*FHICNY001USER*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player account information must be fetched with the sub-entity certificate only if the function is used by the sub-entity.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the player information:***

{

"result": {

"CASINONAME": "drunkenmonkey88",

"PLAYERNAME": "FHICNY001USER",

"ADDRESS": "NA",

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

"CITY": "NA",

"COUNTRYCODE": "CN",

"CURRENCY": "CNY",

"EMAIL": "fhicny001user@fhicny001.com",

"FAX": "11111",

"FIRSTNAME": "NA",

"LASTNAME": "NA",

"NOBONUS": 0,

"OCCUPATION": "NA",

"PHONE": "11111",

"SEX": "M",

"STATE": "NA",

"VIPLEVEL": "1",

"WANTMAIL": 1,

"ZIP": "NA",

"ADVERTISER": "default1j",

"BANNERID": "-",

"CLIENTTYPE": "landbasedcasino",

"CREFERER": "null;null;null;null;null;null;null;null",

"LANGUAGECODE": "EN",

"SERIAL": "1JW20CARDUSER4506671",

"SIGNUPDATE": "2014-09-22 02:41:55.000",

"SIGNUPCLIENTVERSION": "-",

"PASSWORD": "35C6392FB134C3998DDDD731BF8B584E54CF8099",

"FROZEN": 0,

"CLIENTSKIN": "drunkenmonkey88",

"COMMENTS": "Created through Public API. ",

"SIGNUPCLIENTPLATFORM": "download",

"ISINTERNALPLAYER": 0,

"SUSPENDED": 0,

"FULLFIRSTNAME": 0,

"FULLFIRSTSURNAME": 0,

"FULLSECONDSURNAME": 0,

"TAXRESIDENCEREGION": 0,

"CODE": "10849246",

"BALANCE": 0,

"MAXBALANCE": 0,

"RESERVEDBALANCE": 0,

"CURRENTBET": 0,

"CURRENTBONUSBET": 0,

"PENDINGBONUSBALANCE": 0,

"BONUSBALANCE": 0,

"COMPPOINTS": 0,

"TOTALCOMPPOINTS": 0,

"KIOSKCODE": "672411",

"KIOSKADMINCODE": "726091",

"CASINOCODE": "1763",

"LOCKEDMINUTES": 0,

"CURRENCYCODE": 0,

"CASINONICKNAME": 0,

"NETWORKNICKNAMES": [

{

"networkData": [

{

"key": 1,

"value": null

},

{

"key": 2,

"value": "casino"

}

],

"networkId": "569"

}

],

"ISINGAME": 0,

"KIOSKNAME": "FHICNY001",

"KIOSKADMINNAME": "FHICNY001"

}

}

NB! Limited response is shown is this document to save space, the actual response is much longer.

### Updating player account information

The API offers the licensees B2B customers the possibility to update player information from inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To update player information using the Kiosk API the below call should be made:

***Example API input code:***

https://kioskpublicapi.mightypanda88.com/player/update/playername/FHICNY001USER/password/test11

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player account information must be updated with the sub-entity certificate only if the function is used by the sub-entity.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the player info is updated:***

{ "result": "ok" }

### Check if player is online or has existing login session

The API offers the licensees B2B customers the possibility to check if players are online from inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To check if player is online or has existing login session using the Kiosk API the below call should be made:

***Example API input code:***

https://kioskpublicapi.mightypanda88.com/player/online/playername/FHICNY001USER/

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player login status must be checked with the sub-entity certificate only if the function is used by the sub-entity.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the player is online:***

{

"result": false

}

### Force logout player or remove existing session

The API offers the licensees B2B customers the possibility to force logout players or remove old login session from inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To force logout player or to remove existing login session using the Kiosk API the below call should be made:

***Example API input code:***

https://kioskpublicapi.mightypanda88.com/player/logout/playername/FHICNY001USER

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player force logout must be done with the sub-entity certificate only if the function is used by the sub-entity.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the log out:***

{

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

}

### Reset the failed login attempts lock for player accounts

The API offers the licensees B2B customers the possibility to reset the login block on players accounts when the players lock themselves out due to entering the wrong password too many times from inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To reset failed login attempts lock can be done using the Kiosk API the below call should be made:

***Example API input code:***

https://kioskpublicapi.mightypanda88.com/player/resetFailedLogin/playername/FHICNYTEST

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player reset failed login attempts must be done with the sub-entity certificate only if the function is used by the sub-entity.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the unlock result:***

{

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

}

### Search player database

The API offers the licensees B2B customers the possibility to search the database of player accounts in our system. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To search can be done using the Kiosk API the below call should be made:

***Example API input code:***

https://kioskpublicapi.mightypanda88.com/player/list/adminname/FHICNY001

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the player list:***

{

"result": [

{

"PLAYERNAME": "FHICNY001USER",

"SIGNUPDATE": "2014-09-22",

"VIPLEVEL": "1",

"KIOSKNAME": "FHICNY001",

"ADMINNAME": "FHICNY001",

"FROZEN": "0",

"LASTLOGINDATE": "2014-09-22",

"BALANCE": "0",

"BONUSBALANCE": "0"

}

],

"total": {

"TotalCount": "1",

"TotalBalance": 0,

"TotalBonusBalance": 0

},

"pagination": {

"currentPage": 1,

"totalPages": 1,

"itemsPerPage": 50

}

}

### Player Deposit and Withdrawal API

Our integration system supports funding player accounts directly from the sub-entity admin accounts over API calls, thus funding the games for players can be automated.

### Depositing chips into a player account

The API offers the licensees B2B customers the possibility to deposit for players inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To deposit for player using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/* player/deposit/playername/FHICNY001USER/amount/1/adminname/FHICNY001/externaltranid/D123456

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player account funding must be done with the sub-entity certificate only and the player must be placed in the sub-entity ADMIN/KIOSK.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the deposit:***

{

"amount": 1,

"execution\_time": "3647.234 ms",

"external\_transaction\_id": "D123456",

"instant\_cash\_type": "local",

"kiosk\_transaction\_id": "75009722",

"kiosk\_transaction\_time": "2014-09-22 15:29:04",

"pt\_internal\_transaction\_id": "612712598",

"result": "ok"

}

### Withdrawing chips from a player account

The API offers the licensees B2B customers the possibility to withdraw from players inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To withdrawal for player using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/* player/withdraw/playername/FHICNY001USER/amount/1/adminname/FHICNY001/externaltranid/W654321

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player account funding must be done with the sub-entity certificate only and the player must be placed in the sub-entity ADMIN/KIOSK.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the withdraw:***

{

"amount": 1,

"execution\_time": "133.8842 ms",

"external\_transaction\_id": "W654321",

"instant\_cash\_type": "local",

"kiosk\_transaction\_id": "75009898",

"kiosk\_transaction\_time": "2014-09-22 15:32:53",

"pt\_internal\_transaction\_id": "612713668",

"result": "ok"

}

### Checking transaction status

The API offers the licensees B2B customers the possibility to check the player transaction status inside their sub-structures. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access to that specific sub-structure only.

To transaction status checking for player using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/*player/checktransaction/externaltransactionid/D123456

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate. Player account funding must be done with the sub-entity certificate only and the player must be placed in the sub-entity ADMIN/KIOSK.

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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8

***The API will return JSON as response showing the transaction status:***

{

"status": "approved",

"type": "deposit",

"amount": "1",

"kiosk\_transaction\_time": "2014-09-22 15:29:04",

"kiosk\_transaction\_id": "75009501",

"pt\_internal\_transaction\_id": "612711244",

"ip": "203.192.162.121",

"username": "FHICNY001USER",

"admin": "FHICNY001",

"external\_transaction\_id": "D123456",

"execution\_time": "123.404 ms"

}

### Reporting API functions

Our API provides simple and easily adjustable functionality to pull reports and data in real time from our system.

### Show a list of available reports

The API offers the licensees B2B customers the possibility to get a list of all available reports. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access of the sub-entity admin.

To see the list of available reports using the Kiosk API the below call should be made:

***Example API input code:***

*URL: https://kioskpublicapi.mightypanda88.com/customreport/getallreports/*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the sub-entity as created:***

[

{

"name": "KioskAdmin",

"description": "Deposit balance per admin"

},

{

"name": "PlayersAccountStats",

"description": "Owner Accounting Statistics by player"

},

{

"name": "KioskTransaction",

"description": "Kiosk Transactions report"

},

{

"name": "PlayerGames",

"description": "Player games report"

},

{

"name": "PlayersOnline",

"description": "Players online"

},

{

"name": "PlayerLogins",

"description": "Player logins"

},

{

"name": "PlayerTransactions",

"description": "Player transactions"

},

{

"name": "PlayerActivity",

"description": "Player activities"

},

{

"name": "KioskMasterAgent",

"description": "Kiosk master agent"

},

{

"name": "AdminActivity",

"description": "Admin activity log"

},

{

"name": "ConcurrentPlayers",

"description": "Concurrent players"

},

{

"name": "AdminPermissions",

"description": "Admin permissions"

},

{

"name": "PlayerCompensations",

"description": "Player compensations"

},

{

"name": "PlayerStats",

"description": "Player statistics"

},

{

"name": "PlayerGamesCount",

"description": "Player last games count"

},

{

"name": "GameStats",

"description": "Game statistics"

},

{

"name": "KioskAdminTransaction",

"description": "Kiosk Admin Transactions"

}

]

NB! The output of the JSON here is limited to a few reports only as the list of available reports long and increasing on a constant basis.

### Show input options for a specific report

The API offers the licensees B2B customers the possibility to see the list of available input options for all available reports. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access of the sub-entity admin.

To see the input options for a specific report using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/customreport/getform/reportname/KioskAdmin*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing the sub-entity as created:***

{

"result": [

{

"name": "adminname",

"label": "Kiosk admin",

"mandatory": "0",

"description": "Name of the kiosk admin",

"minlength": 1,

"maxlength": 15,

"regex": "/[A-Za-z0-9]+/",

"type": "string"

},

{

"name": "entityname",

"label": "Kiosk entity name",

"mandatory": "0",

"description": "Name of the business entity",

"minlength": 1,

"maxlength": 100,

"regex": "/[A-Za-z0-9.\\/,\_-]+/",

"type": "string"

},

{

"name": "sortby",

"label": "Sort by",

"mandatory": "0",

"description": "available values:(code|adminname|entityname|kioskcode|kioskname|currencycode|depositbalance|balance|bonusbalance|datecreated|casino|viplevel|prefix|frozen|parentntity)",

"values": [

"code",

"adminname",

"entityname",

"kioskcode",

"kioskname",

"currencycode",

"depositbalance",

"balance",

"bonusbalance",

"datecreated",

"casino",

"viplevel",

"prefix",

"frozen",

"parentntity"

],

"type": "string"

},

{

"name": "sortorder",

"label": "In descending order",

"mandatory": "0",

"description": "available values:(0|1)",

"values": [

"1",

"0"

],

"type": "integer"

},

{

"name": "page",

"mandatory": "0",

"description": "page of report default is 1",

"regex": "/^[0-9]+$/",

"type": "integer"

},

{

"name": "perPage",

"mandatory": "0",

"description": "result rows per page in report default is 50",

"regex": "/^[0-9]+$/",

"type": "integer"

}

]

}

NB! The output of the JSON here is not shown as the list of available reports long and increasing on a constant basis.

### Show a report and data

The API offers the licensees B2B customers the possibility to get reports for their data only. The licensee must provide the B2B customer with the private SSL certificate sent by us and the licensee must generate an entity key for the sub-structure to restrict API access of the sub-entity admin.

To see a specific report report using the Kiosk API the below call should be made:

***Example API input code:***

*https://kioskpublicapi.mightypanda88.com/customreport/getdata/reportname/KioskAdmin/adminname/FHICNY001*

Do not forget to add the entity key to the HTTP header. In this example the EMPERORCNY sub-entity key is used together with the certificate.

*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: 083e3cc8450065ade3a5944837ec2ff4fd609537a092b6437bb1f63e85cfe9603b9a23f1a35d0822d03bb9c12d66c864afc657c97d542e9e0ca55c98bf7eeed8*

***The API will return JSON as response showing report data:***

*{*

*"result": [*

*{*

*"CODE": "726091",*

*"ADMINNAME": "FHICNY001",*

*"ENTITYNAME": "FHICNY001",*

*"KIOSKCODE": "672411",*

*"KIOSKNAME": "FHICNY001",*

*"CURRENCYCODE": "CNY",*

*"DEPOSITBALANCE": "0",*

*"BALANCE": null,*

*"BONUSBALANCE": "0",*

*"DATECREATED": "2014-09-18",*

*"CASINO": "drunkenmonkey88",*

*"VIPLEVEL": "1",*

*"PREFIX": null,*

*"FROZEN": "0",*

*"PARENTNTITY": "FHICNYSTRUCTURE",*

*"RNUM": "1"*

*}*

*],*

*"pagination": {*

*"currentPage": 1,*

*"totalPages": 1,*

*"itemsPerPage": 50,*

*"totalCount": 1*

*}*

*}*

## Game limits and VIP levels

In order for any of the games to work in either the Download or Flash client each and every game must be configured with the appropriate limits. The set of limits will be configured by us based on your requirements for each VIP level you wish to have for your player accounts and you will decide what player account will get what VIP level and related limits.

Your dedicated Account Executive will assist you with filling in the form for the game limits.

### Player login integration for flash games

Our API provides simple and transparent API for players to log into the gaming system. This API is based on JavaScript and forces the login over HTTPS in between the player’s web browser and our gaming servers. There is no option for a man in the middle or server side login due to security reasons. For any of our offered web based flash games to function the JavaScript based login method must be used.

### JavaScript login method

In order for the JavaScript based login to work the players browser must be directed to include our JavaScript functions.

***Example JavaScript functions location:***

*<script type="text/javascript" src="http://extdev-devhead-pt800045-banner.extdev.eu/integrationjs.php"></script>*

There is an additional library that must be placed in the web servers folder and readable for the browser. The path of the file must be /integration/ folder at the root of the website url and inside the folder the integrationRedirect.html file must be placed with the file name and content unchanged.

*www.yourdomain.com/integration/integrationRedirect.html*

Your account executive will provide you with the URL for the integrationjs.php file and the integrationRedirect.html file so you can copy this to your web server.

Any website that is desired to allow login of players must also be white listed in our system, logins from unknown websites will not work and will return an error.

There are 2 functions in the JavaScript library, login and logout.

***Example JavaScript functions:***

*function login(realMode)*

*function logout(allSessions, realMode)*

The login function should be used to create a login session on our server side.

Your account executive will provide you with example code for the login and logout method that you can copy and/or adjust based on your website logic.

We will use the domain name that you will delegate to our control to create a simple website with the flash login logic implemented to demonstrate the correct implementation.

### Launching flash games

We will provide you the list of game, currency and language codes to launch the flash games on your website. We will use the domain name that you will delegate to our control to create a simple website that will include all of the flash games and icons integrated to our side so that you can copy learn and copy the logic for your websites.

## 

## Document summary

Version 0.2