

WeCycle API Documentation

Version 1.0

DEVELOPERS: Chinedu Amadi-Ndukwe, Bilal Samiullah & Francesca Okorosobo

API Documentation Version History

Version	Date	Author	Description
1.0	16-Feb-2015	Francesca Okorosobo	Initial draft
1.1	26-Mar-2015	Francesca Okorosobo	Final updated version for WeCycle v1

Table of Contents

API Documentation Version History	1
Introduction	3
To use the API, do the following:	3
Models Defined.....	3
Users	3
Items	3
Resources	4
Routes.....	5
USERS	5
Create a new user	5
Get all users	6
Get user by ID	7
Get user items	8
Get user items 2	9
Update User	10
Delete User	11
ITEMS	12
Create a new item	12
Get all items	13
Get item by ID	14
Update Item	15
Delete Item	16
Glossary	17
Status Codes.....	17

Introduction

Welcome to the API documentation for WeCycle, this document describes the RESTful resources provided by WeCycle.

WeCycle is a platform that encourages university students to donate (recycle) items they no longer need and pass them onto other students who may need the items. Students can upload items they no longer need onto the WeCycle platform and other students can contact them to collect the item. WeCycle solves the problem of waste and encourages students to become more environmentally friendly.

All responses are formatted in JSON.

To use the API, do the following:

- use the POSTMAN client app for Google Chrome
- Use <https://wecycle-api.herokuapp.com> or if running on local machine use: http://localhost:5000/**

Models Defined

Users

Attribute	Type
_id (Auto-Generated)	Long
name	String
email	String
password	String
phone_number	String

Items

Attribute	Type
_id (Auto-Generated)	Long
name	String
description	String
URI	String
_creator (references user's email)	String (type:Schema.Types.ObjectId, ref: 'User')

Resources

Method	URI	Action
GET	/users	Returns all users
GET	/users/{user_id}	Returns a single user
GET	/users/{user_id}/items	Returns all the items posted by a specific user
GET	/users/{user_id}/items/{item_id}	Gets an item posted by a specific user
POST	/users	Adds a new user
PUT	/users/{user_id}	Updates an existing user
DELETE	/users/{user_id}	Deletes a user
GET	/items	Returns all items
GET	/items/{item_id}	Returns an single item
POST	/items	Adds a new item
PUT	/items/{item_id}	Updates an existing item
DELETE	/items/{item_id}	Deletes an existing item

Routes

USERS

Create a new user

Creates a new user

REQUEST

Method	URI
POST	/users

RESPONSE

Status	Response
201	<pre>{ "Message": "User successfully created and added." }</pre> <p><i>Example user created in JSON format</i></p> <pre>{ "_id": "5501d0c6645f9cb101f22264", "name": "Mandy Davies ", "email": "mdavies@gmail.com" "password": "D4v13s!", "phone_number": 07965345321 }</pre>
204	<pre>{ "Error": "No data received, please enter some data." }</pre>
400	<pre>{ "Error": " Unable to add new user, please check the data you entered is correct." }</pre>
500	<pre>{ "Error": "Something went wrong. Please try again later." }</pre>

Get all users

Returns all users in the system

REQUEST

Method	URI
GET	/users

RESPONSE

Status	Response
200	Returns a list of users <i>Example response in JSON format</i> <pre>{ "_id": "5501d0c6645f9cb101f22264", "name": "Mandy Davies ", "email": "mdavies@gmail.com" "password": "D4v13s!", "phone_number": "07965345321" }, { "_id": "5501d071645f9cb101f22263", "email": "sammyj@hotmail.com", "password": "P4r1s!", "name": "Sam Jones ", "phone_number": "07986754678" } }</pre>
404	{"Error": "There was an error getting the list of users, please enter the correct URI."}
500	{"Error": "Something went wrong. Please try again later."}

Get user by ID

Returns a single user

REQUEST

Method	URI
GET	/users/{user_id}

Parameters	Values	Example
user_id	Long	_id: 5501d0c6645f9cb101f22264

RESPONSE

Status	Response
200	Returns a single user <i>Example response in JSON format</i> <pre>{ "_id": "5501d0c6645f9cb101f22264", "name": "Mandy Davies ", "email": "mdavies@gmail.com" "password": "D4v13s!", "phone_number": 07965345321 }</pre>
404	{"Error": " There was an error getting the user, please provide the correct user ID."}
400	{"Error": "Please provide the user ID."}
500	{"Error": "Something went wrong. Please try again later."}

Get user items

Returns all items posted by a specific user

REQUEST

Method	URI
GET	/users/{user_id}/items

Parameters	Values	Example
user_id	Long	_id: 5501d0c6645f9cb101f22264

RESPONSE

Status	Response
200	Returns all the items posted by the specified user <i>Example response in JSON format</i> <pre>{ "_id": "5501d5e4645f9cb101f22265", "name": "frying pan", "description": "Tefal frying pan, no longer need, good condition", "_creator": "5501d0c6645f9cb101f22264", "url": ["http://www.tesco.com/direct/tefal-adventure-32cm-frying-pan/672-9879.prd"] }, { "_id": "5501d7575434c7d2018b7277", "name": "clothes drying rack ", "description": "no longer need, good condition, wilko brand", "_creator": "5501d0c6645f9cb101f22264", "url": ["http://www.wilko.com/indoor+outdoor-airers/wilko-airer-deluxe-14m/invnt/0107006"] }</pre>
404	{"Error": " There was an error getting the user's items, please provide the correct user ID."}
400	{"Error": "Please provide the user ID."}
401	{"Error": "Unauthorised access."}
500	{"Error": "Something went wrong. Please try again later."}

Example request URI(s)

<https://wecycle-api.herokuapp.com/users/5501d0c6645f9cb101f22264/items>

Get user items 2

Returns one item posted by a specific user

REQUEST

Method	URI
GET	/users/{user_id}/items/{item_id}

Parameters	Values	Example
user_id	Long	_id: 5501d0c6645f9cb101f22264
item_id	Long	_id:5501d5e4645f9cb101f22265

RESPONSE

Status	Response
200	Returns one item posted by the specified user <i>Example response in JSON format</i> <pre>{ "_id": "5501d5e4645f9cb101f22265", "name": "frying pan", "description": "Tefal frying pan, no longer need, good condition", "_creator": "5501d0c6645f9cb101f22264", "url": ["http://www.tesco.com/direct/tefal-adventure-32cm-frying-pan/672-9879.prd"] }</pre>
404	{"Error": "Please provide the correct item ID or correct user ID."}
400	{"Error": "Please provide the item ID or user ID."}
401	{"Error": "Unauthorised access."}
500	{"Error": "Something went wrong. Please try again later."}

Update User

Updates an existing user

REQUEST

Method	URI
PUT	/users/{user_id}

Parameters	Values	Example
user_id	Long	_id: 5501d0c6645f9cb101f22264

RESPONSE

Status	Response
200	<p>Updates an existing user</p> <p><i>Example response in JSON format</i></p> <pre>{ "_id": "5501d0c6645f9cb101f22264", "name": "Mandy Davies ", "email": "mdavies@gmail.com" "password": "D4v13s!", "phone_number": 07965345321 }</pre>
404	{"Error": "Please provide the correct user ID."}
400	{"Error": "Please provide the user ID."}
401	{"Error": "Unauthorised access."}
500	{"Error": "Something went wrong. Please try again later."}

Delete User

Deletes an existing user

REQUEST

Method	URI
DELETE	/users/{user_id}

Parameters	Values	Example
user_id	Long	_id: 5501d0c6645f9cb101f22264

RESPONSE

Status	Response
200	{"Message": "User successfully deleted."}
404	{"Error": "Please provide the correct user ID."}
400	{"Error": "Please provide the user ID."}
401	{"Error": "Unauthorised access."}
500	{"Error": "Something went wrong. Please try again later."}

ITEMS

Create a new item

Creates a new item

REQUEST

Method	URI
POST	/items

To post an item by a certain user, use the following attributes: name, description, url and user_email.

RESPONSE

Status	Response
201	<pre>{ "Message": "Item successfully created and added." }</pre> <p><i>Example response in JSON format</i></p> <pre>{ "_id": "5501d7575434c7d2018b7277", "name": "clothes drying rack ", "description": "no longer need, good condition, wilko brand ", "_creator": "5501d0c6645f9cb101f22264", "URL": ["http://www.wilko.com/indoor+outdoor-airers/wilko-airer-deluxe-14m/invnt/0107006"] }</pre>
204	<pre>{ "Error": "No data received, please enter some data." }</pre>
404	<pre>{ "Error": " Something went wrong with trying to create the item, please check the data you entered is correct." }</pre>
500	<pre>{ "Error": "Something went wrong. Please try again later." }</pre>

Get all items

Finds all the items in the system

REQUEST

Method	URI
GET	/items

RESPONSE

Status	Response
200	<p>Returns a list of all the items</p> <p><i>Example response in JSON format:</i></p> <pre>{ "_id": "5501d5e4645f9cb101f22265", "name": "frying pan", "description": "Tefal frying pan, no longer need, good condition", "_creator": "5501d0c6645f9cb101f22264" "URI": ["http://www.tesco.com/direct/tefal-adventure-32cm-frying-pan/672-9879.prd"] }, { "_id": "5501d7575434c7d2018b7277", "name": "clothes drying rack ", "description": "no longer need, good condition, wilko brand ", "_creator": "5501d0c6645f9cb101f22264", "__v": 0, "URI": ["http://www.wilko.com/indoor+outdoor-airers/wilko-airer-deluxe-14m/inv/0107006"] }</pre>
404	<pre>{"Error": " There was an error getting the list of items, please enter the correct URI."}</pre>
500	<pre>{"Error": "Something went wrong. Please try again later."}</pre>

Get item by ID

Returns a single item

REQUEST

Method	URI
GET	/items/{item_id}

Parameters	Values	Example
user_id	Long	_id:5501d7575434c7d2018b7277

RESPONSE

Status	Response
200	Returns a single item <i>Example response in JSON format</i> <pre>{ "_id": "5501d7575434c7d2018b7277", "name": "clothes drying rack ", "description": "no longer need, good condition, wilko brand ", "_creator": "5501d0c6645f9cb101f22264", "URI": ["http://www.wilko.com/indoor+outdoor-airers/wilko-airer-deluxe-14m/invnt/0107006"] }</pre>
404	{"Error": " There was an error getting the item, please provide the correct item ID."}
400	{"Error": "Please provide the item ID."}
500	{"Error": "Something went wrong. Please try again later."}

Update Item

Updates an existing item

REQUEST

Method	URI
PUT	/items/{item_id}

Parameters	Values	Example
user_id	Long	_id:5501d7575434c7d2018b7277

RESPONSE

Status	Response
200	<p>Updates existing item</p> <p><i>Example response in JSON format</i></p> <pre>{ "_id": "5501d7575434c7d2018b7277", "name": "clothes drying rack ", "description": "no longer need, good condition, wilko brand ", "_creator": "5501d0c6645f9cb101f22264", "URI": ["http://www.wilko.com/indoor+outdoor-airers/wilko-airer-deluxe-14m/inv/0107006"] }</pre>
404	{"Error": "Please provide the correct item ID."}
400	{"Error": "Please provide the item ID."}
401	{"Error": "Unauthorised access."}
500	{"Error": "Something went wrong. Please try again later."}

Delete Item

Deletes an existing item

REQUEST

Method	URI
DELETE	/items/{item_id}

Parameters	Values
ID (Auto-Generated)	Long

RESPONSE

Status	Response
200	{"Message": "Item successfully deleted."}
404	{"Error": "Please provide the correct item ID."}
400	{"Error": "Please provide the item ID."}
401	{"Error": "Unauthorised access."}
500	{"Error": "Something went wrong. Please try again later."}

Glossary

Status Codes

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client's part

5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
400	Bad request
401	Authentication failure
404	Resource not found
500	Internal Server Error