



webstoreapi1.0OAS3

API for webstore application as part of the mobile applications course.

Tobi Tögel - Website
Send email to Tobi Tögel

Servers

https://mobileappapiwebstore.herokuapp.com/

Authorize

default

GET

/users/{userId}

Get User Info by User ID

Retrieve the information of the user with the matching user ID.

Parameters

Try it out

Name	Description
userId * required integer (path)	Id of an existing user. <div>userId - Id of an existing user.</div>

Responses

Code	Description	Links
------	-------------	-------

Code	Description	Links
200	User Found	No links
<div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Examples</div><div>Get User Alice Smith</div></div>		
<div><div>Example Value</div><div>Schema</div><div><pre>{ "id": 142, "firstName": "Alice", "lastName": "Smith", "email": "alice.smith@gmail.com", "dateOfBirth": "1997-10-31", "emailVerified": true, "signUpDate": "2019-08-24" }</pre></div></div>		
404	User Not Found	No links

POST

/register

Create New User

Create a new user.

Parameters

Try it out

No parameters

Request body

application/json

Post the necessary fields for the API to create a new user.

Examples:

Create User Bob Fellow

Example Value

Schema

```
{
  "username": "fellowbob",
  "password": "bobfellow",
  "firstName": "Bob",
  "lastName": "Fellow",
  "email": "bob.fellow@gmail.com",
  "dateOfBirth": "1996-08-24"
}
```

Responses		
Code	Description	Links
200	User Created	No links
<div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div></div> <div><div>Examples</div><div>New User Bob Fellow</div></div> <div><div>Example Value</div><div>Schema</div><div><pre>{ "id": 12, "firstName": "Bob", "lastName": "Fellow", "email": "bob.fellow@gmail.com", "dateOfBirth": "1996-08-24", "emailVerified": false, "createDate": "2020-11-18" }</pre></div></div>		
400	Missing Required Information	No links
409	Email Already Taken	No links

POST/login

Post the user login information to server to check if user is authorized to login.

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	OK	No links

Code	Description	Links
401	Unauthorized	No links

GET

/items

Your GET endpoint

Get information of items that are being sold on the webstore

Parameters

Try it out

Name	Description
category	Search and list according to category given by user
string	
(query)	<div>category - Search and list according to category given by user</div>
location	Search and list according to location given by user
string	
(query)	<div>location - Search and list according to location given by user</div>
date of posting	Search and list according to Date of posting given by user
string	
(query)	<div>date of posting - Search and list according to Date of posting ;</div>

Responses

Code	Description	Links
------	-------------	-------

Code	Description	Links
200	<div>OK</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": "string", "title": "string", "description": "string", "category": "string", "images": "string", "price": 0, "date": "string", "delivery type": "string", "seller username": "string", "seller contact": "string", "location": "string"}</pre></div>	No links
400	<div>Bad Request</div>	No links

POST

/items

Create new item to sell in the webstore.

Parameters

Try it out

No parameters

Request body

application/json

Example Value Schema

```
{  "id": "string",  "title": "string",  "description": "string",  "category": "string",  "images": "string",  "price": 0,  "date": "string",  "delivery type": "string",  "seller username": "string",
```

```
"seller contact": "string",
"location": "string"
}
```

Responses

Code	Description	Links
201	Created	No links
400	Bad Request (missing information?)	No links
401	Unauthorized	No links

DELETE

/items/{itemId}

Delete item from Webstore

Parameters

Try it out

Name	Description
itemId * required string (path)	Id of an existing item. <div>itemId - Id of an existing item.</div>

Responses

Code	Description	Links
200	OK	No links
401	Unauthorized	No links

Code	Description	Links
404	Not Found	No links

PUT

/items/{itemId}

Edit existing item in Webstore

Parameters

Try it out

Name	Description
itemId * required string (path)	Id of an existing item. <div>itemId - Id of an existing item.</div>

Request body

application/json

Example Value

Schema

```
{
  "id": "string",
  "title": "string",
  "description": "string",
  "category": "string",
  "images": "string",
  "price": 0,
  "date": "string",
  "delivery type": "string",
  "seller username": "string",
  "seller contact": "string",
  "location": "string"
}
```

Responses

Code	Description	Links
200	OK	No links

Code	Description	Links
400	Bad Request	No links
401	Unauthorized	No links
404	Not Found	No links

Media type

application/json

Example Value

Schema

```
{
  "id": "string",
  "title": "string",
  "description": "string",
  "category": "string",
  "images": "string",
  "price": 0,
  "date": "string",
  "delivery type": "string",
  "seller username": "string",
  "seller contact": "string",
  "location": "string"
}
```

GET	/items/{itemId}	Get single Item
Get single item		
Parameters		Try it out
Name	Description	
itemId * required	Id of an existing item.	
string (path)	<div>itemId - Id of an existing item.</div>	
Responses		

Code	Description	Links
200	<div><div>OK</div><div>Media type<div>application/json</div><div>Controls Accept header.</div></div><div><div>Example Value</div><div>Schema</div><div><pre>{ "id": "string", "title": "string", "description": "string", "category": "string", "images": "string", "price": 0, "date": "string", "delivery type": "string", "seller username": "string", "seller contact": "string", "location": "string"}</pre></div></div></div>	No links

Schemas



```
User  {  
  id*           integer  
                Unique identifier for the given user.  
  
  firstName*    string  
  lastName*     string  
  email*        string($email)  
  dateOfBirth   string($date)  
                example: 1997-10-31  
  emailVerified* boolean  
                Set to true if the user's email has been verified.  
  
  createDate    string($date)  
                The date that the user was created.  
  
  x-examples    {"Alice Smith":  
                {"id":142,"firstName":"Alice","lastName":"Smith","email":"alice.smith@gmail.com",'  
                10-31","emailVerified":true,"signUpDate":"2019-08-24"}}  
}
```



```
item  {  
  id*           string  
  title*        string  
  description    string  
  category*     string  
  images*       string  
  price*        number  
  date*         string  
  delivery type* string  
  seller username* string  
  seller contact* string  
  location*     string  
}
```

