

Chain`s Code Deluxe (B2B)

Базовая интеграция

Прежде всего необходимо получить идентификатор казино и секретный ключ, которые понадобятся для инициализации сеанса игры и подписи данных. Далее **\$CASINO** и **\$SECRET**. Если вы уже получили эти данные в рамках других интеграций - используйте ранее полученные идентификатор и ключ.

Вся интеграция сводится к двум этапам:

- 1) создать сессию игры методом POST, открыв полученную ссылку в iframe (инициализация)
- 2) реализовать на стороне казино принимающий endpoint. (обработка)

Подпись данных

Все обязательные параметры, передаваемые между игрой и казино, должны быть подписаны сигнатурой. POST запросы от игры сопровождаются заголовком **X-SIGNATURE**, в котором передается цифровая подпись. Ответ от казино на POST запрос, подписывается дополнительным параметром **sign**.

Таким образом, игра ожидает подпись в параметре sign, а казино в заголовке X-SIGNATURE.

Подписываются только обязательные параметры, при помощи алгоритма [HMAC-SHA256](#). Параметры перед подписью нужно объединить в строку через символ #. Параметры должны быть отсортированы в алфавитном порядке по ключам массива

Пример подписи post-параметров

```
// POST sign=hash_hmac('100#casino#EU#EUR#12345#https://casino.url/endpoint#100500', $SECRET)
let sign = [balance, casino, country, currency, game, receiver, uid].join('#')
return hash_hmac(sign, $SECRET).toString()
```

Пример проверки на стороне казино

```
signature(data) {
  let hash = data.join('#');
  return hash_hmac(hash, $SECRET).toString();
}
```

```
check(data, signature) {
  return (this.signature(data) === signature);
}
```

```
let hashData = [
  request.body.action,
  request.body.currency,
  request.body.microtime,
  request.body.session,
  request.body.user
]

if(lib.check(hashData, request.header('X_SIGNATURE'))){
  //..some code
}
```

Инициализация игры (методом POST)

POST параметры:

* - обязательные параметры

Параметр	Тип	Описание
casino *	string	Идентификатор казино \$CASINO, по которому определяются персональные настройки игры и секретный ключ для проверки подписи.
balance *	string	Текущий баланс в переданной валюте
currency *	string	Текущая валюта пользователя EUR USD BTC
game *	string	Идентификатор игры на стороне казино
uid *	integer	Id пользователя казино, для которого будет создана новая игровая сессия
receiver *	string	Ресивер казино передается в зависимости от окружения, в котором работает игра (тестовый сервер, боевой сервер) Example: https://mycasino.com/api/endpoint
sign *	string	Подпись обязательных параметров
country *	string	Код страны пользователя в формате ISO 3166-1 alpha-2 (RU, US)

```
// POST sign=hash_hmac('{balance}#{casino}#{country}#{currency}#{game}#{receiver}#{uid}', $SECRET)
let sign = [POST.balance, POST.casino, POST.country, POST.currency, POST.game, POST.receiver, POST.uid].join('#')
```

Если все обязательные параметры переданы и сигнатура валидна - будет создана игровая сессия и в ответ будет получен url с сессией игры, который необходимо открыть в iframe.

```
// POST response
{
  "status": true,
  "type": "success",
  "url": "https://cc.trueflip.co/?session=CCS-07e4d51345929a3cb6ab78b563d0e695",
  "uuid": "CCS-07e4d51345929a3cb6ab78b563d0e695",
  "sign": "fb70489caa44a7a2288ba5ee5373a32bbbe876314938fa3127846560cc3f3af7"
}
```

Сессию необходимо активировать в течение 20 секунд (открыть url в iframe)

```
// <iframe src=?></iframe>
'https://cc.trueflip.co/?session=CCS-07e4d51345929a3cb6ab78b563d0e695'
```

Игровые события

В процессе игры, на ресивер казино будут отправляться события с детальной информацией о самой ставке.

* - обязательные параметры

Параметр	Тип	Описание
action *	string	событие в игре
currency *	string	валюта в которой сделана ставка
game *	string	идентификатор игры с которым стартовала сессия
microtime *	integer	время операции в миллисекундах
session *	string	игровая сессия
user *	integer	id пользователя казино
uuid	string	уникальный идентификатор ставки
amount	decimal	размер ставки (проверить баланс, списать)
won	decimal	размер выигрыша (начислить)
is_won	boolean	победная ли ставка
details	json	делали по ставке, содержит поля: <ul style="list-style-type: none">● round - раунд на котором завершена игра● round_amount - выигрыш в текущем раунде● coinValue - размер ставки● currency - валюта ставки● coefficients - коэффициент выигрыша в текущем раунде● status - статус ставки

```
// POST + X-SIGNATURE
let sign = [POST.action, POST.currency, POST.game, POST.microtime, POST.session, POST.user].join('#')
// X-SIGNATURE = hash_hmac(sign, $SECRET)
```

Пример (+header: X-SIGNATURE)

```
{
  action: 'pushBet',
  currency: $CURRENCY,           // EUR
  game: $GAMEID,                 // cc123
  microtime: new Date().getTime(), // 154964395134
  session: $SESSION,            // CCS-07e4d51345929a3...
  user: $USERID,                // 100500

  uuid: 'cc-1fbe876314...',
  amount: 0.5,
  won: 1.2,
  // balance += (bet.won - bet.amount)
  is_won: true,
  details: {...{}}
}
```

Ответ от казино

В ответ на всех запросы игра ожидает следующие поля: user, balance, currency, sign. Если для ставки оказалось недостаточно средств, то нужно вернуть помимо баланса код ошибки 110. Этот код выбросит событие modalRefill к родительскому окну, по которому можно открыть окно пополнения баланса в самом казино. Поле error необязательное, в подписи не участвует.

```
{
  user: $USERID,
  balance: 0.05,
  currency: 'BTC',
  sign: '409e91738ea2f137...', // 100500#0.05#BTC
  error: 110, // if there is not enough money to bet
}
```

Тестовое окружение и список игр

Все игры доступны в тестовом окружении, для тестирования интеграций. Можно получить список доступных игр, изображения к ним, и точки входа в эти игры, отправив POST запрос:

- [api-test.trueflip.co/games/list](https://api.trueflip.co/games/list) (development)
- api.trueflip.co/games/list (production)

Для вашей тестовой среды используйте development окружение. Для продакшена соответственно production окружение.

POST api.trueflip.co/games/list

```
"body": [  
  // {other games}...  
  {  
    "name": "Chains Code",  
    "enabled": true,  
    "game_uuid": "da9215e16d95c636f44eb6aab1dcb8d6a82f66ba0b547c8e173cf7a0936cc775",  
    "thumb": {  
      "480x580": "https://api.trueflip.co/assets/CC/cc-480x580.png",  
      "480x580@2x": "https://api.trueflip.co/assets/CC/cc-480x580@2x.png",  
      "500x500": "https://api.trueflip.co/assets/CC/cc-500x500.png",  
      "500x500@2x": "https://api.trueflip.co/assets/CC/cc-500x500@2x.png",  
      "560x420": "https://api.trueflip.co/assets/CC/cc-560x420.png",  
      "560x420@2x": "https://api.trueflip.co/assets/CC/cc-560x420@2x.png"  
    },  
    "background": {},  
    "params": {  
      "has_mobile": true,  
      "has_desktop": true,  
      "category": "slots",  
      "payout": 95.5,  
      "lines": 1  
    },  
    "url": {  
      "href": "https://cc.trueflip.co/?lang=eu",  
      "domain": "cc.trueflip.co",  
      "protocol": "https:",  
      "origin": "https://cc.trueflip.co",  
      "port": "",  
      "qs": "?lang=eu"  
    },  
    "blocked_countries": [  
      "RU",  
      "US"  
    ]  
  },  
  // {other games}...  
]
```

Локализации игр

Во всех играх доступны локализации, которые могут быть подключены при помощи GET параметра **lang**, передаваемого в iframe игры. Если указанная локализация недоступна в игре - будет подключена версия по умолчанию (en).

Есть простой способ получить все игры с выбранной локализацией: достаточно передать параметр lang в POST запросе к методу получения списка игр

POST api.trueflip.co/games/list

Параметр	Тип	Описание
lang	string	Локализация по стандарту ISO 3166-1 . Указывается в нижнем регистре. Например: ru, en, es, kr, tr

В результате для каждой игры будет сформирован готовый URL в параметре **{game}.url.href**, который Вам необходимо использовать для инициализации сессии в игре

```
{
  "name": "Chains Code",
  // ...
  "url": {
    "href": "https://cc.trueflip.co/?lang=eu",
    "domain": "cc.trueflip.co",
    "protocol": "https:",
    "origin": "https://cc.trueflip.co",
    "port": "",
    "qs": "?lang=eu"
  },
}
```