

# GTSALIVE

## ALIVE Platform

### NCDB Orders API Integration Manual

# Table of Contents

Table of Contents .....	2
Revision History .....	4
Introduction .....	5
Workflows.....	5
New Order .....	5
Get Details About Issued Cards.....	5
Cardholder Data Update.....	6
API Behavior.....	6
Parameter and Attribute Validations .....	6
Listings.....	6
Process Model.....	6
Authentication .....	6
Authorization .....	6
External Identifiers.....	7
New Order Creation .....	7
Data Model .....	8
General Objects .....	8
[OBJECT] address .....	8
[OBJECT] registeredAddress.....	9
[OBJECT] error.....	9
[OBJECT] errorResponse .....	10
[OBJECT] collection .....	10
[OBJECT] roles.....	10
[OBJECT] administrativeUnits .....	11
Data Objects .....	13
[MODEL] cardOrder .....	13
[MODEL] cardOrderEntry.....	13
[MODEL] cardholder .....	14
[MODEL] card.....	14
[MODEL] cardIssuer .....	15
Shared Enums .....	17
[ENUM] Card Type .....	17
[ENUM] Card Order Request Type.....	17
[ENUM] Card Order Status.....	17
[ENUM] Card Status .....	18
Resources .....	19
Cardholders .....	19
GET /cardholders .....	19
POST /cardholders .....	19

GET /cardholders/{id}.....	19
PUT /cardholders/{id} .....	19
GET /cardholders/{id}/photo .....	20
PUT /cardholders/{id}/photo .....	20
<b>Card Orders .....</b>	<b>21</b>
GET /cardOrders .....	21
POST /cardOrders .....	21
GET /cardOrders/{id}.....	21
PUT /cardOrders/{id} .....	21
DELETE /cardOrders/{id}.....	22
GET /cardOrders/{id}/entries .....	22
POST /cardOrders/{id}/entries .....	22
GET /cardOrders/{orderId}/entries/{entryId}.....	22
DELETE /cardOrders/{orderId}/entries/{entryId} .....	22
<b>Cards .....</b>	<b>23</b>
GET /cards.....	23
GET /cards/{id}.....	23
<b>Card Issuers .....</b>	<b>24</b>
GET /cardIssuers .....	24
GET /cardIssuers/{id} .....	24
<b>Administrative Units.....</b>	<b>25</b>
GET /administrativeUnits .....	25
<b>Test Environment.....</b>	<b>26</b>
<b>Production Environment .....</b>	<b>26</b>
<b>Contacts.....</b>	<b>26</b>

# Revision History

Date	Version	Author	Description of Changes
20. 09. 2017	1.0	Radek Klein	Initial release.
29. 04. 2019	1.1	Lukas Kabicek	Added new endpoints to get list and detail of the card issuers and list of local country administrative units. Minor graphical and formatting adjustments.
24. 10. 2019	1.2	Lukas Kabicek	Test system Base URL was changed.

# Introduction

This document describes REST API of the information system NCDB which could be used for the partner system integration allowing automated cardholder and card order creation, cards data retrieving and cardholder data updating.

## Workflows

In the following text the „client“ means ISIC exclusive representative’s business partner who uses the API.

### New Order

- The client sends query to the endpoint **POST /cardOrders** containing parameters of the order like name and note. If the query is accepted, the response will contain information about the created order and its ID which will be used by the client for creating card order entries. The success HTTP response code is **201 Created**.
- If the cardholders are not already in the NCDB the client is requested to create them:
  - Using the endpoint **GET /cardholders** the client will ask NCDB for the cardholder using his name, surname and date of birth or external ID (if the client previously saved it with the cardholder).
  - If the cardholder is not in the NCDB, the client will use the endpoint **POST /cardholders** to create him/her.
- The client will create card order entries using the endpoint **POST /cardOrders/{id}/entries**. The client has the order ID stored from the time of its creation. Each card order entry should be sent separately as a standalone API query. The success HTTP response code is **201 Created** containing the information about created card order entry. In case of error the HTTP response code is **400 Bad Request** and it includes the information about the error (error code and localized error message).
  - In case of error or any other reason the client can delete the order using the endpoint **DELETE /cardOrders/{id}** (in case of error it’s a recommended process).
- Once all card order entries are imported to the card order, the client is requested to change the order state from DRAFT to CREATED using the endpoint **PUT /cardOrders/{id}**.

### Get Details About Issued Cards

The client gets the information about issued cards using the endpoint **GET /cards**. To get only newly created or updated cards from the previous API call the client should remember the last API call date and use the *lastModifiedFrom* parameter.

## Cardholder Data Update

- The client gets the information about the cardholder using the endpoint **GET /cardholders** with appropriate filter parameters. This provides client cardholder identification including cardholder's ID for future API calls.
- The client compares received data with the ones in his system and in case of an update necessity the client can use the endpoint **PUT /cardholders/{id}** to update cardholder in NCDB.
- The client should know when the last change of cardholder data was done in his system and send only changed cardholder data to the NCDB (not compare everyone's data with NCDB) to prevent high data transfers.

## API Behavior

### Parameter and Attribute Validations

- The API validates query parameters and attributes of modified objects.
- Any errors are in the body of HTTP 400 response (see object model „Error“).
- The error message language resolution is based on the Accept-Language header.
- Clients are requested to ignore all unknown attributes to preserve compatibility in the future.
- The order of attributes in the JSON response is not firm and can be changed in the future, clients are requested to not rely on it.

### Listings

- The API supports „offset“ and „limit“ parameters in data listing requests.
- The “limit” parameter is mandatory, its maximal value is not restricted.
- The API uses logical operator AND between query parameters.

## Process Model

### Authentication

- The authentication uses HTTP BASIC standard.
- The API is not session based. The authentication data needs to be sent with each request.
- The credentials authentication is processed using the same components as for the GUI. This means every client user with access to NCDB's GUI can successfully authenticate via API.

### Authorization

- The authorization is the same with the NCDB GUI. It means that every user of the client with access to NCDB's GUI has the same privileges also in the API.

## External Identifiers

- The Cardholder object contains special attribute „externalId“.
- The attribute can be set and read only via API. It's not shown in the NCDB GUI.

## New Order Creation

- A special DRAFT attribute is available for the card orders.
- Card orders in the DRAFT state are not visible in the NCDB's GUI. They can be found only via API.
- Old card orders in the DRAFT state are automatically deleted. The job DOES NOT delete Cardholder objects.

# Data Model

- The communication format is JSON.
- Photos are expected as direct data (octet-stream) with appropriate Content-Type header. API doesn't support base64 encoded photos.
- Some API responses include nested referenced objects (e.g. the Card includes information about the Cardholder) but not all the data attributes of the referenced objects are included. In case the client needs the whole object it's necessary to send separate query using the appropriate endpoint and object reference ID.

## Sample

```
{
  "id": 1,
  "cardNumber": "S000123456789H",
  "cardholder": {
    "id": 2, /* use this to call GET /cardholders/2 */
    "firstName": "John",
    "lastName": "Doe"
    /* other attributes might not be fetched */
  }
  /* ... */
}
```

## General Objects

### [OBJECT] address

Shared object for address data.

Attribute Name	Type	Description
address1	string(255)	First address line
address2	string(255)	Second address line
city	string(255)	City
district	enum	State/Province/County
postalCode	string(10)	Postal Code/ZIP

## Sample

```
{
  "address1": "High Street 10B",
  "address2": "Level 6",
  "city": "Prague 3",
  "district": "Svitavy",
  "postalCode": "13000"
}
```



## [OBJECT] registeredAddress

Shared object for card issuer registered address data.

Attribute Name	Type	Description
addressLine1	string(255)	First address line
addressLine2	string(255)	Second address line
city	string(255)	City
district	enum	State/Province/County
postalCode	string(10)	Postal Code/ZIP
country	string(3)	3 letter country code (ISO 3166-1 alpha-3)

### Sample

```
{
  "addressLine1": "High Street 10B",
  "addressLine2": "Level 6",
  "city": "Prague 3",
  "district": "Svitavy",
  "postalCode": "13000",
  "country": "CZE"
}
```

## [OBJECT] error

The object representing validation (or other) error.

Attribute Name	Type	Description
path	string	Property path for the affected value
code	string	Technical code of the error
params	object	Key-value parameters of the error
message	string	Error message to show to the user

### Sample

```
{
  "path": "firstName",
  "code": "tooLong",
  "params": {
    "max": 255
  },
  "message": "The value can't be longer than 255 characters."
}
```

## [OBJECT] errorResponse

The response in case of the wrong query.

Attribute Name	Type	Description
message	string	General reason of the refusal
errors	error[]	Individual errors detected by the NCDB

### Sample

```
{
  "message": "The request is not valid.",
  "errors": [
    { ... },
    ...
  ]
}
```

## [OBJECT] collection

The object representing list of the found objects.

Attribute Name	Type	Description
items	object[]	Found objects list

### Sample

```
{
  "items": [
    { ... },
    { ... },
    ...
  ]
}
```

## [OBJECT] roles

Array for card issuer roles which is included in the response of the card issuer related endpoints.

Attribute Name	Type	Description
id	number	Internal card issuer role identifier
name	string	Card issuer role name (internal constant)

## Sample

```
1 {
2   "roles": [
3     {
4       "id": 5,
5       "name": "university"
6     },
7     {
8       "id": 4,
9       "name": "highSchool"
10    }
11  ]
12 }
```

## [OBJECT] administrativeUnits

The object representing local country administrative units (regions and districts) to be used while creating or updating cardholders. Array with districts is nested under superior region.

Attribute Name	Type	Description
id	number	Internal region or district identifier
name	string	Region or District (State/Province/County) name

## Sample

```
1 {
2   "items": [
3     {
4       "id": 1,
5       "name": "Hlavní město Praha",
6       "districts": [
7         {
8           "id": 222,
9           "name": "Hlavní město Praha"
10        }
11      ]
12    },
13    { ... },
14    {
15      "id": 14,
16      "name": "Zlínský",
17      "districts": [
18        {
19          "id": 77,
20          "name": "Kroměříž"
21        },
22        {
23          "id": 78,
24          "name": "Uherské Hradiště"
25        }
26      ]
27    }
28  ]
29 }
```

```
25     },
26     {
27         "id": 79,
28         "name": "Vsetín"
29     },
30     {
31         "id": 80,
32         "name": "Zlín"
33     }
34 ]
35 }
36 ]
37 }
```

# Data Objects

## [MODEL] cardOrder

The object representing card order.

Attribute Name	Type	Description
id	number	Internal card order identifier
status	enum	Card order status by the internal list
name	string	Card order name

### Sample

```
{
  "id": 1,
  "state": "DRAFT",
  "name": "Peter's order"
}
```

## [MODEL] cardOrderEntry

The object representing card order entry. The API doesn't disclose insurance order attributes.

Attribute Name	Type	Required	Description
id	number	Yes	Internal card order entry identifier
cardholderId	number	Yes	Reference to the cardholder
requestType	enum	Yes	Card order request type
cardType	enum	Yes	Card type
validFrom	date	Yes	Start of the card validity
validTo	date	Yes	End of the card validity

### Sample

```
{
  "id": 1,
  "cardholderId": 2,
  "cardType": "ISIC_SCHOLAR",
  "requestType": "NEW_CARD",
  "validFrom": "2017-09-01",
  "validTo": "2018-12-31"
}
```

## [MODEL] cardholder

The object representing cardholder.

Attribute Name	Type	Required	Description
id	number	Yes	Internal cardholder identifier
externalId	string(255)	No	External cardholder identifier
personalId	string(255)	No	Cardholder's personal ID
firstName	string(255)	Yes	Cardholder's name
lastName	string(255)	Yes	Cardholder's surname
prefixTitle	string(255)	No	Cardholder's prefix title
suffixTitle	string(255)	No	Cardholder's suffix title
dateOfBirth	date	Yes	Cardholder's date of birth
className	string(32)	No	Cardholder's school class name
student	boolean	No	Cardholder has student status
studyMode	enum	No	Cardholder's study mode ('FULL_TIME', 'PART_TIME').
email	string(255)	No	Cardholder's e-mail address
cellPhone	string(32)	No	Cardholder's phone number
address	Address	No	Cardholder's address

## Sample

```
{
  "id": 1,
  "externalId": "123456",
  "firstName": "John",
  "lastName": "Doe",
  "dateOfBirth": "1999-01-01",
  "address": {
    ...
  }
}
```

## [MODEL] card

The object representing issued card.

Attribute Name	Type	Description
id	number	Internal card identifier

cardType	enum	Card type
cardNumber	string(255)	Card serial number (including type and check letter).
mifareChipNumber	String(255)	Mifare RFID chip number
emMarineChipNumber	string(255)	EM-Marine RFID chip number
status	enum	Card status
validFrom	date	Card validity start
validTo	date	Card validity end
cardholder	Cardholder	Cardholder

## Sample

```
{
  "id": 1,
  "cardType": "ISIC_SCHOLAR",
  "cardNumber": "S420123456789A",
  "status": "VALID",
  "validFrom": "2017-01-01",
  "validTo": "2018-01-31",
  "cardholder": {
    "id": 2,
    ...
  }
}
```

## [MODEL] cardIssuer

The object representing card issuer (e.g. school, appointed office).

Attribute Name	Type	Description
cardIssuerId	number	Internal identifier of the card issuer
name	string	Name of the card issuer to be displayed to cardholders/applicants
nameOnCardLine1	string	Card issuer name which will be printed on the plastic card / displayed on the virtual card – line 1.
nameOnCardLine2	string	Card issuer name which will be printed on the plastic card / displayed on the virtual card – line 2.
agent	boolean	If true, the card issuer can issue cards for applicants from any eligible school – usually used for appointed offices.
autocompleteEnabled	boolean	If true, the card issuer can be listed in the card issuers autocomplete in the Online Ordering and Card Issuing ALIVE Platform modules and in the third-party client systems.
issuesOwnCards	boolean	If true, the card issuer (usually school) issues own co-branded cards for its students/teachers. Third-party clients should inform applicants about that to prevent complaints.

contactPerson	string	Name of the person in the school who is responsible for the cooperation with ISIC Exclusive Representative.
personalPickupAddress	string	A place in the school where cardholders may collect their cards.
personalPickupOpeningHours	string	Opening hours of the place in the school where cardholders may collect their cards.
roles	[OBJECT] roles	Card Issuer Roles
registeredAddress	[OBJECT] registeredAddress	Card Issuer Registered Address
companyRegistrationNumber	string	Government given company registration number (CRN, WIN, IČO)

## Sample

```

1 {
2   "cardIssuerId": 11015,
3   "name": "Vysoká škola logistiky, Přeřov",
4   "nameOnCardLine1": "Vysoká škola logistiky o.p.s.",
5   "nameOnCardLine2": "Palackého 1381/25, Přeřov",
6   "agent": false,
7   "autocompleteEnabled": false,
8   "issuesOwnCards": false,
9   "contactPerson": "Hana Frýdlová",
10  "personalPickupAddress": "Studijní oddělení, budova B, 2. patro",
11  "personalPickupOpeningHours": "Po-Pá: 8-17 hod.",
12  "roles": [
13    {
14      "id": 5,
15      "name": "university"
16    }
17  ],
18  "registeredAddress": {
19    "addressLine1": "Palackého 1381/25",
20    "addressLine2": "Budova B",
21    "city": "Přeřov",
22    "district": "Přeřov",
23    "postalCode": "750 02",
24    "country": "CZE"
25  },
26  "companyRegistrationNumber": "123456789"
27 }

```



## Shared Enums

### [ENUM] Card Type

Value	Description
ISIC	ISIC card (for full-time students).
ISIC_SCHOLAR	ISIC SCHOLAR card (in the Czech Republic it's also known as ISIC Školák).
ITIC	ITIC card (for teachers).
IYTC	IYTC card (for youth people).
ALIVE	ALIVE card (for members of sport associations, issued only in the Czech Republic).
ALIVE_STUDENT	ALIVE Student card (for part-time students, issued only in the Czech Republic).
ALIVE_STAFF	ALIVE Staff card (for non-teaching staff of schools, issued only in the Czech Republic).
ALIVE_ALUMNI	ALIVE Alumni card (for university graduates, issued only in the Czech Republic).

### [ENUM] Card Order Request Type

Value	Description
NEW_VIRTUAL	New virtual card.
NEW_CARD	New virtual and plastic card.

Card orders support more request types but only the above-listed are supported by this API at the moment. All unknown response values have to be ignored by the client (e.g. *DUPLICATE\_OTHER*, *INSURANCE*).

### [ENUM] Card Order Status

Value	Description
DRAFT	Card order created using this API which can be edited only via this API. It's not shown in NCDB GUI.
CREATED	Draft card order which can be edited by card issuer' user in NCDB GUI. Initial status for card orders created via NCDB GUI.
SENT	Completed card order sent by card issuer' user to be processed by ISIC Exclusive Representative.
READY	Card order sent to production (plastic cards printing) by ISIC Exclusive Representative.
PROCESSED	All card order entries were processed, plastic cards printed. Card order is waiting for an invoice.
ACCOUNTED	An invoice was issued. The card order is waiting for an expedition.
ON_THE_WAY	The card order was shipped to the card issuer and waiting for delivery confirmation.

DELIVERED	Card order delivery was confirmed by the card issuer' user. The last status of each card order.
-----------	---

## [ENUM] Card Status

Value	Description
VALID	Valid card in meaning "properly issued" – <b>doesn't inform about card validity period!</b>
CANCELLED	Cancelled (voided) card (e.g. returned, damaged, lost).

# Resources

## Cardholders

### GET /cardholders

Cardholder search. It's possible to use parameters below. API returns list of cardholders.

#### Query Parameters

Name	Type	Description
externalId	string	External cardholder identifier (case-insensitive).
firstName	string	Cardholder name (case-insensitive).
lastName	string	Cardholder surname (case-insensitive).
dateOfBirth	date	Cardholder date of birth.

#### Response

Collection<cardholder>

### POST /cardholders

Create cardholder.

#### Query

cardholder

#### Response

cardholder

### GET /cardholders/{id}

Get cardholder with given identifier.

#### Response

cardholder

### PUT /cardholders/{id}

Update cardholder with given identifier.

### **Query**

cardholder

### **Response**

cardholder

### **GET /cardholders/{id}/photo**

Get photo of the cardholder with given identifier.

### **Response**

Photo

### **PUT /cardholders/{id}/photo**

Add or update photo of the cardholder with given identifier.

### **Query**

Photo

### **Response**

Photo

# Card Orders

## GET /cardOrders

Get list of the orders, search option

### Query Parameters

Name	Type	Description
name	string	Card order name (case-insensitive).

### Response

Collection<cardOrder>

## POST /cardOrders

Create card order.

### Query

cardOrder

### Response

cardOrder

## GET /cardOrders/{id}

Get card order with given identifier.

### Response

cardOrder

## PUT /cardOrders/{id}

Update card order with given identifier.

### Query

cardOrder

### Response

cardOrder

## **DELETE /cardOrders/{id}**

Delete order in DRAFT or CREATED status.

### **Response**

-

## **GET /cardOrders/{id}/entries**

Get card order entries list from the order with given identifier.

### **Response**

Collection<cardOrderEntry>

## **POST /cardOrders/{id}/entries**

Create card order entry in the order with given identifier.

### **Query**

cardOrderEntry

### **Response**

cardOrderEntry

## **GET /cardOrders/{orderId}/entries/{entryId}**

Get card order entry with given identifier from the order with given identifier.

### **Response**

cardOrderEntry

## **DELETE /cardOrders/{orderId}/entries/{entryId}**

Delete card order entry with the given identifier from the order with given identifier.

### **Response**

-

# Cards

## GET /cards

Get issued cards list, search option.

### Query Parameters

Name	Type	Description
lastModifiedFrom	datetime	Search for cards issued or modified after the specified date and time.

### Response

Collection<card>

## GET /cards/{id}

Get issued card with given identifier.

### Response

card

# Card Issuers

## GET /cardIssuers

Get card issuers list, search option.

### Query Parameters

Name	Type	Required	Description
search	string	No	String for fulltext search in the card issuer's name and registered address (case-insensitive).
autocompleteEnabled	boolean	No	Issuing of the cards through the Online Ordering and Card Issuing ALIVE Platform modules as well as through the third-party client systems is enabled. By default, use value <b>true</b> .
limit	number	Yes	Specifies how many items will be returned for one query.
offset	number	No	Zero based index of the items to be returned.

### Response

Collection<cardIssuer> (simplified)

```
1 {
2   "items": [
3     {
4       "cardIssuerId": 10017,
5       "name": "Gymnázium Nad Štolou",
6       "registeredAddress": {
7         "addressLine1": "Nad štolou 1510/1",
8         "addressLine2": "Budova A",
9         "city": "Praha 7",
10        "district": "Hlavní město Praha",
11        "postalCode": "17000",
12        "country": "CZE"
13      }
14    },
15    { ... }
16  ]
17 }
```

## GET /cardIssuers/{id}

Get card issuer with given identifier.

### Response

cardIssuer



## Administrative Units

### GET /administrativeUnits

Get local country administrative units (regions and districts also known as state/province/county).

#### Response

Collection<administrativeUnits>

# Test Environment

The following details are necessary for web service integration testing. Please ask your ISIC Exclusive representative for authentication details and card issuer setup based on your contract specification.

**Swagger Documentation:** <https://ncdb.test.aliveplatform.com/rest/2.0/swagger-ui.html>

**Base URL:** <https://ncdb.test.aliveplatform.com/rest/2.0>

**Username:** *ask your ISIC ER*

**Password:** *ask your ISIC ER*

**Card Issuer ID (optional):** *ask your ISIC ER*

# Production Environment

Access details to the production environment will be provided after successful implementation against the test environment.

**Swagger Documentation:** <https://ncdb.aliveplatform.com/rest/2.0/swagger-ui.html>

**Base URL:** <https://ncdb.aliveplatform.com/rest/2.0>

**Username:**

**Password:**

**Card Issuer ID (optional):**

# Contacts

For all support queries please contact GTS ALIVE IT department at [support@gtsalive.com](mailto:support@gtsalive.com).