Article

Overview

This API provides methods for article service

Document created : 2019-10-02 07-32

Version information

Version : v3.0

URI scheme

Host : hostname is the same as your POS Cloud.
BasePath : /api/articles/v3.0

Paths

Create a single article group.

```
POST /api/articles/v3.0/article_groups
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Body	**articleGroup** *required*	Data for article group.	ArticleGroup

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	ArticleGroupSingle
201	Entity was created successfully.	ArticleGroupSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Get article groups data.

```
GET /api/articles/v3.0/article_groups
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Responses

HTTP Code	Description	Schema
200	Current state of article groups.	ArticleGroupList
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

GET /api/articles/v3.0/article_groups/name/{articleGroupName}

Parameters

Type	Name	Schema
Header	**X-Business-Units** *required*	string
Path	**articleGroupName** *required*	string

Responses

HTTP Code	Description	Schema
200	successful operation	ArticleGroupSingle

Produces

- `application/json`

Get article group data.

```
GET /api/articles/v3.0/article_groups/{articleGroupId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleGroupId** *required*	ID of article group.	integer (int64)

Responses

HTTP Code	Description	Schema
200	Current state of article group.	ArticleGroupSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Put new data of a single article group.

```
PUT /api/articles/v3.0/article_groups/{articleGroupId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string

Type	Name	Description	Schema
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleGroupId** *required*	ID of article group.	integer (int64)
Body	**articleGroupEXT** *required*	Data for article group.	ArticleGroup

Responses

HTTP Code	Description	Schema
200	Current state of article group (updated).	ArticleGroupSingle
201	Current state of article group (created).	ArticleGroupSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Delete single article group of business unit id.

```
DELETE /api/articles/v3.0/article_groups/{articleGroupId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleGroupId** *required*	ID of article group.	integer (int64)

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	Data conflict, provided state can't be accepted.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Patch new data of a single article group.

```
PATCH /api/articles/v3.0/article_groups/{articleGroupId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleGroupId** *required*	ID of article group.	integer (int64)
Body	**articleGroupEXT** *required*	Data for article group.	ArticleGroup

Responses

HTTP Code	Description	Schema
200	Current state of article group.	ArticleGroupSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- application/json

Create single article type.

```
POST /api/articles/v3.0/article_types
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Body	**articleType** *required*	Data for article type	ArticleType

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	ArticleTypeSingle
201	Entity was created successfully.	ArticleTypeSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	Data conflict, provided state of article type can't be accepted.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- application/json

Read all article types of business unit.

```
GET /api/articles/v3.0/article_types
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Responses

HTTP Code	Description	Schema
200	Current state of article types.	ArticleTypeList
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

GET /api/articles/v3.0/article_types/name/{articleTypeName}

Parameters

Type	Name	Schema
Header	**X-Business-Units** *required*	string
Path	**articleTypeName** *required*	string

Responses

HTTP Code	Description	Schema
200	successful operation	ArticleTypeSingle

Produces

- `application/json`

Get information about single article type.

```
GET /api/articles/v3.0/article_types/{articleTypeId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleTypeId** *required*	ID of article type	integer (int64)

Responses

HTTP Code	Description	Schema
200	Current state of article type.	ArticleTypeSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Put new data of a single article type.

```
PUT /api/articles/v3.0/article_types/{articleTypeId}
```

Description

The maximum value of {articleTypeId} is 32767.

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleTypeId** *required*	ID of article type	integer (int64)
Body	**articleType** *required*	Data for article type	ArticleType

Responses

HTTP Code	Description	Schema
200	Current state of article type (updated).	ArticleTypeSingle
201	Current state of article type (created).	ArticleTypeSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	Data conflict, provided state of article type can't be accepted.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Delete single article type of business unit id.

```
DELETE /api/articles/v3.0/article_types/{articleTypeId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleTypeId** *required*	ID of article type	integer (int64)

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	Data conflict, provided state can't be accepted.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Patch new data of a single article type.

```
PATCH /api/articles/v3.0/article_types/{articleTypeId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleTypeId** *required*	ID of article type	integer (int64)
Body	**articleType** *required*	Data for article type	ArticleType

Responses

HTTP Code	Description	Schema
200	Current state of article type.	ArticleTypeSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	Data conflict, provided state of article type can't be accepted.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- application/json

Read all articles of business unit.

```
GET /api/articles/v3.0/articles
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	ArticleList
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- application/json

Update articles data.

```
PUT /api/articles/v3.0/articles/list
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Type	Name	Description	Schema
Body	**articles** *required*	Data of article to create.	< ArticleInList > array

Responses

HTTP Code	Description	Schema
200	Current state of articles after import: if no new instances.	ArticleList
201	Current state of articles after import: if at least one instance created.	ArticleList
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Consumes

- `application/json`

Produces

- `application/json`

GET /api/articles/v3.0/articles/name/{articleName}

Parameters

Type	Name	Schema
Header	**X-Business-Units** *required*	string
Path	**articleName** *required*	string

Responses

HTTP Code	Description	Schema
200	successful operation	ArticleSingle

Produces

- `application/json`

Patch additional information field by root element of a single article.

```
PATCH /api/articles/v3.0/articles/{articleId}/additionalInformation/{rootElement}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleId** *required*	ID of article.	integer (int64)
Path	**rootElement** *required*	Name of root element.	string
Body	**value** *required*	Json sub-tree of root element.	string

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	object
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	The syntax of the request entity is correct (400 status code is inappropriate), but was unable to process the contained instructions, e.g. attempt to link to nonexistent instance.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Patch additional information field of a single article: search inside of root element and replace the values.

```
PATCH
/api/articles/v3.0/articles/{articleId}/additionalInformation/{rootElement}/{jsonEleme
nt}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleId** *required*	ID of article.	integer (int64)
Path	**jsonElement** *required*	Name of json element(s) to replace.	string
Path	**rootElement** *required*	Name of root element.	string
Body	**value** *required*	Json sub-tree of json - the new value after replacing.	string

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	object
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	The syntax of the request entity is correct (400 status code is inappropriate), but was unable to process the contained instructions, e.g. attempt to link to nonexistent instance.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Get information about single article.

```
GET /api/articles/v3.0/articles/{articleSku}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleSku** *required*	SKU of article	integer (int64)

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	ArticleSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Put new data of a single article.

```
PUT /api/articles/v3.0/articles/{articleSku}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleSku** *required*	SKU of article	integer (int64)
Body	**article** *required*	Data for article	Article

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	ArticleSingle
201	Entity was created successfully.	ArticleSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
422	The syntax of the request entity is correct (400 status code is inappropriate), but was unable to process the contained instructions, e.g. attempt to link to nonexistent instance.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Patch new data of a single article.

```
PATCH /api/articles/v3.0/articles/{articleSku}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**articleSku** *required*	SKU of article.	integer (int64)
Body	**article** *required*	Data for article.	Article

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	ArticleSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse

HTTP Code	Description	Schema
404	Not found, e.g. ID is not found.	ErrorResponse
409	The syntax of the request entity is correct (400 status code is inappropriate), but was unable to process the contained instructions, e.g. attempt to link to nonexistent instance.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Get icons.

```
GET /api/articles/v3.0/icons
```

Parameters

Type	Name	Description	Schema
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Responses

HTTP Code	Description	Schema
200	Current state of icons	IconsList
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Get measurement units.

```
GET /api/articles/v3.0/measurement_units
```

Parameters

Type	Name	Description	Schema
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Responses

HTTP Code	Description	Schema
200	Current state of measurement units.	MeasurementUnitList
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Create single price level.

```
POST /api/articles/v3.0/price_levels
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Body	**PriceLevel** *required*	Data for price level	PriceLevel

Responses

HTTP Code	Description	Schema
200	The request has succeeded.	PriceLevelSingle
201	Entity was created successfully.	PriceLevelSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Read all price levels of business unit.

```
GET /api/articles/v3.0/price_levels
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string

Responses

HTTP Code	Description	Schema
200	Current state of price levels.	PriceLevelList
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Get information about single price level.

```
GET /api/articles/v3.0/price_levels/{PriceLevelId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**PriceLevelId** *required*	ID of price level	integer (int64)

Responses

HTTP Code	Description	Schema
200	Current state of price level.	PriceLevelSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Delete single price level of business unit id.

```
DELETE /api/articles/v3.0/price_levels/{PriceLevelId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**PriceLevelId** *required*	ID of price level	integer (int64)

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
409	Data conflict, provided state can’t be accepted.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Patch new data of a single price level.

```
PATCH /api/articles/v3.0/price_levels/{PriceLevelId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**PriceLevelId** *required*	ID of price level	integer (int64)
Body	**PriceLevel** *required*	Data for price level	PriceLevel

Responses

HTTP Code	Description	Schema
200	Current state of price level.	PriceLevelSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Put new data of a single price level.

```
PUT /api/articles/v3.0/price_levels/{priceLevelId}
```

Parameters

Type	Name	Description	Schema
Header	**X-Business-Units** *required*	Business unit designates one business venue, which data is requested. Make sure your token allows to access the given business unit, you can check it in the cloud.	string

Type	Name	Description	Schema
Header	**X-Token** *required*	You can obtain this token in the cloud.	string
Path	**PriceLevelId** *required*	ID of price level	integer (int64)
Body	**PriceLevel** *required*	Data for price level	PriceLevel

Responses

HTTP Code	Description	Schema
200	Current state of price level (updated).	PriceLevelSingle
201	Current state of price level (created).	PriceLevelSingle
400	Bad Request - the given JSON or params is/are not valid, e.g. businessUnitId is less than 0.	ErrorResponse
404	Not found, e.g. ID is not found.	ErrorResponse
500	Internal server error, e.g. the database could not be accessed.	ErrorResponse

Produces

- `application/json`

Definitions

AggregatedData

Article aggregated data, external API representation.

Name	Description	Schema
items *required*	List of articles with prices in aggregated mode. Patch operation overwrites all of collection at once.	< AggregatedStep > array
shown *optional*	Show constituents on receipt or not. Default false. **Example** : `false`	boolean

AggregatedStep

Single step of article aggregated entity, external API representation.

Name	Description	Schema
article *required*	One article which belongs to the main item with aggregated mode.	SkuName
price *required*	Price of one article item. **Example** : `199.99`	number

Article

Article entity, external API representation.

Name	Description	Schema
active *optional*	True: the article is presented on the POS. False: it is marked in the cloud as deactivated and is not displayed on the POS. Default true. **Example** : `true`	boolean
additionalInformation *optional*	Additional field for storing generic information in json format. Please adhere to the following structure of the JSON: 'yourProductsName': {'element': {'subelement1': 1, 'subelement2': 2}}. Always use similar name of your product as a root element. For existing articles use only patch operations for updating the field. Please note there is a special endpoint of PATCH type for updating the certain root element. **Example** : `"'FoodNotify': {'allergens': {'X': 1, 'Y': 2}"`	object
aggregated *optional*	Aggregated Data, used with mode 'AGGREGATED'.	AggregatedData
articleGroup *required*	Nested object, containing the information about an id and name of the article group, assigned to this article.	IdName

Name	Description	Schema
articleType *required*	Nested object, containing the information about an id and name of the article type, assigned to this article.	IdName
colorCode *optional*	The HEX color code. The defined color is applied to the article's icon changing it's color. **Length** : `0 - 7` **Example** : `"#d60000"`	string
constraints *optional*	Constraints Data, used with mode 'CONSTRAINTS'.	ConstraintData
description *optional*	Description of the article. Can be added by user. This field is called "Info" in the cloud. **Length** : `0 - 4096` **Example** : `"Some details."`	string
image *optional*	It shows the file name of the default image to the article. It can be used to assign the proper icon to the article in the platform interface. To allow the end user to choose the icon they must be downloaded from platform store. **Length** : `0 - 255` **Example** : `"default/pizza.png"`	string
internal *optional*	Internal / Control String. **Example** : `"1000 ml"`	string
measures *optional*	The measurement of article. Customizable in Master Data / Article (Selling-Unit). Default 0. **Example** : `1.0`	number (double)
mode *optional*	Mode of article: 1) "REGULAR" (default) - single type article without any modifiers (relations to other articles). Default state. 2) "CONSTRAINTS" - article with this mode can refer to constraint articles, like salami for pizza. 3) "AGGREGATED" - article with this mode includes aggregated items with defined price for the whole set, e.g. "Happy Meal" or "Breakfast". **Example** : `"CONSTRAINTS"`	enum (REGULAR, CONSTRAINTS, AGGREGATED)
modified *optional* *read-only*	Timestamp of last article modification. For example, it can be used to update only positions, which were changed. The format is "yyyy-MM-dd'T'HH:mm:ss.SSSZ".	string (date-time)
name *required*	The name of the article. Must be assigned by user. **Length** : `1 - 60` **Example** : `"Name of article"`	string
pluAndBarcodes *optional*	List of PLUs and EAN Codes for the article. Patch operation overwrites all of collection at once. Values should be different - equal numbers will be stored as one number.	< string > array
price *optional*	The full price of an article, including taxes. It can include two decimal signs. Default 0. **Example** : `199.99`	number

Name	Description	Schema
priceLevelPrices *optional*	Nested array of objects, each of them contains the information about an id and name of the price level, related to this article. Price level is created separately and causes the change of the article's price in the predifined time period. Patch operation overwrites all of collection at once.	< string, number > map
shortName *required*	The short name of an article. This name is displayed on the POS. Must be assigned by user. **Length** : `1 - 30` **Example** : `"ShortName of article"`	string
sku *optional* *read-only*	SKU (Stock Keeping Unit) of article. Each article has it's own unique SKU, that can't be changed after initial creation of article. **Example** : `100`	integer (int64)
stockAmount *optional*	The amount of article of position, that are available on stock in current business id. **Example** : `100`	integer (int32)

ArticleGroup

Article group entity, external API representation.

Name	Description	Schema
applyColorToArticles *optional*	The boolean indicator, if true the articles of the group will be colored in the specified group's color, if they don't have specified colour. **Example** : `false`	boolean
applyImageToArticles *optional*	The boolean indicator, if true the articles of the group will get the image, provided in the field "image" as a filename, in case if they don't have already assigned image. **Example** : `false`	boolean
articleType *optional*	Default article type for articles inside that group.	IdName
bonPrinters *optional*	Nested array of objects, each of them contains the information about an id and name of the printer, assigned to this group of articles. By bon-printing all articles of current group will be listed on the bon printed on the given printers. If nullified the list will be empty. Patch operation overwrites all of collection at once.	< IdName > array
colorCode *optional*	The HEX color code. The defined color is applied to the icon of all articles of the group, changing their color. Two conditions for this setting to be applied to each article: the boolean attribute applyColorToArticles must be true and article must have no image being assigned. **Length** : `0 - 7` **Example** : `"#d60000"`	string

Name	Description	Schema
discountable *optional*	The boolean indicator, if true the articles of this group will be considered as discountable, if discount is applied. **Example** : `true`	boolean
externalVat *optional*	Nested object, containing the information about an id and name of the VAT rate for articles of the article group, considered to be ordered to take away. In Germany there are separate VAT rates for in house consumption and orders to be taken away. In other countries the content for externalVAT is either being duplicated (rather rarely) or left blank.	IdName
fixedAt *optional*	The boolean indicator, if true the articles of this group will be printed on extra ticket. **Example** : `false`	boolean
id *optional* *read-only*	ID of article group. Each type of article group has it's own unique id. **Example** : `1`	integer (int64)
image *optional*	The file name of the default image, which will be applied to all articles of the group, if they don't have assigned images and the applyImageToArticles is set to true. Used to assign the proper icon to the articles in the platform interface. To allow the end user to choose the icon they must be downloaded from the platform store. **Length** : `0 - 255` **Example** : `"default/rum.png"`	string
internalVat *optional*	Nested object, containing the information about an id and name of the VAT rate for articles of the article group, considered to be consumed in restaurant. In Germany there are separate VAT rates for in house consumption and orders to be taken away. In other countries the content for externalVAT is either being duplicated (rather rarely) or left blank.	IdName
measurementUnit *optional*	Defines the unit of measuring for all of articles that belong to this article-group. Default setting is a value called "piece", that can not be separated (it conforms with the most of PLUs).	IdName
name *required*	Name of article group. Must be assigned by user. **Length** : `0 - 255` **Example** : `"Alkohol. Getränke"`	string
parent *optional*	Nested object, containing the information about an id and name of the parent group of the current article group. Empty field means the current group has no parent groups, thus is a supergroup or parent group itself	IdName
shortName *required*	Short name of article group, used by printing for bills/bons and in some parts of App interface. **Length** : `0 - 100` **Example** : `"Alkohol"`	string

Name	Description	Schema
ticketLayout *optional*	Nested array of objects, each of them contains the information about an id and name of the layout format, assigned to this group of articles for using by printing ordering tickets.	IdName
ticketVoidLayout *optional*	Nested array of objects, each of them contains the information about an id and name of the layout, assigned to this group of articles for using by printing voiding tickets.	IdName

ArticleGroupList

List of article groups in response, external API representation.

Name	Description	Schema
articleGroups *required*	Data of article groups.	< ArticleGroup > array

ArticleGroupSingle

One article group in response, external API representation.

Name	Description	Schema
articleGroup *required*	Article group data.	ArticleGroup

ArticleInList

Article entity, external API representation, in-list form.

Name	Description	Schema
active *optional*	True: the article is presented on the POS. False: it is marked in the cloud as deactivated and is not displayed on the POS. Default true. **Example** : `true`	boolean
additionalInformation *optional*	Additional field for storing generic information in json format. Please adhere to the following structure of the JSON: 'yourProductsName': {'element': {'subelement1': 1, 'subelement2': 2}}. Always use similar name of your product as a root element. For existing articles use only patch operations for updating the field. Please note there is a special endpoint of PATCH type for updating the certain root element. **Example** : `"'FoodNotify': {'allergens': {'X': 1, 'Y': 2}"`	object
aggregated *optional*	Aggregated Data, used with mode 'AGGREGATED'.	AggregatedData

Name	Description	Schema
articleGroup *required*	Nested object, containing the information about an id and name of the article group, assigned to this article.	IdName
articleType *required*	Nested object, containing the information about an id and name of the article type, assigned to this article.	IdName
colorCode *optional*	The HEX color code. The defined color is applied to the article's icon changing it's color. **Length** : `0 - 7` **Example** : `"#d60000"`	string
constraints *optional*	Constraints Data, used with mode 'CONSTRAINTS'.	ConstraintData
description *optional*	Description of the article. Can be added by user. This field is called "Info" in the cloud. **Length** : `0 - 4096` **Example** : `"Some details."`	string
image *optional*	It shows the file name of the default image to the article. It can be used to assign the proper icon to the article in the platform interface. To allow the end user to choose the icon they must be downloaded from platform store. **Length** : `0 - 255` **Example** : `"default/pizza.png"`	string
internal *optional*	Internal / Control String. **Example** : `"1000 ml"`	string
measures *optional*	The measurement of article. Customizable in Master Data / Article (Selling-Unit). Default 0. **Example** : `1.0`	number (double)
mode *optional*	Mode of article: 1) "REGULAR" (default) - single type article without any modifiers (relations to other articles). Default state. 2) "CONSTRAINTS" - article with this mode can refer to constraint articles, like salami for pizza. 3) "AGGREGATED" - article with this mode includes aggregated items with defined price for the whole set, e.g. "Happy Meal" or "Breakfast". **Example** : `"CONSTRAINTS"`	enum (REGULAR, CONSTRAINTS, AGGREGATED)
modified *optional* *read-only*	Timestamp of last article modification. For example, it can be used to update only positions, which were changed. The format is "yyyy-MM-dd'T'HH:mm:ss.SSSZ".	string (date-time)
name *required*	The name of the article. Must be assigned by user. **Length** : `1 - 60` **Example** : `"Name of article"`	string
pluAndBarcodes *optional*	List of PLUs and EAN Codes for the article. Patch operation overwrites all of collection at once. Values should be different - equal numbers will be stored as one number.	< string > array
price *optional*	The full price of an article, including taxes. It can include two decimal signs. Default 0. **Example** : `199.99`	number

Name	Description	Schema
priceLevelPrices *optional*	Nested array of objects, each of them contains the information about an id and name of the price level, related to this article. Price level is created separately and causes the change of the article's price in the predifined time period. Patch operation overwrites all of collection at once.	< string, number > map
shortName *required*	The short name of an article. This name is displayed on the POS. Must be assigned by user. **Length** : `1 - 30` **Example** : `"ShortName of article"`	string
sku *optional*	SKU - stock keeping unit. Can be specified by creating the new position and can't be changed anymore. **Example** : `100`	integer (int64)
stockAmount *optional*	The amount of article of position, that are available on stock in current business id. **Example** : `100`	integer (int32)

ArticleList

List of articles in response, external API representation.

Name	Description	Schema
articles *required*	Data of articles.	< Article > array

ArticleSingle

One article in response, external API representation.

Name	Description	Schema
article *required*	Article data.	Article

ArticleType

Article type entity, external API representation.

Name	Description	Schema
constraintOnly *optional*	True: the article of this type has special properties on the POS. It is not displayed on layout by default and can be hidden on the receipts. The parameter is used for optional or constraint articles, for instance, to present some component of main article, like salami for pizza, or the state of article, like spicy. Default false. **Example** : `false`	boolean

Name	Description	Schema
description *optional*	Description of the article type. Can be added by user. **Length** : `0 - 255` **Example** : `"Some details."`	string
fixedName *optional*	True: the article of this type is presented on the POS as free name article. When choosing such kind of articles the user is asked to input the name. Default true. **Example** : `true`	boolean
fixedPrice *optional*	True: the article of this type is presented on the POS as free price article. When choosing such kind of articles the user is asked to input the price. Default true. **Example** : `true`	boolean
id *optional* *read-only*	ID of article type. Each type of article has it's own unique id. **Maximum value** : `32767` **Example** : `1`	integer (int64)
info *optional*	True: the article of this type is presented on the POS as info article. Info articles are shown additionally during the ordering process, for instance, the user can choose info article "hot" or "cold" or "sugar-free" additionally to other article. Default false. **Example** : `false`	boolean
measure *optional*	True: the article type is measurable. Default false. **Example** : `false`	boolean
name *required*	Name of article type. Must be assigned by user. **Length** : `0 - 40` **Example** : `"free price and free name"`	string
nonTurnover *optional*	True: the article participated in the turnover. The "true" values are actual for special kinds of articles, like Tobacco. Default false. **Example** : `false`	boolean
voucher *optional*	True: the article of this type is presented in reports as voucher type. It is used to distinguish between regular non-turnover articles and non-turnover voucher articles in reports. You can not enable "Voucher" while "Non Revenue" is disabled. The field is used in section "VOUCHER AMOUNT PER TENDER". Default false. **Example** : `false`	boolean

ArticleTypeList

List of article types in response, external API representation.

Name	Description	Schema
articleTypes *required*	Data of article types.	< ArticleType > array

ArticleTypeSingle

One article type in response, external API representation.

Name	Description	Schema
articleType *required*	Article type data.	ArticleType

ConstraintData

Article constraints definition entity, external API representation.

Name	Description	Schema
deleteOriginal *optional*	True: then the id (SKU) of the main article will be deleted after making an order, only SKUs of constrain articles will be displayed. Default false. **Example** : `false`	boolean
newPos *optional*	True: then each of the constraint articles will be presented as a separate position on the POS. That means the user will see Cola and Ice instead of Cola, Ice position. Default false. **Example** : `false`	boolean
steps *required*	Nested array of objects, that represent constraint articles and settings of steps ordering process, which will be displayed on POS in process of ordering the main article, which has assigned constraint articles. Patch operation overwrites all of collection at once.	< ConstraintStep > array

ConstraintStep

Single step of article constraint entity, external API representation.

Name	Description	Schema
articles *required*	Nested array of objects, that represent constraint articles, which will be displayed on POS in process of ordering of the main article.	< SkuName > array
multipleSelects *optional*	True: it's possible to choose more than one constraint article at each step. Default false. **Example** : `false`	boolean
name *optional*	Name of the step in the process of ordering with constraints. For instance, first step for burger would be - choose the bun. **Length** : `1 - 60` **Example** : `"Depot"`	string
optional *optional*	True: the articles of each step are not necessarily must be chosen, i.e. can be skipped. Default false. **Example** : `false`	boolean

ErrorResponse

Name	Description	Schema
errorMessage *required*	Error message as text. **Example** : `"A database error occurred."`	string

Icon

Icon in response, external API representation.

Name	Description	Schema
name *optional*	Filename of icon. **Example** : `"fishdish.png"`	string

IconsList

List of icons in response, external API representation.

Name	Description	Schema
icons *required*	Data of icons.	< Icon > array

IdName

Name	Description	Schema
id *required*	ID of instance. **Example** : `13`	integer (int64)
name *optional* *read-only*	Name of instance. **Example** : `"NAME"`	string

MeasurementUnit

One measurement unit in response, external API representation.

Name	Description	Schema
baseMeasurement *optional*	**Example** : `16.0`	number (double)
baseUnit *optional*		PairIdName

Name	Description	Schema
id *required*	The id of unit, measuring the quantity of article, if it was assigned. Customizable in Master Data / Article Group (the article type with activated “Measure” should be created in Setup / Article-Type and assigned to the current article). **Example** : `7`	integer (int64)
name *required*	The name of unit, measuring the quantity of article, if it was assigned. Customizable in Master Data / Article Group (the article type with activated “Measure” should be created in Setup / Article-Type and assigned to the current article). **Length** : `0 - 255` **Example** : `"ounce"`	string

MeasurementUnitList

List of measurement units in response, external API representation.

Name	Description	Schema
measurementUnits *required*	Data of neasurement units.	< MeasurementUnit > array

PairIdName

Name	Description	Schema
id *required*	ID of instance. **Example** : `13`	integer (int64)
name *optional* *read-only*	Name of instance. **Example** : `"NAME"`	string

PriceLevel

Price level entity, external API representation.

Name	Description	Schema
active *optional*	The boolean indicator, if true the Price Level is available. **Example** : `true`	boolean
description *optional*	A short description for the given price level. May be entered by user. **Length** : `0 - 255` **Example** : `"Weekday, daytime"`	string
friday *optional*	The boolean indicator, if true the Price Level is active on Fridays, so that the entered special price for each article will be used also on Fridays in the given time. **Example** : `false`	boolean

Name	Description	Schema
id *optional* *read-only*	ID of price level. Each price level has it's own unique id. **Maximum value** : `32767` **Example** : `1`	integer (int64)
monday *optional*	The boolean indicator, if true the Price Level is active on Mondays, so that the entered special price for each article will be used also on Mondays in the given time. **Example** : `true`	boolean
name *required*	Name of price level. Must be assigned by user. **Length** : `0 - 255` **Example** : `"Happy Hour"`	string
saturday *optional*	The boolean indicator, if true the Price Level is active on Saturdays, so that the entered special price for each article will be used also on Saturdays in the given time. **Example** : `false`	boolean
sunday *optional*	The boolean indicator, if true the Price Level is active on Sundays, so that the entered special price for each article will be used also on Sundays in the given time. **Example** : `true`	boolean
thursday *optional*	The boolean indicator, if true the Price Level is active on Thursdays, so that the entered special price for each article will be used also on Thursdays in the given time. **Example** : `true`	boolean
timeBegin *optional*	Time since price level is active, in minutes (hours * 60 + minutes). Default 0. **Example** : `480`	integer (int32)
timeEnd *optional*	Time till price level is active, in minutes (hours * 60 + minutes). Default 1439. **Example** : `960`	integer (int32)
tuesday *optional*	The boolean indicator, if true the Price Level is active on Tuesdays, so that the entered special price for each article will be used also on Tuesdays in the given time. **Example** : `true`	boolean
wednesday *optional*	The boolean indicator, if true the Price Level is active on Wednesdays, so that the entered special price for each article will be used also on Wednesdays in the given time. **Example** : `true`	boolean

PriceLevelList

List of price levels in response, external API representation.

Name	Description	Schema
priceLevels *required*	Data of price levels.	< PriceLevel > array

PriceLevelSingle

List of articles in response, external API representation.

Name	Description	Schema
priceLevel *required*	Price level data.	PriceLevel

SkuName

Name	Description	Schema
name *optional* *read-only*	Name of instance. **Example** : `"NAME"`	string
sku *required*	SKU - stock keeping unit. Can be specified by creating the new position and can’t be changed anymore. **Example** : `10`	integer (int64)

.

Appendix A: Image Library API

It can be used to download Icons.

Download Urls:

- */icash/static/icons/icons.zip* - file which contains folders and icons with default size (40x40)
- */icash/static/icons/icons2x.zip* - file which contains folders and icons with double size (80x80)
- */icash/static/icons/icons3x.zip* - file which contains folders and icons with triple size (120x120)

The server has ETag support. The response contains ETag header (e.g., ETag: "592c1371-3dc34").
If client sends a header If-None-Match with value from previous response (like If-None-Match: "592c1371-3dc34") and value is equal to the current ETag the server will return HTTP code 304 instead of sending the file.