

Contenido

Panel de administración	1
Información Básica	2
Comando: getBalance - (recibir el saldo en balance para el jugador)	3
Comando: writeBet - (el resultado de la apuesta)	4
Obtención de lista de juegos	6
Reglas de clasificación de la lista de juegos:	7
Ingreso al juego	7
Funcionalidad de pre recarga de imágenes de juegos	9
Apagando la paginación en iframe de juegos IOS 15	14
Cierre de juegos	15
Agregar funcionalidad para guardar la posición de la página después de cierre del juego	16
Agregar funcionalidad para cerrar el juego haciendo click en el botón "atrás" en el navegador	16
Ocultando el enlace mediante el cual se inicia el juego y las imágenes de los juegos	16
(en el lobby) usando RewriteEngine	17
Creación de sala y cambio de ajustes por API:	18
Historia de cada juego	19
Giros Gratis	21

Breve descripción de la integración:

Comando **getBalance** manda POST solicitud al servidor del sitio para verificar el saldo del jugador

Comando **writeBet** manda POST solicitud al servidor del sitio que cambie el saldo del jugador

1. Crear API file en la parte del sitio, conectado a base de datos real para procesar comandos **getBalance** y **writeBet**
2. En los ajustes de la sala en panel de administración de API agregar **Hall key** y **Callback URL**
3. Usando **TEST API** verificar las respuestas en los comandos getBalance y writeBet
4. **TEST API** - Recibir la lista de juegos
5. **TEST API** - Abrir el juego
6. Cerrar el juego, ejemplo de realización demo.gamesapi.biz en computadora y en el celular

Panel de administración

1. Crea una sala (una sala una divisa)

2. Ajustes de la sala

RTP: win/bet

MIN TOTAL BET, MAX TOTAL BET (**los ajustes de apuestas, se aplica para cada juego de tragamonedas excepto el proveedor EGT**)

Hall key: se cambia manualmente

Callback URL: dirección a API para el file en la parte del sitio

3. TEST API comandos **getBalance** y **writeBet**

request = solicitud del servidor

response = respuesta de API de sitio

Sí response está vacío se tiene que revisar API con ayuda de Postman solicitud POST con método form-data

4. Pestaña GAMES / CONFIG

4.1 NOVOMATIC, IGROSOFT, ARISTOCRAT_FLASH

Bet Limit: limitación de posible multiplicación por un giro por apuesta total

Ejemplo: sí Bet Limit 4000 el jugador jamás podrá ganar multiplicación por 5000

**Ejemplo de combinación: 5 symboles MAN en el juego de Novomatic BOOK OF RA jamás*
caye

Risk limit: limitación de posible premio en juego de riesgo

4.2 IGT, EGT, AMATIC, ARISTOCRAT, APOLLO, WAZDAN

Bet Limit AMATIC IGT EGT : limitación de posible multiplicación por 1 giro por apuesta total


Ejemplo: sí Bet Limit 1000 jugador jamás ganará multiplicación Total Bet por 1000

Además, admite entrada múltiple de multiplicación máxima separada por comas

**Ejemplo: 100,100,100,100,50 (sin espacio)*

En este caso los jugadores van a ganar con apuestas bajas de 0 a 80% multiplicado por 100 i el jugador quien va a jugar con apuesta alta de 80% a 100% recibirá multiplicación por 50.

Es posible indicar 4, 5, 10 combinaciones Bet Limit, eso va a influir proporcionalmente en la multiplicación de las apuestas .

5. Visualización de estadísticas de los jugadores de la sala el botón terminales de la sala  o en los ajustes de la sala el botón USERS, en la parte derecha del jugador el botón palanca de mando



muestra las historias del juego en imágenes

Las historias de los juegos se almacenan solo durante 10 días

Información Básica

METHOD	POST
REQUEST	json
RESPONSE	json
TIME ZONE	GTM + 0 (Server time zone)

Comando: **getBalance** - (recibir el saldo en balance para el jugador)

Key	Value	Description
cmd	getBalance	
hall	3	Hall id
key	123456	Hall key
login	letst1001	User login on site side

EJEMPLO REQUEST FROM GAME SERVER

```
{
  "cmd": "getBalance",
  "hall": "3",
  "key": "123456",
  "login": "test"
}
```

EJEMPLO RESPONSE FROM SITE

```
{
  "status": "success",
  "error": "",
  "login": "test",
  "balance": "1000.00", // formato de datos para el balance: Decimal
12,2
  "currency": "LAK"
}
```

EJEMPLO fail response:

```
{
  "status": "fail",
  "error": "ERROR CODE"
}
```

*El formato de datos para el balance: Decimal 12,2 (Ejemplo: 1000.00)

ERROR CODE	Descripción
user_not_found	User not found on the side site

Comando: **writeBet** - (el resultado de la apuesta)

El parámetro "**bet**" solo necesita verificar si el saldo es suficiente antes de procesar el valor de "**win**". Si el saldo es suficiente, reste la "**bet**" del saldo y agregue el valor de "**win**".

Sí el balance no es suficiente para "**bet**":

```
{
  "status": "fail",
  "error": "fail_balance"
}
```

SOLO PARA SportBetting

Sí viene el parámetro "**refund**", entonces revisión del balance no es necesario, en este caso haciendo el recálculo de la apuesta el balance del jugador puede entrar a menos

EJEMPLO REQUEST FROM GAME SERVER:

```
{
  "cmd": "writeBet",
  "login": "letst1001",
  "hall": "3",
  "key": "123456",
  "sessionId": "22333427",
  "bet": "100",
  "win": "100",
  "tradeId": "9999",
  "betInfo": "bet",
  "gameId": "1",
  "matrix": "",
  "date": "2015-02-11 16:35:02",
  "WinLines": ""
}
```

EJEMPLO RESPONSE FROM SITE:

```
{
  "status": "success",
  "error": "",
  "login": "letst1001",
  "balance": "900.00",
  "currency": "LAK",
  "operationId": "3234234"
}
```

operationId minimo 3 - 10 numeros, Data type: int 11

Ejemplo de respuesta errónea:

```
{
  "status": "fail",
  "error": "ERROR CODE"
}
```

ERROR CODE	Descripción
user_not_found	Usuario no encontrado en el sitio
fail_balance	No hay balance suficiente para realizar apuesta

if bet	win	Balance before	Balance after
100	100	1000.00	1000.00
100	0	1000.00	900.00
100	10	1000.00	910.00

if bet	win	Balance before	Balance after
100	100	50.00	fail_balance
100	0	50.00	fail_balance
100	10	50.00	fail_balance

Key	Value	Description	Data types
cmd	writeBet		
hall	3	Hall id	int
key	123456	Hall key	varchar
login	letst1001	User login on site side	varchar
bet	100	Spin bet	decimal 12,2
win	-100	Spin result	decimal 12,2
tradeId	9999	Spin id	varchar
betInfo	bet	Bet Info	varchar

gameId	1	Game id	int
matrix		Screen Matrix	varchar
date		Spin dade	datetime
WinLines		Description	varchar(5000)
sessionId	2222	Session ID	int

OPCIONAL firma adicional para comandos **getBalance** y **writeBet**

La función se activa por una solicitud adicional a nuestro equipo de soporte técnico.

En el momento de activación de los comandos **getBalance** y **writeBet** en vez de parámetro "**key**", de nuestro servidor va a llegar el parámetro "**sign**" en formato sha256, cual ud debería revisar por su parte.

El parámetro "**sign**" no es necesario enviar a nosotros

El parámetro key queda solo por la solicitud de comando **openGame**

Ejemplo de función para revisión de firma, donde ud tiene que entregar los datos de **writeBet** y **Hall Key**:

```
function sign($data,$key){
    ksort($data, SORT_STRING);
    array_push($data, $key);
    $data = implode(':', $data);
    $sign=hash('sha256',$data);
    return $sign;
}
```

\$data - los parámetros, cuales llegan a writeBet

\$key - hall key

Obtención de lista de juegos

Obtención automática de lista de juegos, recomendación de solicitar la lista de juegos despues de la entrada al usuario o guardar y actualizar lo no menos que una vez en una hora

```
url="http://SERVER/API/";
{
    "hall":"hall_id",
```

```

    "key": "hall_key",
    "cmd": "gamesList",
    "cdnUrl": "http://domain.com/resources"
}

```

Llega la lista **"titles"** y la lista de juegos con imágenes para el lobby

"cdnUrl" - parámetro para reescribir la url src de las imágenes con su dominio

```

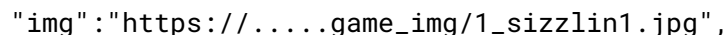
gameTitles: ["novomatic", "ainsworth", "pragmatic", "NetEnt", "microgaming", "scientific_games", "aristocrat", "quickspin", "igt", "scratch", "igrosoft", "amatic", "apex", "merkur", "table_games", "gclub", "habanero", "apollo", "wazdan", "egt", "roulette", "bingo", "keno"]

```

Imágenes **"titles"** se puede [descargar aquí](#)

Reglas de clasificación de la lista de juegos:

```

"id": "1",
"name": "Sizzling Hot Quattro",


```

```

"title": "novomatic", // game title

```

```

"flash": "0", // так как flash не поддерживается необходимо скрывать
"flash": "1" ("0"=false, "1"=true)

```

```

"device": "0" // device "0" для pc, device "1" mobile, device "2" pc +
mobile и необходимо скрывать device "3"

```

Ingreso al juego

```

url="http://SERVER/API/openGame/";

```

```

{
    "cmd": "openGame",
    "hall": "hall_id",
    "domain": "https://domain",
    "exitUrl": "https://domain/close.php",
    "language": "en",
    "continent": "eur",
    "key": "hall_key",
    "login": "player_login",
    "gameId": "1",
    "cdnUrl": "http://domain.com/resources",
}

```

```
"demo": "0"  
}
```

"**cdnUrl**" – parámetro en **openGame** para reemplazar el enlace con su dominio para los juegos que admiten **rewriterule**

"**domain**" – el parámetro es necesario para definir el protocolo

En el momento de envío **http** -> el link del juego volverá de acuerdo al protocolo **http**, para **https** -> envíe el parámetro "**domain**" con **https**

Ejemplo: "**domain**":"**https://your_domain.com/**"

Adicionalmente, el parámetro "**domain**" se usa para el cierre de algunos juegos, el jugador lo redirecciona a **domain/close.php**, los cierres de juegos están descritos más allá en el capítulo Cierre de Juegos..

Usar "**exitUrl**" sí es necesario crear un link individual para el cierre de juegos sin **close.php**

"**continent**": "**eur**" // manda al jugador a proxy servidor indicado eur, usa, asia, australia

"**demo**":"0" // modo de demo para usuarios registrados, "0"=false "1"=true

EJEMPLO DE RESPUESTA DEL SERVIDOR

```
{  
  "status": "success",  
  "error": "",  
  "content": {  
    "game": {  
      "url": game_url, // link de juego  
      "iframe": "1" // soporte iframe  
      "width": "1"  
      "withoutFrame": "1"  
      "exitButton_mobile": "1" // flag for add in game iframe  
      exit button on mobile  
      "exitButton": "1" // flag for add in game iframe exit  
      button on pc  
      "disableReload": "1"  
      "wager": "1"  
      "bonus": "1"  
      "rewriterule": "1" // soporte RewriteEngine  
    },  
    "gameRes": {  
      "sessionId": session_id // id de session del juego  
    }  
  }  
}
```

"**width**": para pc

Sí "**width**":"0" juego se puede iniciar en modo 3:4

Sí "**width**":"1" juego solo admite 16:9

"withoutFrame": para dispositivos móviles

Sí **"0"** el juego se tiene que abrir en iframe en la página del sitio y eso ayuda a realizar el cierre de juegos manteniéndose en el mismo lugar en la lista de juegos

Sí el juego no funciona con iframe viene **"withoutFrame": "1"** es necesario hacer redirect al link, cual transmite el servidor

"disableReload": en el momento de realizar comandos **updateBalance** aparece una posibilidad automática de actualizar el balance adentro del juego en el momento de la recarga estando en el juego, **"disableReload": "1"** apaga la actualización del juego, donde funciona **updateBalance**

"wager": "1" // se permite el uso de **wager**

"bonus": "1" // se permite usar el bono del depósito (cashback)

Funcionalidad de pre recarga de imágenes de juegos

Para pre recarga de imágenes se utiliza [Intersection Observer API](#). En este ejemplo estamos creando objeto-observador con función de llamada y con configuraciones de seguimiento, donde

- rootMargin - sangría alrededor de espacio observable (en predeterminados se usa el espacio de visibilidad del buscador)
- threshold - significado, indicando en qué porcentaje de visibilidad del elemento observable será llamado función de callback

Luego se eligen los elementos necesarios y en el ciclo se asigna el observador.

En el callback función se está revisando, si vemos ahora el elemento

`entries[i].isIntersecting`

y si no había cargado el elemento

```
var observer = new IntersectionObserver(function (entries, observer) {
  for (var i = 0; i < entries.length; i++) {
    var $target = $(entries[i].target);
    if (entries[i].isIntersecting && !$target.hasClass('loaded')) {
      var img = new Image();
      img.target = $target;
      img.target.addClass('loading');
      img.src = $(entries[i].target).attr('data-img');
      img.onload = function () {
        this.target.css('background-image', 'url(' + this.src +
        '))');

        this.target.removeClass('loading');
        this.target.addClass('loaded');
      }
      img.onerror = img.onload;
      observer.unobserve(entries[i].target);
    }
  }
});
```

```

}, {
  rootMargin: '0px',
  threshold: 0.1
});

document.querySelectorAll('.game-element').forEach(game=>{
  observer.observe(game);
})

```

En el momento de mostrar la lista de juegos se tiene que indicar en el link en la imagen en data de atributo - `data-img='${game['img']}'` para prevenir carga anticipada. Después de entrar a campo de visión y carga completa de la imagen en callback del observador data-atributo se cambia a `background-image`.

Añadiendo funcionalidad de pantalla completa en iframe

Todos los buscadores, menos buscadores de iOS, soporta [Fullscreen API](#). Para agregar transición a pantalla completa se tiene que:

1) agregar transición a pantalla completa

```

function toggleFullScreen(isFull) {
  if ((document.fullScreenElement && document.fullScreenElement !==
null) ||
    (!document.mozFullScreen && !document.webkitIsFullScreen)) {
    if (isFull===false) {
      return;
    }
    if (document.documentElement.requestFullScreen) {
      document.documentElement.requestFullScreen();
    } else if (document.documentElement.mozRequestFullScreen) {
      document.documentElement.mozRequestFullScreen();
    } else if (document.documentElement.webkitRequestFullScreen) {
      document.documentElement.webkitRequestFullScreen();
    }
  } else {
    if (isFull===true) {
      return;
    }
    if (document.cancelFullScreen) {
      document.cancelFullScreen();
    } else if (document.mozCancelFullScreen) {
      document.mozCancelFullScreen();
    } else if (document.webkitCancelFullScreen) {
      document.webkitCancelFullScreen();
    }
  }
}

```

```
}
```

2) Agregar llamado de función por click en el elemento necesario (en este ejemplo el click se pone en **\$(document)** y va funcionar en cualquier parte del sitio). Sí será necesario se puede agregar revisión por tipo de equipo - pantalla, teléfono celular y tablet.

```
$(document).click(() => {  
    toggleFullScreen(true);  
});
```

En este caso como iOS no es compatible con Fullscreen API, el paso a pantalla completa en iOS se realiza con agregación de sobreposición encima de todo el sitio con la altura mayor, que la altura del equipo. Entonces hojeando lo se activa un paso sistemático a pantalla completa como en normal chequeo de sitio. En el momento de paso a pantalla completa pasa una revisión - si altura/ancho del sitio coinciden con altura/ancho de equipo, entonces overlay se oculta. La activación se realiza con el mismo modo como se cambia el tamaño.

IMPORTANTE: la superposición personalizada en iOS debe estar habilitada solo en modo horizontal, no mostrarla en modo vertical.

1) en HTML file se tiene que agregar un elemento, en el scroll del cual se hará la transición.

```
<div class="safarihelper-bg">  
    <div id="safarihelper"></div>  
</div>
```

2) agregar los estilos necesarios

```
body.core-scrollcheck {  
    width: 100%;  
    height: 3000px;  
    position: absolute;  
    overflow: auto;  
}  
  
body.core-scrollcheck .core-bonus-message {  
    position: fixed !important;  
}  
  
.notFullscreenSafari {  
    overflow: auto !important;  
}  
  
.fullscreenSafari {  
    overflow: hidden !important;  
    touch-action: none;  
    -ms-touch-action: none;  
    position: relative !important;
```

```
}  
#safarihelper{  
  color: white;  
  font-size: 26px;  
  font-family: 'tekomedium','serif';  
  text-align: center;  
  touch-action: none;  
  pointer-events: none;  
  position: fixed;  
  height: 304px;  
  width: 90px;  
  top: 10px;  
  right: 40px;  
  transform-origin: center top;  
  z-index: 10000;  
  cursor: pointer;  
  background-image: url(../images/fullscreenanim_ios_hand_move.png);  
  background-repeat: no-repeat;  
  background-position: 0 -310px;  
}  
  
.safarihelper-bg {  
  position: fixed;  
  top: 0;  
  left: 0;  
  right: 0;  
  bottom: 0;  
  width: 100%;  
  height: 100vh;  
  background-color: rgba(0,0,0,0.5);  
  z-index: 999999;  
}  
  
.safarihelper-bg, #safarihelper {  
  opacity: 0;  
  visibility: hidden;  
  transition: opacity .5s linear, visibility .5s linear;  
}  
  
.safarihelper-bg.active, #safarihelper.active {  
  opacity: 1;  
  visibility: visible;  
}  
  
.safarihelper-bg:not(.active), #safarihelper:not(.active) {
```

```
    pointer-events: none;
}
```

3) agregar función de transición a pantalla completa para iOS

```
function windowResize() {
    var w=window.innerWidth;
    var h=window.innerHeight;
    var isFullScreen = false;
    var ua = window.navigator.userAgent,
        iOS = !!ua.match(/iPad/i) || !!ua.match(/iPhone/i),
        webkit = !!ua.match(/WebKit/i),
        iOSSafari = iOS && webkit && !ua.match(/CriOS/i);

    if (window.orientation!==undefined && ((w>h &&
window.orientation==0 )|| (w<h && window.orientation!=0))) {
        setTimeout(function() {windowResize();},1000);
        return;
    }

    if ((device.mobile() || device.tablet()) && window.orientation==0)
{
        $('<div>.device-rotate</div>').addClass('active');
    } else {
        $('<div>.device-rotate</div>').removeClass('active');
    }

    if (device.mobile() || device.tablet()) {
$(<div>'body</div>').attr('orientation',window.orientation==0?'portrait':'landscape');
    }

    if (device.ios() && !device.desktop() && !device.tablet() &&
iOSSafari) {
        if (($(<div>'body</div>').hasClass('mobile-game') &&
!$(<div>'body</div>').hasClass('game-without-scrollcheck')) {
            $('<div>'body</div>').addClass('core-scrollcheck');
        } else {
            $('<div>'body</div>').removeClass('core-scrollcheck');
        }

        if (window.screen.width === h && window.orientation!=0) {
            isFullScreen = true;
        } else {
            isFullScreen = false;
        }
    }
}
```

```

        if (!isFullScreen && window.orientation!=0 &&
(!$('body').hasClass('game-without-scrollcheck') &&
$('body').hasClass('mobile-game')))) {
            $('html,
body').removeClass('fullscreenSafari').addClass('notFullscreenSafari');
        } else if (!$('body').hasClass('game-without-scrollcheck') &&
$('body').hasClass('mobile-game')) {
            $('html,
body').addClass('fullscreenSafari').removeClass('notFullscreenSafari');
        }
    }
}
}

```

4) Agregar la llamada de esta función en el momento de carga y de cambio del tamaño del sitio

```

windowResize()
$(window).resize(function() {windowResize()});

```

Este código funciona correctamente solo en Safari en iOS, porque en otros buscadores el link se oculta no completamente

Apagando la paginación en iframe de juegos IOS 15

Por lo que en iOS 15 no funcionan las maneras de apagar la paginación en la página en el momento de prender las ventanillas modales (overflow: hidden; o event.preventDefault(); en el evento touchmove), se tiene que usar el ejemplo [de este artículo](#).

```

var syncWindowHeight = () => {
    if (pageYOffset!=0) {
        localStorage.pageScroll = Math.round(pageYOffset);
    }

    if (device.portrait()) {
        document.querySelector('body').classList.add('disable-scroll');
        document.querySelector('html').classList.add('disable-scroll');
    } else {
        document.querySelector('body').classList.remove('disable-scroll');
        document.querySelector('html').classList.remove('disable-scroll');
    }

    setTimeout(()=>{

```

```

        document.documentElement.style.setProperty(
            "--window-inner-height",
            `${window.innerHeight}px`
        );
    }, 100)
}

```

Estamos haciendo llamado de función al abrir ventanilla modal y colgamos el manejador por cambio del tamaño con llamado.

```

syncWindowHeight();
window.addEventListener("resize", syncWindowHeight);

```

Eliminamos el manejador de cambio del tamaño en el momento de apagar la ventanilla modal.

```

window.removeEventListener("resize", syncWindowHeight);
window.scrollTo(0, localStorage.pageScroll);

```

Además agregamos los estilos para apagar la paginación.

```

.disable-scroll {
    height: calc(var(--window-inner-height) - 1px);
    overflow: hidden;
}

```

También es recomendable agregar un cheque a la vista del dispositivo y llamar

`syncWindowHeight();` solo para iPhones `if (device.iphone()) {` - la manera más fácil - uso de biblioteca [device.js](https://github.com/eligrey/device.js).

Cierre de juegos

Existen 2 tipos de eventos, cuales acontecen en el momento de cierre del juego (para cada juego uno de dos tipos):

1. La página de juego envía **postMessage** una solicitud con cuerpo de consulta **"closeGame", "close"** o **"GAME_MODE:LOBBY"**
2. Todos los demás juegos: la pagina se redirecciona a **domain/close.php**, donde **domain** - dirección del sitio, enviado en **gameOpen**.

Recomendaciones y ejemplos: procesar los dos tipos de eventos de cierre del juego en una función en ejemplos y luego - **closeGame**:

1. Ejemplo de procesar postMessage en javascript:

```

window.onmessage=function(event) {

```

```

        if (event.data=='closeGame' || event.data=='close' ||
event.data=='notifyCloseContainer' || (event.data.indexOf '&&
event.data.indexOf("GAME_MODE:LOBBY")>=0)) {
            closeGame();
        }
    }
}

```

2. Ejemplo [close.php](#) funciona en el momento de abrir el juego con ayuda de [window.open](#) y al abrir juego en iframe

Agregar funcionalidad para guardar la posición de la página después de cierre del juego

Para guardar la posición de la pagina se usa el evento [beforeunload](#). Antes de que el usuario dejará la página nosotros cumplimos con guardar la coordenada del eje Y.

```

window.onunload = window.onbeforeunload = function () {
    localStorage.pageScroll = Math.round(pageYOffset);
};

```

Al abrir la pagina realizamos la paginación hasta el significado guardado.

```

window.scrollTo(0, localStorage.getItem('pageScroll'));

```

Agregar funcionalidad para cerrar el juego haciendo click en el botón "atrás" en el navegador

Para agregar el cierre se tiene que agregar

1) escuchando el evento ["popstate"](#)

```

window.addEventListener('popstate', function (e) {
    if (document.querySelector('body').classList.contains('game-open')) {
        /* функция для закрытия игры */
    }
})

```

2) al abrir el juego cambiamos la grabación en historia con llamado [history.pushState\(\)](#). En este ejemplo se agrega la grabación en historia con id de juego abierto.

```

history.pushState(null, null, window.location.search+'&id='+id);

```

3) La condición obligatoria para un funcionamiento correcto del evento ["popstate"](#) es la eliminación del elemento iframe de la página al cerrar el juego.

```

document.getElementById('gameFrame').innerHTML = '';

```


Ocultando el enlace mediante el cual se inicia el juego y las imágenes de los juegos

(en el lobby) usando RewriteEngine

Al abrir el juego cambiar el link a dominio , con cual se comienza el juego, los juegos se van a cargar mediante su **domain/resources**

"**cdnUrl**": "<http://domain.com/resources>" – parámetro en **openGame** para reemplazar el enlace con su dominio para los juegos que admiten **rewriterule**

Módulo rewrite para el servidor **linux**:

```
a2enmod rewrite headers proxy proxy_http ssl status
```

Example **apache**:

```
Header set Access-Control-Allow-Origin "*"
```

RewriteEngine on

ProxyPass /resources <http://cdn.lvslot.net/resources>

ProxyPassReverse /resources <http://cdn.lvslot.net/resources>

Example **nginx**:

minimum configuration

```
server {  
  
    root /var/www/resources;  
    index index.html index.php;  
    server_name resources.youdomain.com;  
  
    location /resources {  
        rewrite / (.*) /$1 break;  
        proxy_pass http://cdn.lvslot.net/resources;  
    }  
}
```

La excepción para este tipo de comenzar el juego: **rewriterule=0**

Creación de sala y cambio de ajustes por API:

host: http://api_server/API/

Creación de la sala

```
{
  "cmd": "createHall",
  "api_key": "tbs2_agent_api_key",
  "agent": "tbs2_agent_Id",
  "key": "hall_key",
  "host": "http://callback_url/",
  "hall": "",
  "login": "id hall in you system",
  "currency": "USD"
}
```

Recibir la lista de ajustes disponibles

```
{
  "cmd": "changeHallConfig",
  "hall": "hall_ID",
  "key": "hall_key"
}
```

Cambiar los ajustes disponibles

```
{
  "cmd": "changeHallConfig",
  "hall": "hall_ID",
  "key": "hall_key",
  "data": {
    "hall": {
      "ratio": "3x4",
      "language": "AUTO",
      "rtp": "95",
      "egt_jp": "1",
      "bet_min": "0.01",
      "bet_max": "5000.00"
    },
    "ig": {
      "igroDenom": "1.0",
      "denom": "true",
      "minLine": "0",
      "betLimit": "5000",
      "riskLimit": "1000"
    },
    "slgames": {
      "betLimit": "1000",
      "riskLimit": "300"
    }
  }
}
```

```
}
```

*Por comandos adicionales de control de los ajustes comuniquese con nuestro equipo de soporte técnico

Historia de cada juego

EJEMPLO REQUEST FROM SITE:

```
url="http://SERVER/API/";
```

```
{
  "cmd": "gameSessionsLog",
  "hall": "hallid",
  "key": "hall_key",
  "sessionsId": "session id(string)",
  "count": " row count in page(int)",
  "page": "page number(int)"
}
```

Ejemplo de respuesta:

```
{
  "status": "success",
  "error": "",
  "content": {
    "info": {
      "page": 1, // número de página
      "pages": 1 // cantidad de páginas
    },
    "log": [{
      "id": "3091109", // id de apuesta
      "status": "1", // estado
      "BetInfo": "|bet", // info de apuesta
      "gameId": "2", // id de juegos en sistema tbs2
      "gameName": "Tornado", // nombre de juego en tbs2
      "cdn": "https://resources/thumbnails/568",
      // carpeta con imágenes
      "before": "447.38", // balance antes de apuesta
      "bet": "0.20", // la apuesta general
      "win": "0.00", // la ganancia general
      "dateTime": "2020-09-01 12:06:52", // fecha y tiempo de
      // apuesta
      "matrix": {...} // matrix }, ...]
    }
  }
}
```

Las historias de sesiones de juegos se envían solo por **session id**

Es necesario descargar y conectar los files: **matrix.js** y **matrix.css**

Script en **matrix.js** funciona con **jquery**, si no está conectado con el sitio se tiene que conectar lo.

Si es necesario se puede cambiar **matrix.css** de acuerdo al estilo del sitio. **matrix.js** es recomendable de no cambiar.

"распарсить" содержимое ответа и вывести в виде таблицы массив

`response["content"] ["log"]`.

Luego en js para cada matrix (`response["content"] ["log"] [i] ["matrix"]`) llamado

`matrixDraw(elem, matrix)`, donde:

`elem` - elemento

`jquery` - elemento, a donde se va a colocar matrix,

`matrix` - matrix de apuestas en formato json o líneas.

LISTA ERROR CODE

ERROR CODE	Descripción
session_not_found	Sesión no encontrada
session_closed	Sesión cerrada
writeBet:fail_bet	Apuesta más alta que el balance
writeBet:fail_balance	Error del balance de compañero
writeBet:mtwpu	Error el menos maximo de ajustes de la sala
writeBet:actionId_exist	Asi evento ya fue procesado
writeBet:prePayment	Error de prepago, no hay créditos
writeBet:fail_response	Respuesta incorrecta del compañero

Giros Gratis

En `gamesList`, el parámetro "bm" indica si el juego admite la función de giros gratis:

"bm": "0" - el juego no es compatible con esta característica

"bm": "1" - el juego admite la función de giros gratis

EJEMPLO REQUEST FROM SITE:

```
url="http://SERVER/API/openGame/";
{
    "cmd": "gameOpen",
```

```

    "gameId":"6157",
    "localhost":"0",
    "mobile":"0",
    "demo":"0",
    "domain":"https://domain",
    "continent":"eur",
    "login":"login",
    "language":"en",
    "bm":"15|10.00"
}

```

15|10.00 (15 - free games | 10.00 total bet)

Ejemplo de respuesta:

```

{
    "cmd":"writeBet",
    "hall":"hall ID",
    "key":"d251a66a2f3d114437870bd5aa0e9b5c",
    "login":"login",
    "tradeId":"106200091_1_129",
    "bet":"0",
    "action":"spin",
    "win":1,
    "gameId":"6745",
    "sessionId":"22427",
    "bm":"{\"id\":\"22427\", \"_spinCount\":5, \"max\":20, \"totalBet\":1, \"win\":0.75}"
}

```

"_spinCount":5 // número de giro para este momento

Protección free-spins:

Cada nueva sesión abierta se guarda como sesión abierta free-spins. Sesiones abiertas free-spins pueden ser 2 o más. Con cualquier apuesta en una sesión abierta free-spins todas las demás sesiones se cierran (jugar free-spins se puede solo en una sesión). Si sesión de free-spins está cerrada, entonces ninguna apuesta se acepta en la sesión actual con la devolución del error: "status":"fail", "error":"fail_balance"

En el momento de realizar apuesta cada vez, cual se cuenta como comienzo de apuesta, medidor de free-spins se incrementa (o se iguala a cantidad de free-spins si alcanzo el limite).

\$isBet=isset(\$_POST['action']) &&
in_array(\$_POST['action'],array('spin','SpinNormal','','PLAY','Bet'));- es la condición cual dice que esta solicitud se cuenta como el comienzo de la apuesta.

Sí la cantidad de las apuestas es igual a cantidad de free-spins entregados, entonces los free-spins se cierran y ya no se va a poder abrir los al comenzar nuevo juego. Hasta entonces cuando se va a abrir un nuevo juego en tbs2 se transmite la cantidad de apuestas=count-played (la cantidad general de apuestas y cantidad de apuestas jugadas respectivamente). En este caso incluso sí los free-spins están cerrados, las apuestas pueden ser agregadas (sí el jugador no cerró la ventanilla y los últimos giros fueron con premios, entonces tbs2 va a dar una apuesta hasta que último giro no va a ser sin premio).

Un detalle importante, se tiene que diferenciar una sesión de free-spins y el mismo free-spins. Sí la sesión de free-spins está cerrada, entonces ninguna apuesta en esta sesión se acepta. Pero sí el mismo free-spins está cerrado, entonces tiene que haber una posibilidad de agregar allí apuestas.