

# Shop

## Overview

Endpoint per le operazioni sullo shop

## Version information

*Version* : 1.2.0

## URI scheme

*BasePath* : /shop

## Tags

- Item e prodotti

## Paths

# Ricerca degli item secondo alcuni criteri

```
GET /shop/items/
```

## Description

Ricerca tra gli item quelli che soddisfano i criteri richiesti

## Parameters

Type	Name	Description	Schema
Query	<code>page_size</code> <i>required</i>	Numero di item da restituire	number (int)
Query	<code>page_number</code> <i>required</i>	Offset per paginazione	number (int)
Query	<code>category</code> <i>optional</i>	Categoria dell'item	string
Query	<code>name</code> <i>optional</i>	Stringa che l'item deve contenere nel nome	string
Query	<code>price_asc</code> <i>optional</i>	Ordina per prezzo crescente	boolean
Query	<code>price_desc</code> <i>optional</i>	Ordina per prezzo decrescente	boolean
Query	<code>name_asc</code> <i>optional</i>	Ordina per nome in ordine alfabetico	boolean
Query	<code>name_desc</code> <i>optional</i>	Ordina per nome in ordine alfabetico inverso	boolean

## Responses

HTTP Code	Description	Schema
200	OK	< <a href="#">Item</a> > array
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Item e prodotti

# Creazione di un nuovo item nello shop

POST /shop/items/

## Description

Gestisce l'inserimento di un nuovo item e i prodotti inclusi.

Richiede permessi ["operator", "shop\_write"]

## Body parameter

Dati dell'item

*Name* : body

*Flags* : required

Name	Schema
<b>name</b> <i>required</i>	string
<b>description</b> <i>optional</i>	string
<b>category</b> <i>required</i>	string
<b>relevance</b> <i>optional</i>	number
<b>products</b> <i>required</i>	< <a href="#">products</a> > array

## products

Name	Schema
<b>barcode</b> <i>required</i>	string
<b>name</b> <i>required</i>	string
<b>description</b> <i>optional</i>	string
<b>price</b> <i>required</i>	number
<b>quantity</b> <i>required</i>	number
<b>target_species</b> <i>optional</i>	< string > array
<b>images</b> <i>optional</i>	< <a href="#">images</a> > array

## images

Name	Schema
path <i>required</i>	string
description <i>required</i>	string

## Responses

HTTP Code	Description	Schema
201	Inserito con successo Headers : <b>location</b> (string) : Endpoint da cui è possibile accedere all'item.	<a href="#">Item</a>
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
409	Il barcode di uno dei prodotti è già esistente	<a href="#">FieldErrorMessage</a>
500	Errore lato server	No Content

## Consumes

- **application/json**

## Produces

- **application/json**

## Tags

- Item e prodotti

# Ricerca di item

```
GET /shop/items/{item_id}
```

## Description

Ricerca di un item dato il suo id.

## Parameters

Type	Name	Description	Schema
Path	<code>item_id</code> <i>required</i>	Id dell'item	string (ObjectId)

## Responses

HTTP Code	Description	Schema
200	Item ricercato	< <a href="#">Item</a> > array
400	Parametri malformati	<a href="#">ValidatorErrorMessage</a>
404	Nessun item trovato	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Item e prodotti

# Modifica di un item

```
PUT /shop/items/{item_id}
```

## Description

Gestisce la modifica dei dati di un item.

Richiede permessi ["operator", "shop\_write"]

## Parameters

Type	Name	Description	Schema
Path	<b>item_id</b> <i>required</i>	Id dell'item da modificare	string (ObjectId)

## Body parameter

Dati dell'item

*Name* : body

*Flags* : optional

Name	Schema
<b>name</b> <i>optional</i>	string
<b>description</b> <i>optional</i>	string
<b>category</b> <i>optional</i>	string
<b>relevance</b> <i>optional</i>	number (int)
<b>products</b> <i>optional</i>	< <a href="#">products</a> > array

## products

Name	Schema
<b>barcode</b> <i>optional</i>	string
<b>name</b> <i>optional</i>	string
<b>description</b> <i>optional</i>	string
<b>target_species</b> <i>optional</i>	< string > array
<b>price</b> <i>optional</i>	number
<b>quantity</b> <i>optional</i>	number

Name	Schema
<code>images</code> <i>optional</i>	<code>&lt; images &gt; array</code>

`images`

Name	Schema
<code>path</code> <i>required</i>	<code>string</code>
<code>description</code> <i>required</i>	<code>string</code>

## Responses

HTTP Code	Description	Schema
200	Modificato con successo	<code>Item</code>
400	Parametri malformati	<code>ValidationErrorMessage</code>
401	Non autenticato	<code>ErrorMessage</code>
403	Permessi insufficienti	<code>ErrorMessage</code>
404	Nessun item trovato	<code>ErrorMessage</code>
500	Errore lato server	<code>No Content</code>

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Item e prodotti

## Cancellazione di un item

```
DELETE /shop/items/{item_id}
```

### Description

Gestisce la cancellazione di un item e di tutti i suoi prodotti associati.

Richiede permessi ["operator", "shop\_write"]

### Parameters

Type	Name	Description	Schema
Path	<code>item_id</code> <i>required</i>	Id dell'item da cancellare	string (ObjectId)

### Responses

HTTP Code	Description	Schema
204	Cancellato con successo	No Content
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Nessun item trovato	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

### Tags

- Item e prodotti



# Ricerca di un item per barcode di un prodotto

```
GET /shop/items/barcode/{barcode}
```

## Description

Ricerca l'item che contiene un determinato prodotto ricercato per barcode.

Richiede permessi ["operator", "shop\_read"]

## Parameters

Type	Name	Description	Schema
Path	barcode <i>required</i>	Barcode del prodotto	string

## Responses

HTTP Code	Description	Schema
200	OK	<a href="#">Item</a>
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Nessun item trovato	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Item e prodotti

## Lista di tutte le categorie

```
GET /shop/categories/
```

### Description

Restituisce la lista di tutte le categorie disponibili.

### Responses

HTTP Code	Description	Schema
200	Lista di tutte le categorie disponibili	< <a href="#">Category</a> > array
500	Internal server error	No Content

### Produces

- [application/json](#)

### Tags

- Categorie

# Inserimento di categorie

POST /shop/categories/

## Description

Creazione di una nuova categoria.

Richiede permessi ["operator", "shop\_write"]

## Body parameter

Dati della categoria

*Name* : body

*Flags* : required

*Type* : [Category](#)

## Responses

HTTP Code	Description	Schema
201	Creato con successo	<a href="#">Category</a>
400	Input malformato	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi mancanti	<a href="#">ErrorMessage</a>
409	Categoria con lo stesso nome già esistente	<a href="#">FieldErrorMessage</a>
500	Internal server error	No Content

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Categorie

# Aggiornamento di categorie

PUT /shop/categories/{category}

## Description

Aggiornamento di una categoria.

Richiede permessi ["operator", "shop\_write"]

## Parameters

Type	Name	Description	Schema
Path	category <i>required</i>	Categoria da aggiornare	string

## Body parameter

Dati da aggiornare della categoria

*Name* : body

*Flags* : optional

*Type* : [Category](#)

## Responses

HTTP Code	Description	Schema
200	Aggiornato con successo	<a href="#">Category</a>
400	Input malformato	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi mancanti	<a href="#">ErrorMessage</a>
404	Categoria non esistente	<a href="#">ErrorMessage</a>
500	Internal server error	No Content

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Categorie

# Cancellazione di categorie

```
DELETE /shop/categories/{category}
```

## Description

Cancellazione di una categoria.

Richiede permessi ["operator", "shop\_write"]

## Parameters

Type	Name	Description	Schema
Path	<code>category</code> <i>required</i>	Categoria da cancellare	string

## Responses

HTTP Code	Description	Schema
204	Cancellato con successo	No Content
400	Input malformato	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi mancanti	<a href="#">ErrorMessage</a>
404	Categoria non esistente	<a href="#">ErrorMessage</a>
500	Internal server error	No Content

## Tags

- Categorie

# Ricerca di ordini

```
GET /shop/orders/
```

## Description

Gestisce la ricerca degli ordini, ordinati in senso decrescente.

I clienti posso cercare solo i propri ordini, gli operatori tutti.

Richiede permessi [ ["customer"] ], [ ["admin"], ["operator", "warehouse"] ]

## Parameters

Type	Name	Description	Schema
Query	<code>page_size</code> <i>required</i>	Numero di post per pagina	number (int)
Query	<code>page_number</code> <i>required</i>	Numero di pagina	number (int)
Query	<code>status</code> <i>optional</i>	Stato dell'ordine	string
Query	<code>start_date</code> <i>optional</i>	Inizio intervallo temporale di ricerca	string (date-time)
Query	<code>end_date</code> <i>optional</i>	Fine intervallo temporale di ricerca	string (date-time)
Query	<code>customer</code> <i>optional</i>	Username dell'autore dell'ordine	string

## Responses

HTTP Code	Description	Schema
200	Trovato con successo	< <a href="#">Order</a> > array
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Ordini

# Creazione di ordini

POST /shop/orders/

## Description

Gestisce la creazione di un nuovo ordine associato al cliente.

Richiede permessi [ ["customer"] ] , [ ["admin"] ]

## Body parameter

Name : body

Flags : required

Name	Description	Schema
products <i>required</i>		< <a href="#">products</a> > array
pickup <i>required</i>	true per ritiro in sede, false per consegna	boolean
hub_code <i>optional</i>	Codice dell'hub di ritiro. Obbligatorio se si richiede il ritiro	string
address <i>optional</i>	Indirizzo di consegna. Obbligatorio se si richiede la consegna	<a href="#">Address</a>

## products

Name	Schema
barcode <i>required</i>	string
quantity <i>required</i>	number

## Responses

HTTP Code	Description	Schema
200	Inserito con successo Headers : <a href="#">location</a> (string) : Endpoint da cui è possibile accedere ai dati dell'ordine.	<a href="#">Order</a>
400	Parametri malformati, quantità insufficiente	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Parametri fanno riferimento a entità inesistenti	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Consumes

- [application/json](#)

## Produces

- `application/json`

## Tags

- Ordini



# Ricerca ordine per id

```
GET /shop/orders/{order_id}
```

## Description

Gestisce la ricerca di uno specifico ordine per id.

I clienti possono ricercare solo i propri ordini, gli operatori tutti.

Richiede permessi [ ["customer"] ], [ ["admin"], ["operator", "warehouse"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>order_id</code> <i>required</i>	Id dell'ordine	string (ObjectId)

## Responses

HTTP Code	Description	Schema
200	Trovato con successo	<a href="#">Order</a>
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Ordine inesistente	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Ordini

# Aggiornamento ordine per id

```
PUT /shop/orders/{order_id}
```

## Description

Gestisce l'aggiornamento di uno specifico ordine per id.

Richiede permessi [], [ ["admin"], ["operator", "warehouse"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>order_id</code> <i>required</i>	Id dell'ordine	string (ObjectId)

## Body parameter

*Name* : body

*Flags* : required

Name	Schema
<code>status</code> <i>optional</i>	string

## Responses

HTTP Code	Description	Schema
200	Aggiornato con successo	<a href="#">Order</a>
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Ordine inesistente	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Ordini

# Cancellazione ordine per id

```
DELETE /shop/orders/{order_id}
```

## Description

Gestisce la cancellazione di uno specifico ordine per id.

I clienti possono cancellare i propri ordini che non sono in lavorazione, gli operatori tutti.

Richiede permessi [ ["customer"] ], [ ["admin"], ["operator", "warehouse"] ]

## Parameters

Type	Name	Description	Schema
Path	<code>order_id</code> <i>required</i>	Id dell'ordine	string (ObjectId)

## Responses

HTTP Code	Description	Schema
204	Cancellato con successo	No Content
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Ordine inesistente	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Ordini

# Ricerca contenuto carrello

```
GET /users/customers/{username}/cart/
```

## Description

Gestisce la ricerca del contenuto nel carrello dell'utente.

Nota: non controlla la presenza di inconsistenze tra quantità richiesta e quantità disponibile.

Richiede permessi [ ["customer"] ], [ ["admin"] ]

## Parameters

Type	Name	Schema
Path	<code>username</code> <i>required</i>	string

## Responses

HTTP Code	Description	Schema
200	Carrello dell'utente	<a href="#">Cart</a>
400	Parametri malformati	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Utente inesistente	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Produces

- `application/json`

## Tags

- Carrello

# Modifica carrello

```
PUT /users/customers/{username}/cart/
```

## Description

Gestisce la modifica del contenuto nel carrello

Richiede permessi [ ["customer"] ], [ ["admin"] ]

## Parameters

Type	Name	Schema
Path	<code>username</code> <i>required</i>	string

## Body parameter

*Name* : body

*Flags* : required

*Type* : < [body](#) > array

body

Name	Schema
<code>barcode</code> <i>required</i>	string
<code>quantity</code> <i>required</i>	number

## Responses

HTTP Code	Description	Schema
200	Modificato con successo	<a href="#">Cart</a>
400	Parametri malformati, quantità insufficiente	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Utente o prodotto inesistente	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Consumes

- [application/json](#)

## Produces

- [application/json](#)

## Tags

- Carrello

# Inserimento in carrello

```
POST /users/customers/{username}/cart/
```

## Description

Gestisce l'inserimento di un prodotto nel carrello. Se già presente, la quantità viene aggiornata.

Richiede permessi [ ["customer"] ], [ ["admin"] ]

## Parameters

Type	Name	Schema
Path	<b>username</b> <i>required</i>	string

## Body parameter

*Name* : body

*Flags* : required

Name	Schema
<b>barcode</b> <i>required</i>	string
<b>quantity</b> <i>required</i>	number

## Responses

HTTP Code	Description	Schema
200	Inserito con successo	<a href="#">Cart</a>
400	Parametri malformati, quantità insufficiente	<a href="#">ValidationErrorMessage</a>
401	Non autenticato	<a href="#">ErrorMessage</a>
403	Permessi insufficienti	<a href="#">ErrorMessage</a>
404	Utente o prodotto inesistente	<a href="#">ErrorMessage</a>
500	Errore lato server	No Content

## Consumes

- `application/json`

## Produces

- `application/json`

## Tags

- Carrello

# Definitions

## Category

Name	Description	Schema
<b>name</b> <i>required</i>		string
<b>icon</b> <i>required</i>	Pattern : <code>"^([A-Za-z0-9+/{4})*([A-Za-z0-9+/{2}== [A-Za-z0-9+/{3}=])?\$"</code>	string (byte)

## Item

Name	Schema
<b>id</b> <i>required</i>	<code>ObjectId</code>
<b>name</b> <i>required</i>	string
<b>description</b> <i>required</i>	string
<b>category</b> <i>required</i>	string
<b>relevance</b> <i>required</i>	number
<b>products</b> <i>required</i>	< <code>Product</code> > array

## Product

Name	Schema
<b>barcode</b> <i>required</i>	string
<b>name</b> <i>required</i>	string
<b>description</b> <i>required</i>	string
<b>images</b> <i>required</i>	< <code>images</code> > array
<b>target_species</b> <i>required</i>	< string > array
<b>price</b> <i>required</i>	number
<b>quantity</b> <i>required</i>	number

### images

Name	Schema
<b>path</b> <i>required</i>	string



Name	Schema
<b>description</b> <i>required</i>	string

## Order

Name	Description	Schema
<b>id</b> <i>required</i>		ObjectId
<b>customer</b> <i>required</i>	Username del cliente	string
<b>products</b> <i>required</i>	Informazioni sui prodotti dell'ordine (snapshot di ciò che c'era al momento dell'ordine)	< <b>products</b> > array
<b>total</b> <i>required</i>		number
<b>pickup</b> <i>required</i>	true per ritiro in sede, false per consegna	boolean
<b>hub_code</b> <i>optional</i>	Codice dell'hub in cui ritirare	string
<b>address</b> <i>optional</i>	Indirizzo di consegna	string
<b>status</b> <i>required</i>	Stato dell'ordine	string
<b>creationDate</b> <i>required</i>		string (date-time)

### products

Name	Schema
<b>barcode</b> <i>required</i>	string
<b>name</b> <i>required</i>	string
<b>price</b> <i>required</i>	number
<b>quantity</b> <i>required</i>	number

## Address

Name	Schema
<b>city</b> <i>optional</i>	string
<b>street</b> <i>optional</i>	string
<b>number</b> <i>optional</i>	string
<b>postalcode</b> <i>optional</i>	string

## Cart

*Type* : < [Cart](#) > array

### Cart

Name	Schema
<b>product</b> <i>required</i>	<a href="#">Product</a>
<b>quantity</b> <i>required</i>	number

## ValidationErrorMessage

*Type* : < [FieldErrorMessage](#) > array

## FieldErrorMessage

Name	Description	Schema
<b>field</b> <i>required</i>	Il campo errato	string
<b>message</b> <i>optional</i>	Informazioni sull'errore	string

## ErrorMessage

Name	Description	Schema
<b>message</b> <i>optional</i>	Informazioni sull'errore	string

## ObjectId

*Type* : string (ObjectId)