API Documentation

Most updated API documentation can be found at: https://documenter.getpostman.com/view/1901388/ece-inventory-system/6YwyvcJ

Basic Format:

API endpoints always start with /api
Authentication related: /api/auth/…
User related: /api/user/…
Custom fields: /api/customField/…
Logging: /api/log/…
Cart: /api/cart/…
Item: /api/item/…
Requests: /api/request/…
Tags: /api/tag/...

To get an API key, first login to the web interface, then hit the /api/user/apiKey/get endpoint with a browser. Then, put the API key as "x-key" field in GET/POST header.

Login and OAuth:

Login with credentials
POST /api/auth/login
Sample JSON:

```
{
  "username": "test1",
  "password": "12345"
}
```

Login with OAuth
GET /api/auth/oauth

Create an Admin account (Can only be used once)
/api/auth/createAdmin
Sample JSON:

```
{
  "username": "admin",
```

```
  "name": "Super User",
  "password": "admin"
}
```

User:

Get current user info, if logged in (require login)
GET /api/user

E.g. not logged in:
{"status":403,"message":"Not Authorized"}
Logged in:

```
{
  "username": "admin",
  "name": "",
  "netId": "",
  "email": "",
  "status": "admin",
  "active": true,
  "apiKey":
"0c0817ecf9799c7dffc288b9ae92bab80c9696dd25969c145df0729daa7b50781370926e31767
96bc4f187e37bc808d505f2fdedf9671bf43d7db67f66adbfd6"
}
```

Get the API key for the current user (require login)
GET /api/user/apiKey/get

Revoke the API key for the current user (require login)
GET /api/user/apiKey/revoke

Get the user info whose primary key in the database is :id (require admin or manager)
GET /api/user/:id/info

View a list of all users (require admin or manager)
GET /api/user/show

Add a user (require admin or manager)
POST /api/user/add

Sample JSON:

```
{
  "username": "test",
  "name": "Test User",

  "password": "12345",
  "netId": "",
  "email": "test@example.com",
  "status": "user",
  "active": true
}
```

Update a user (require admin or manager)
POST /api/user/update
Sample JSON:

```
{
  "_id": < _id of the user, can get from /api/user/show, or leave empty to update the current
 user>,
  "username": "test1",
  "name": "Test User",

  "password": "12345",
  "netId": "",
  "email": "test@example.com",
  "status": "user",
  "active": true
}
```

Delete a user (require admin or manager)
POST /api/user/del
Sample JSON:

```
{
  "_id": < _id of the user, can get from /api/user/show>
}
```

Custom Fields:
Show all custom fields:
GET /api/customField/show

Add:
POST /api/customField/add
JSON fields:
name: required
```

type: string, required, one of "short", "long", "int", "float"
private: boolean, not required, default false
req: boolean, not required, default false
default: same type as "type", not required, default ""

**Update:**
POST /api/customField/update
JSON fields:
name: required
type: not required
private: not required
req: not required
default: not required

**Delete:**
POST /api/customField/del
JSON fields:
name: required

**Logging**
**Show a list of logs**
GET /api/log/show?limit=20

* If limit param is supplied, the # of log entries will be subject to that param. Otherwise, the default limit is 20. The list is always sorted by descending time.

**Filter a list of logs**
GET or POST /api/log/filter

Sample JSON:
```
{
  "limit": 40,
  "init_user": "Mike", (optional, regex match on name field of users)
  "rec_user": "Jason", (optional, regex match on name field of users)
  "item": "computer chip", (optional, regex match on name field of items)
}
```

**Cart:**
**All following APIs will return a JSON of all current items in the cart**

**Show all items in the cart**
GET /api/cart/show

**Add an item to the cart**
POST /api/cart/add

Required JSON fields:
item: the _id of an item (can get from /api/item/show)
quantity: integer quantity

**Delete an item from the cart**
/api/cart/delete

Required JSON fields:
item: the _id of an item (can get from /api/item/show)

**Update the quantity of an item to the cart**
/api/cart/update

Update works the same way as add. If the item is not currently in the cart, it will be added. If the item is already in the cart, its quantity will be updated.

Required JSON fields:
item: the _id of an item (can get from /api/item/show)
quantity: integer quantity

# Item
**Show a list of items (require login)**
GET /api/item/show?limit=20

* If limit param is supplied, the # of items will be subject to that param. Otherwise, the default limit is 20.

**Add an item (require admin or manager)**
POST /api/item/add

Sample JSON:

```
{
    "name": "i7 Chip",
    "fields": {  // a list of custom fields
      "location": "room1",
```

```
      "note": "note"
    },
    "tags": ["chip","latest"], // an array of tags
    "description": "latest intel chip",
    "model": "i7 7700k",
    "quantity": 100
  }
```

Delete an item (require admin or manager)
POST /api/item/del
Sample JSON:

```
  {
    "name": "i7 Chip",
  }
```
OR
```
  {
    "_id": "58ab62ceca32af82ffdef58b", (can get from /api/item/show)
  }
```

Update an item (require admin or manager)
POST /api/item/update

Only name field is required.

Sample JSON:

```
  {
    "name": "i7 Chip",
    "fields": {  // a list of custom fields
      "location": "room1",
      "note": "note"
    },
    "tags": ["chip","latest"], // an array of tags
    "description": "latest intel chip",
    "model": "i7 7700k",
    "quantity": 100
  }
```

Requests
Show a list of requests, ordered by descending date (Require login)

GET /api/request/show

Add a request from cart (Require login)
POST /api/request/add

Sample JSON:
```
{
    "reason": "Add a request"
}
```

Close a request (Require login)
POST /api/request/close

Sample JSON:
```
{
    "_id": <The _id of request, can get from /api/request/show>
}
```

Delete a request (Require admin/manager)
POST /api/request/del

Sample JSON:
```
{
    "_id": <The _id of request, can get from /api/request/show>
}
```

Update a request (Require admin/manager for approval)
POST /api/request/update

Sample JSON:
```
{
    "_id": <The _id of request, can get from /api/request/show>
   "status": "approved",
   "note": "Empty"
}
```


Tags
Show a list of all tags. No paging here, since there aren't that many tags. (Req login)
GET /api/tag/show

Add a tag. (Require admin/manager)

POST /api/tag/add

Sample JSON:
```
{
    "name": "cheap",
}
```

Delete a tag. (Require admin/manager)
POST /api/tag/del

Sample JSON:
```
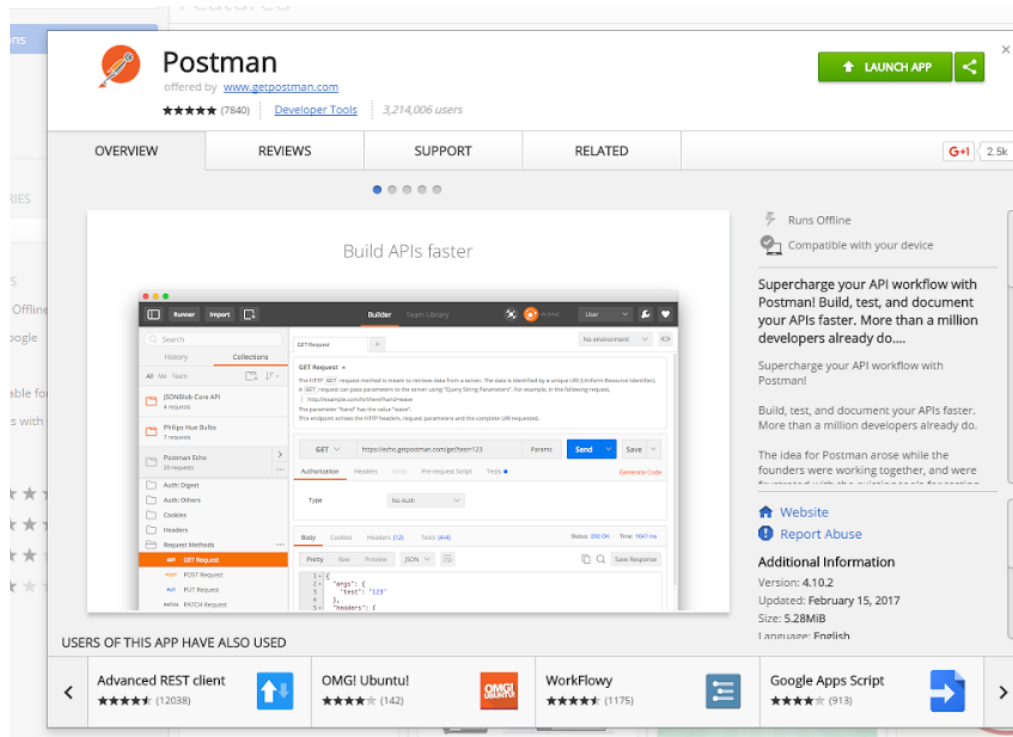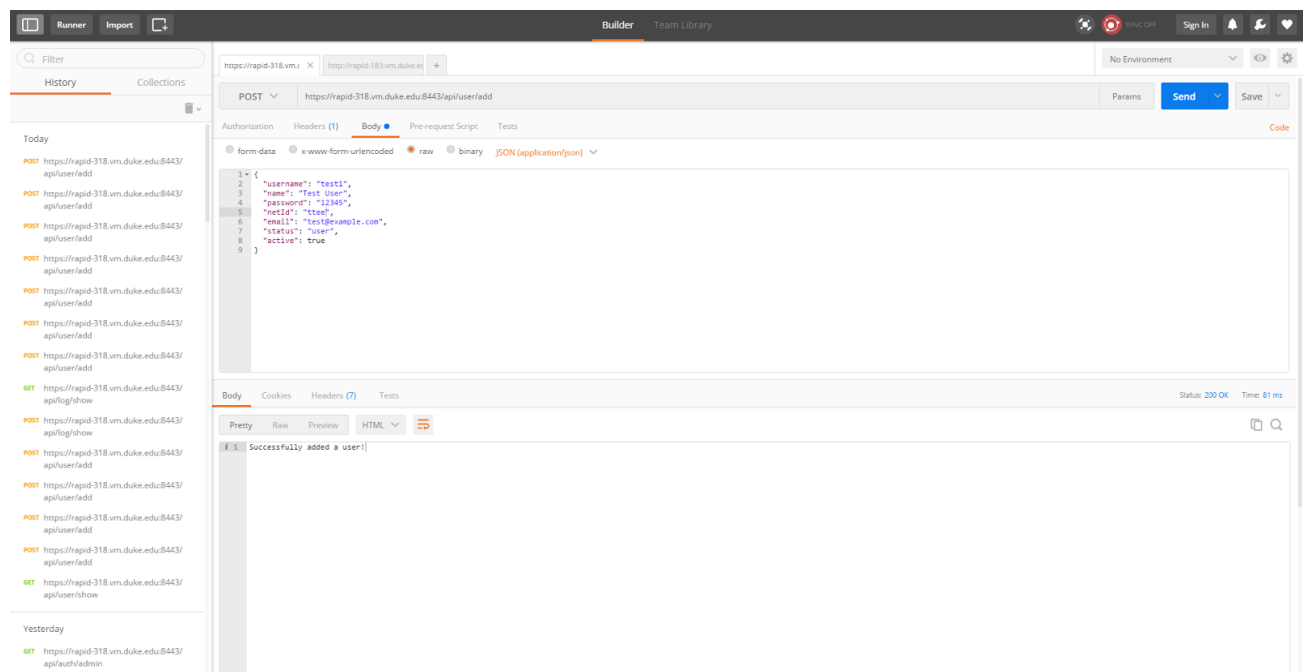{
    "name": "free",
}
```

<u>Testing</u>

To test our API, the developers recommend using Postman, which can be found free on the Chrome Store:

Once the application is launched, the interface is as follows:

You can select the type of request you want to make to the API (GET or POST) from the dropdown shown next to the URL; if it is a POST, you have the option of modifying the body of the request. Above is an example use of the Postman interface to test an API method.