

# 3rd Party API

- Security & Authentication
- Environments
- API Services
  - Receipt Registration - POST /receipts
    - Description
    - Request
    - Response - Success
    - Response - Failure

Účtenkovka's 3rd Party API is public API which can be used by partners to provide their customers with an automatic registration of receipts to their player accounts in Účtenkovka. The API is implemented as simple RESTful HTTP API which allows clients to easily integrate Účtenkovka API to wide variety of IT platforms since RESTful API is based on open standards and implementation of a RESTful API client does not require any language specific features or concepts.

The API for registering receipts consists of one API method described below.

## Security & Authentication

All communication between clients and Účtenkovka API is carried over a secured HTTPS channel which uses bidirectional encryption of sent data and provides protection against man-in-the-middle attacks or similar threats.

Partners identify themselves with a provided SSL client certificate in order to establish trusted relationship and reliably authenticate calling clients. Each partner (i.e. API client) is given a dedicated SSL certificate which they must present when establishing secured TLS connection with Účtenkovka API. Client certificates are issued by Účtenkovka's dedicated Certification Authority. Server SSL certificate presented by Účtenkovka API is issued by a commonly trusted CA.

## Environments

There are two separate, independent environments:

- TESTING environment - for testing and debugging purposes. No lottery draws are conducted.
- PRODUCTION environment - real environment with real receipts and real winnings.

Access to both environments is granted to partners manually. At first, a partner receives an SSL certificate for authentication on TESTING environment. When the partner meets all requirements and shows working integration with API on TESTING environment, they receive a new SSL certificate for accessing the PRODUCTION environment.

## API Services

### Receipt Registration - POST /receipts

URL (TESTING): <https://extapi.uctenkovka-test.cz/receipts>

URL (PRODUCTION): <https://extapi.uctenkovka.cz/receipts>

HTTP Method: POST

### Description

Method allows third parties to register receipts into Účtenkovka. The newly registered receipt is assigned to a player account which is identified by either player's e-mail address or by player's phone number (only if e-mail address is not specified). If no existing player account can be found by provided e-mail address or phone number, new player account with basic registration will be created. That is why it is important to verify validity of provided e-mail/phone number with the customer as it may result in registration of receipts to a wrong player account.

### Request

### Parameters

Name	Type	Mandatory	Description	Example(s)
email	String	Y (if phone not specified)	Player's e-mail address, used to identify the player in Účtenkovka. If specified, phone parameter is ignored.	jan.novak@example.org
phone	String	Y (if email not specified)	Player's phone number, used to identify the player in Účtenkovka only if email parameter is not specified.  Only phone numbers without international prefix or with the Czech international prefix (+420) are accepted, whitespaces are ignored.	+420 777 123 456  777 123 456  777123456
basicConsent	Boolean	Y	Player's explicit consent to the processing of their personal data. If not true, the receipt registration is rejected.	true  false
fik	String	Y (if bkp not specified)	FIK code. Only the first 3 parts (18 characters) of FIK are required, or it must be specified in full form.  This regular expression applies for validation of specified values:  $^{\wedge}[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{4\}-4[0-9a-fA-F]\{3\}-[89abAB][0-9a-fA-F]\{3\}-[0-9a-fA-F]\{12\}-[0-9a-fA-F]\{2\})? \$$	B3A09B52-7C87-4014  b3a09b52-7c87-4014  b3a09b52-7c87-4014-a496-4c7a53cf9125-03
bkp	String	Y (if fik not specified)	BKP code. Only the first 2 parts (17 characters) of BKP are required, or it must be specified in full form.  This regular expression applies for validation of specified values:  $^{\wedge}[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{8\}-[0-9a-fA-F]\{8\})? \$$	01234567-89abcdef  01234567-89ABCDEF  01234567-89abcdef-01234567-89abcdef-01234567
date	Date	Y	Date of sale in ISO-8601 format.	2018-03-17
time	Time	Y	Time of sale in ISO-8601 format, seconds are optional.	16:41  08:13:55
amount	Number	Y	Total amount (in hellers).	100 (for 1 CZK)  4570 (for 45.70 CZK)
simpleMode	Boolean	N (defaults to false)	Sale regime, false for regular regime (default), true for simplified regime.	true  false

## Examples

#### 10 CZK, regular sale regime, FIK in short form, player identified by e-mail

```
{
  "email": "jan.novak@example.org",
  "basicConsent": true,
  "fik": "b039bd2f-7c87-4014",
  "date": "2017-05-10",
  "time": "09:23:02",
  "amount": 1000
}
```

#### 421.50 CZK, regular sale regime, FIK in full form, player identified by phone number

```
{
  "phone": "+420756809809",
  "basicConsent": true,
  "fik": "b039bd2f-7c87-4014-a496-4c7a53cf9125-03",
  "date": "2017-05-10",
  "time": "09:23:02",
  "amount": 42150,
  "simpleMode": false
}
```

#### 100 CZK, simplified sale regime, BKP in short form, player identified by e-mail

```
{
  "email": "someemail@test.com",
  "basicConsent": true,
  "bkp": "7f67ea35-6d9f77fb",
  "date": "2017-05-10",
  "time": "09:23:02",
  "amount": 10000,
  "simpleMode": true
}
```

## Response - Success

HTTP 201 - CREATED

If a request for receipt registration passes all validation checks and is successfully processed, response with HTTP status 201 is sent back to the client and it contains JSON data with the following parameters:

### Parameters

Name	Type	Mandatory	Description	Allowed values
------	------	-----------	-------------	----------------

receiptStatus	String	Y	Status of newly registered receipt.	<p>NEW - Receipt has been accepted but not yet successfully verified in ADIS. Verification will be repeated.</p> <p>REJECTED - Receipt could not be verified in ADIS (probably due to incorrect receipt data) and will not be registered into any lottery draw.</p> <p>VERIFIED - Receipt has been successfully verified in ADIS and will be registered into a lottery draw if it passes additional checks of lottery rules.</p>
playerAssignmentStatus	String	Y	Status of receipt assignment to player account.	<p>BASIC_PLAYER_CREATED - The receipt has been assigned to a newly created basic player account, i.e. the given e-mail or phone does not belong to any existing players and a new player account connected with the specified e-mail or phone had to be created.</p> <p>ADDED_TO_BASIC_PLAYER - The receipt has been assigned to an existing basic player account registered to the specified e-mail or phone.</p> <p>ADDED_TO_FULL_PLAYER - The receipt has been assigned to an existing fully registered player account with the specified e-mail or phone.</p>

## Examples

### Receipt accepted, not verified in ADIS, assigned to a newly created basic player account

```
{
  "receiptStatus": "NEW",
  "playerAssignmentStatus": "BASIC_PLAYER_CREATED"
}
```

### Receipt successfully verified in ADIS and assigned to an existing fully registered player account

```
{
  "receiptStatus": "VERIFIED",
  "playerAssignmentStatus": "ADDED_TO_FULL_PLAYER"
}
```

## Response - Failure

### HTTP 400 - BAD REQUEST

If a request for receipt registration does not pass all validation checks, it is rejected and a response with HTTP status 400 is sent back to the client containing description of validation errors in JSON format. Each validation error is specified as a JSON object with 3 attributes:

- field - designation of attribute which is not valid

- code - error code describing nature of validation breach
- message - additional textual description of validation breach

## Examples

### Neither e-mail nor phone is specified, FIK in invalid format

```
[
  {
    "field": "email",
    "code": "string.blank",
    "message": "Either e-mail or phone must be specified"
  },
  {
    "field": "fik",
    "code": "object.invalid",
    "message": "Invalid FIK format"
  }
]
```