

# ITEMS-MANA GER

～  
機能一覧  
～

# ITEMS-MANAGERとは

商品情報を管理するためのWebAPIです。

商品情報を取得・登録・更新・削除などのCRUD処理が全て実装されています。

商品情報は永続的に保存できるよう、データベースで管理することを想定して開発しております。

また、商品情報は管理者以外の者が取り扱うことが出来ないよう、ユーザー登録機能やログイン機能を実装しております。

パスワードの保存方法については、ハッシュ化して管理するなど、第三者にパスワードが解読されないようセキュリティも強化しておりますので、安心してお使いいただけます。

このWebAPIが、あなたにとって役立つことを祈っております。

# ユーザー登録(Create User)

POST

/auth/signup

Create User

⌵

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{  "username": "user1",  "password": "test1234"}
```

Responses

Code	Description	Links
201	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{  "id": 1,  "username": "user1",  "created_at": "2024-03-13T14:45:58.585Z",  "updated_at": "2024-03-13T14:45:58.585Z"}
```

機能名: ユーザー(管理者)を登録

処理内容:

- ・入力フォームからjson形式でユーザー情報を受け取る
- ・ユーザー情報をusersテーブルに追加

必要なデータ:

- ・ユーザー情報

注意事項:

- ・ユーザー名にユニーク制約をかけているため、同一名で登録するとエラーが発生する

ユーザー操作:

- ・右上にある「Try it out」ボタンをクリック
- ・サンプルのユーザー名とパスワードを変更し、「Execute」を実行

# ログイン(Login)

POST

/auth/login Login

Cancel

Reset

Parameters

No parameters

Request body required

application/x-www-form-urlencoded

grant\_type

grant\_type

☒ Send empty value

username \* required

string

username

password \* required

string

password

scope

string

scope

☒ Send empty value

client\_id

client\_id

☒ Send empty value

client\_secret

client\_secret

☒ Send empty value

Execute

機能名: ユーザーのログイン認証を行う

## 処理内容:

- ・入力フォームからjson形式でユーザー情報を受け取る
  - ※usernameとpassword以外は空欄の状態の問題ありません。
- ・DBから一致するデータを取得し、ログイン認証を行う

## 必要なデータ:

- ・ユーザー情報

## ユーザー操作:

- ・username: ユーザー名を入力
- ・password: パスワードを入力
- ・「Execute」をクリックし、実行
- ※ページの一番右上にある「Authorize」ボタン からログイン認証を行うことができます

# 商品検索(Find All)

GET

/items Find All

⌵

Parameters

最初は「Try it out」と表示されている→

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/items' \
  -H 'accept: application/json'
```

Request URL

```
http://127.0.0.1:8000/items
```

Server response

Code

Details

200

Response body

```
[
  {
    "id": 1,
    "name": "PC",
    "price": 100000,
    "description": "2023年の夏モデルです。",
    "category": "PC",
    "status": "ON_SALE",
    "stock": 100,
    "created_at": "2024-03-13T11:28:42.049354",
    "updated_at": "2024-03-13T11:28:42.049354",
    "user_id": 1
  }
]
```

Download

機能名: 商品情報を検索する

処理内容:

- DBから全商品情報を取得し表示する

必要なデータ:

- 商品情報  
→ 予めDBに登録する必要がある

ユーザー操作:

- 右上にある「Try it out」ボタンをクリック
- 「Execute」ボタンを押して実行

# 商品ID検索(Find By ID)

GET

/items/{id} Find By Id

🔒 ^

Parameters

Cancel

Name	Description
<div><div>id <small>★ required</small></div><div>integer (path)</div></div>	<div><div>1</div></div>

Execute

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header

Example Value | Schema

```
{  "id": 1,  "name": "PC",  "price": 100000,  "description": "2023年の夏モデルです。",  "category": "PC",  "status": "ON_SALE",  "stock": 100,  "created_at": "2024-03-13T15:51:04.912Z",  "updated_at": "2024-03-13T15:51:04.912Z",  "user_id": 0}
```

機能名: 商品IDで検索する

処理内容:

- ・入力フォームから商品IDを受け取る
- ・商品IDに紐づく商品データをDBから取得する
- ・対象データを表示する

必要なデータ:

- ・商品ID

ユーザー操作:

- ・id: 商品IDを入力
- ・「Execute」をクリックし実行
- ・検索結果がない場合は[“Item not found”]と表示される

# キーワード検索(Find By Name)

GET

/items/ Find By Name

↑

Parameters

Cancel

Name	Description
<b>name</b> * required	
string	<input type="text" value="name"/>
(query)	
minLength: 2	
maxLength: 20	

Execute

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header:

Example Value | Schema

```
[
  {
    "id": 1,
    "name": "PC",
    "price": 100000,
    "description": "2023年の夏モデルです。",
    "category": "PC",
    "status": "ON_SALE",
    "stock": 100,
    "created_at": "2024-03-13T20:53:26.785Z",
    "updated_at": "2024-03-13T20:53:26.785Z",
    "user_id": 0
  }
]
```

機能名: 商品名で検索する(部分一致)

処理内容:

- ・入力フォームから検索ワードを受け取る
- ・検索ワードに部分一致した商品名に紐づく商品データをDBから取得する
- ・対象データを表示する

必要なデータ:

- ・検索ワード
- ※2文字以下の文字列を入力するとエラーになる

ユーザー操作:

- ・商品を検索するキーワードを入力
- ・「Execute」をクリックし、実行
- ・検索結果がない場合は、「Item not found」と表示される

# 商品を登録(Create)

POST

/items Create

🔒 ⬆

Parameters

Cancel

No parameters

Request body required

application/json ⬇

```
{  "name": "PC",  "price": 100000,  "description": "2023年の夏モデルです。",  "category": "PC",  "status": "ON_SALE",  "stock": 100}
```

Execute

Responses

Code	Description	Links
201	Successful Response <div><div>Media type<div>application/json ⬇</div></div><div>Controls Accept header:</div><div>Example Value   Schema</div></div>	No links

機能名: 商品情報を登録する

処理内容:

- ・入力フォームから商品情報を受け取る
- ・json形式で取得したデータをDBに登録する

必要なデータ:

- ・商品情報

ユーザー操作:

- ・name: 商品名を入力
- ・price: 商品価格を入力
- ・description: 商品説明を入力
- ・category: 商品カテゴリーを入力
  - ※カテゴリーは、schemasの**ItemCategory**から選択してください
- ・status: 商品ステータスを入力
  - ※ステータスは、schemasの**ItemStatus**から選択してください
- ・stock: 商品在庫を入力

・「Execute」をクリックし、実行  
→登録処理に成功した場合は201番が返却



## 商品情報を更新(Update)

PUT

/items/[id] Update

PUT

Update

Parameters

Cancel

Name	Description
<div><div>id</div><div><div>*</div> required</div></div> <div><div>integer</div><div>(path)</div></div> <div><div>1</div></div>	

Request body

required

application/json

```
{
  "name": "PC",
  "price": 100000,
  "description": "2023年の夏モデルです。",
  "category": "Pc",
  "status": "ON_SALE",
  "stock": 100
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:8000/items/1' \
  -H 'accept: application/json' \
  -d '{
    "name": "PC",
    "price": 100000,
    "description": "2023年の夏モデルです。",
    "category": "Pc",
    "status": "ON_SALE",
    "stock": 100
  }'
```

機能名:商品情報を更新する

**处理内容:**

- ・入力フォームから商品情報を受け取る
- ・json形式でデータを受け取り、入力されている項目のみデータを更新する。

### 必要なデータ:

- ・商品情報

**ユーザー操作:**

- ・変更項目のみ入力を行う
- ・変更しない項目は、“param”: “value”ごとjsonから削除する
- ・「Execute」をクリックし、実行
  - 更新処理に成功した場合は200番が返却
- ・商品情報が変わられていることを確認する
- ・さらにFind ALLで商品検索を行い、対象商品のデータが更新されていることを確認する

# 商品情報を削除>Delete

**DELETE** /items/{id} Delete

Parameters

Name	Description
<b>id</b> * required integer (path)	<input type="text" value="id"/>

Execute

Responses

Code	Description	Links
200	Successful Response	No links

Media type

Controls Accept header.

Example Value

Schema

```
{
  "id": 1,
  "name": "PC",
  "price": 50000,
  "description": "2023年の夏モデルです。",
  "category": "PC",
  "status": "ON_SALE",
  "stock": 100,
  "created_at": "2024-03-13T20:43:52.836Z",
  "updated_at": "2024-03-13T20:43:52.836Z",
  "user_id": 0
}
```

機能名: 商品情報を削除する

処理内容:

- ・入力フォームから商品IDを受け取る
- ・商品IDに紐づく商品情報をDBから取得し、DBから対象データの削除処理を行う

必要なデータ:

- ・商品ID

ユーザー操作:

- ・商品IDを入力する
- ・「Execute」をクリックし、実行  
→削除処理に成功した場合は200番が返却
- ・Find ALLで商品検索を行い、対象商品のデータが削除されていることを確認する

実装機能は以上になります。