**Program: application.py**

* Author: Perry Brandiezs
* Date: May 1, 2019
* Last updated: May 3, 2019

This program demonstrates CRUD operations using an Item Catalog.

* Create: Ability to create an airplane item
* Read: Ability to read an inventory list showing category name, item name, item description. Ability to show item detail, login required.
* Update: Ability to edit item detail, login required.
* Delete: Ability to delete an item, login required and must be item creator.

This program demonstrates OAuth2 authentication and authorization using a third party provider.

* Login / Logout using Facebook is provided, link can be found at the top-right of the main screen.
* Login required to display item detail, update an item, or delete an item.
* Must also be the item creator to delete.

This program demonstrates API endpoints.

* Display all items
* Display specific item detail
* Display all users
* Display all categories

**Please see the README.md at vagrant/catalog/README.md, for additional detail and Test User information.**

# Expected Output:

## Initial DB setup:

vagrant@vagrant:/vagrant/catalog$ python models.py

## Initial Data Loading:

vagrant@vagrant:/vagrant/catalog$ python create\_planes.py

User Perry Brandiezs added

User Mickey Mouse added

Category Gulfstream added

Category Crop Duster added

Category Falcon added

Category Embraer added

Category Aerospatiale/BAC added

Plane G650ER added

Plane G550 added

Plane G650ER added

Plane Piper Cub added

Plane 10X added

Plane Phenom 100 added

Plane Concorde added

## vagrant@vagrant:/vagrant/catalog$

## Webserver start:

vagrant@vagrant:/vagrant/catalog$ python application.py

\* Serving Flask app "application" (lazy loading)

\* Environment: production

WARNING: Do not use the development server in a production environment.

Use a production WSGI server instead.

\* Debug mode: on

\* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)

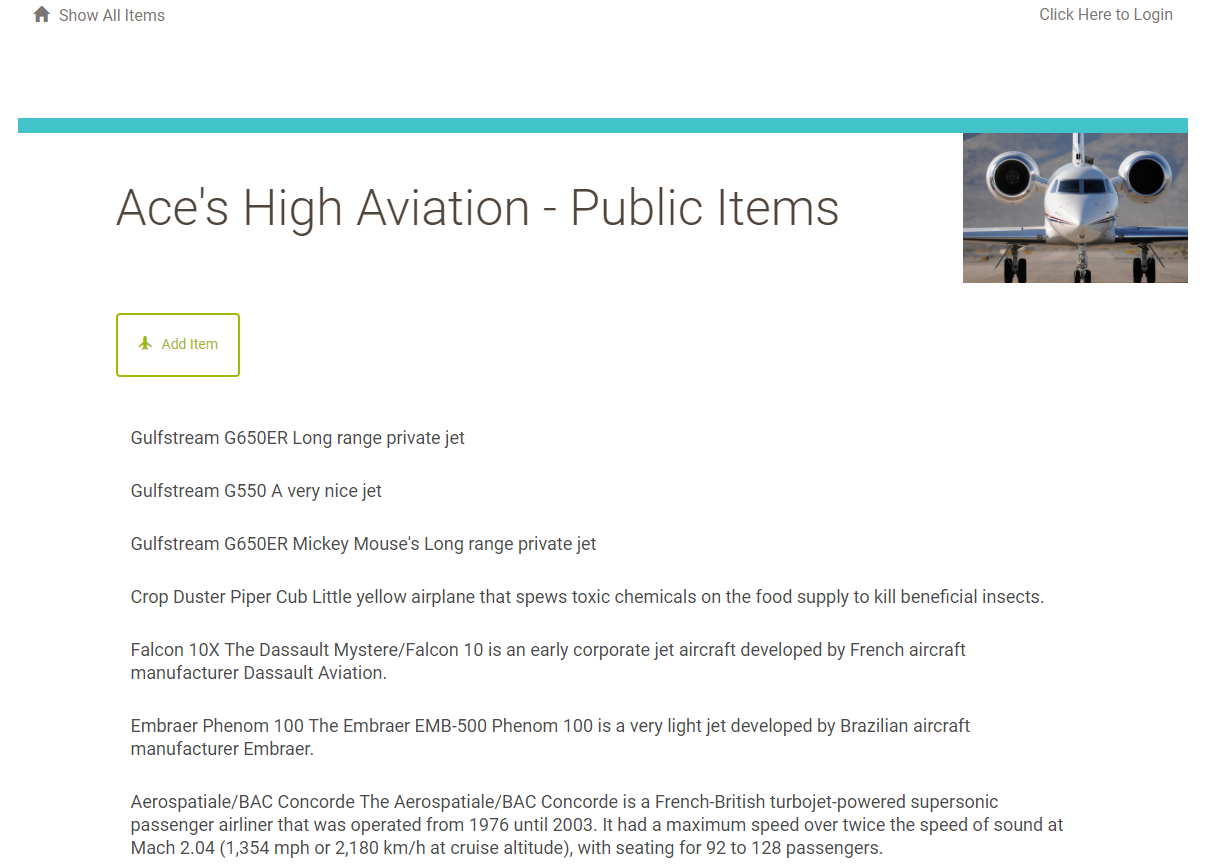
\* Restarting with stat

\* Debugger is active!

\* Debugger PIN: 325-904-188

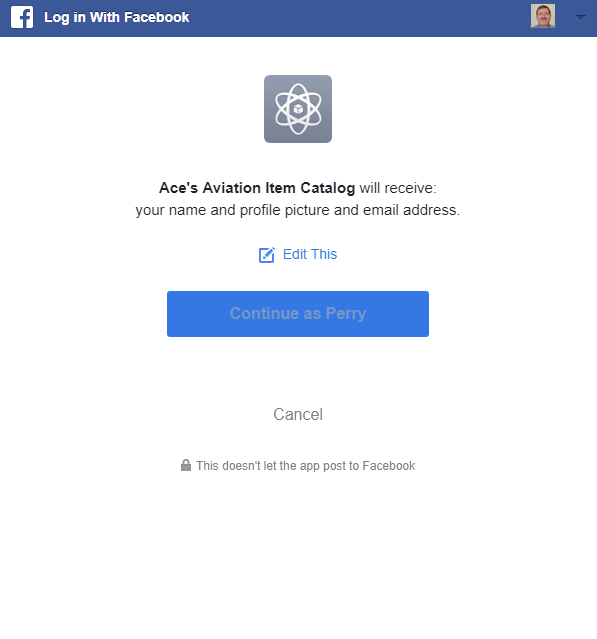
## Main screen

<http://localhost:8000/>



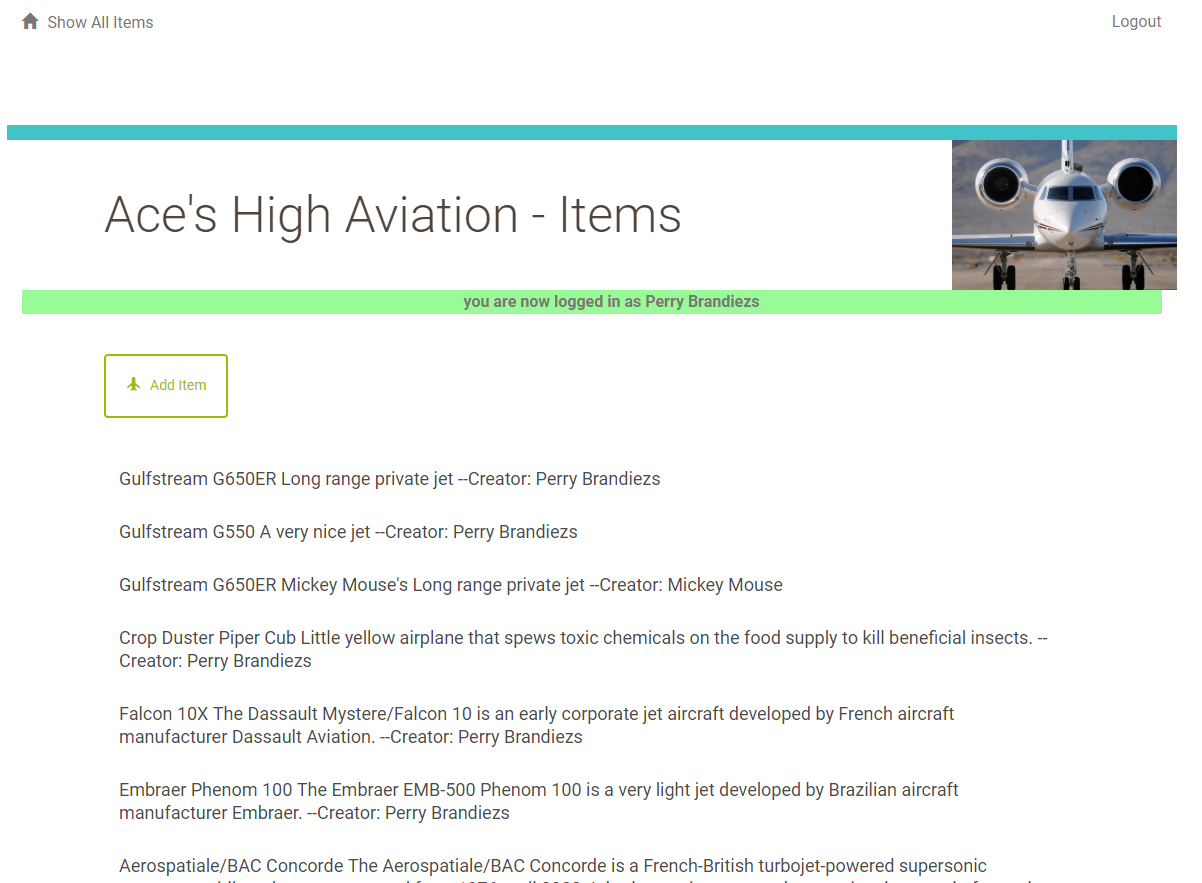
## Login screen:

<http://localhost:8000/login>



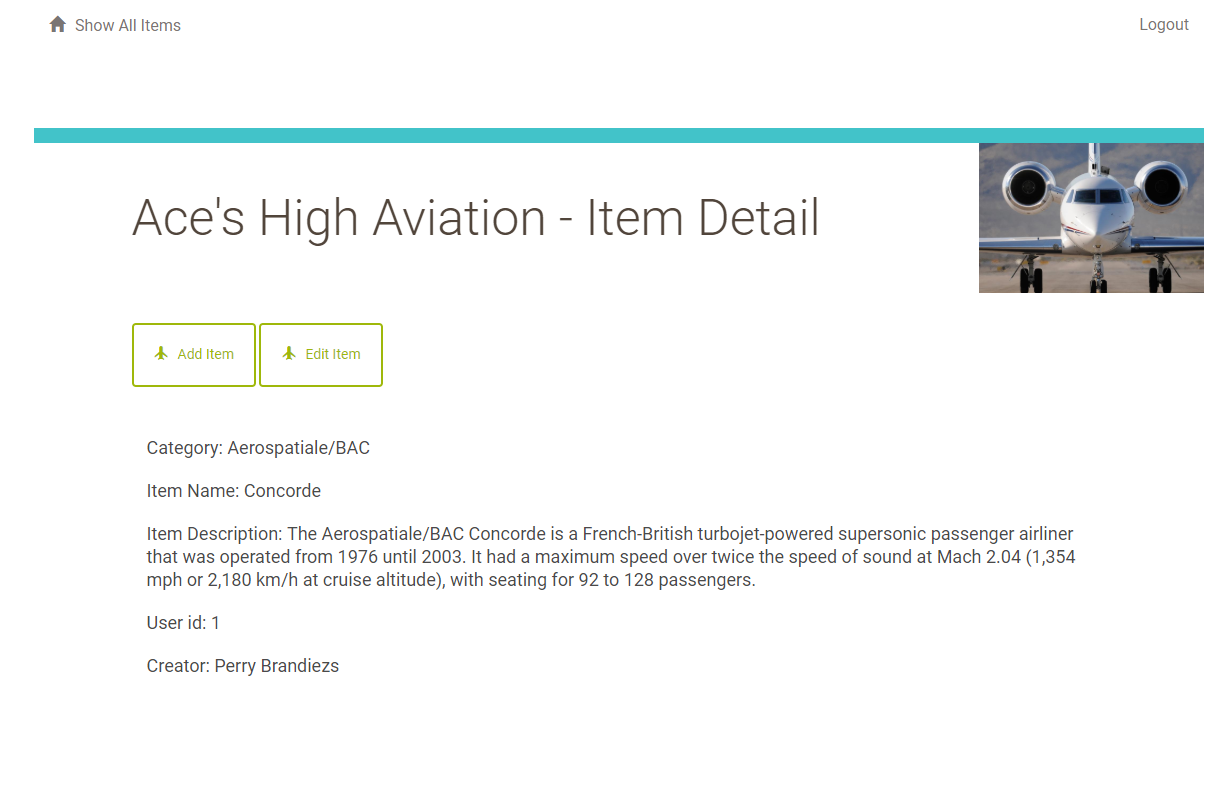
## Main screen – logged in

<http://localhost:8000/>



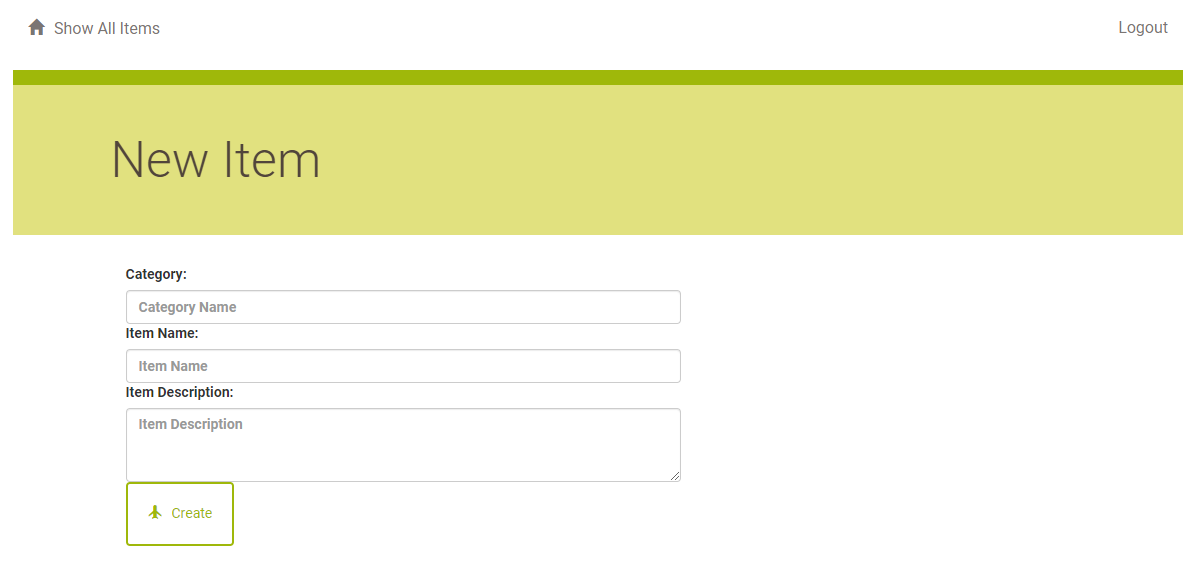
## Item Detail (click on a specific item)

<http://localhost/item/7/>



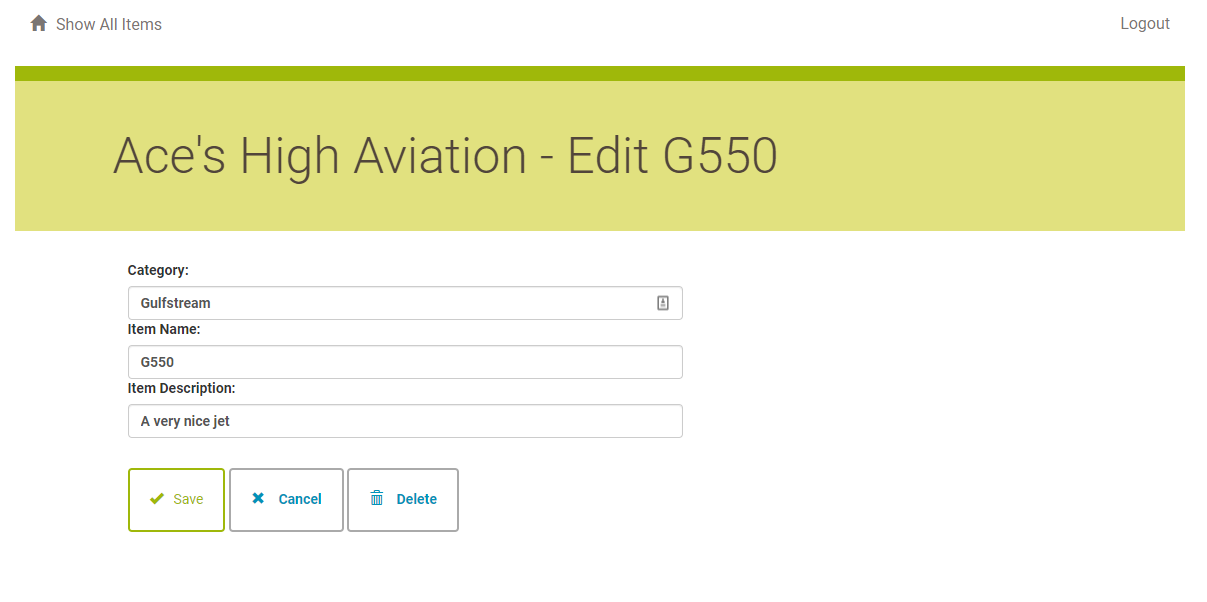
## Add item:

<http://localhost:8000/item/new/>



## Edit item:

<http://localhost:8000/item/2/edit/>

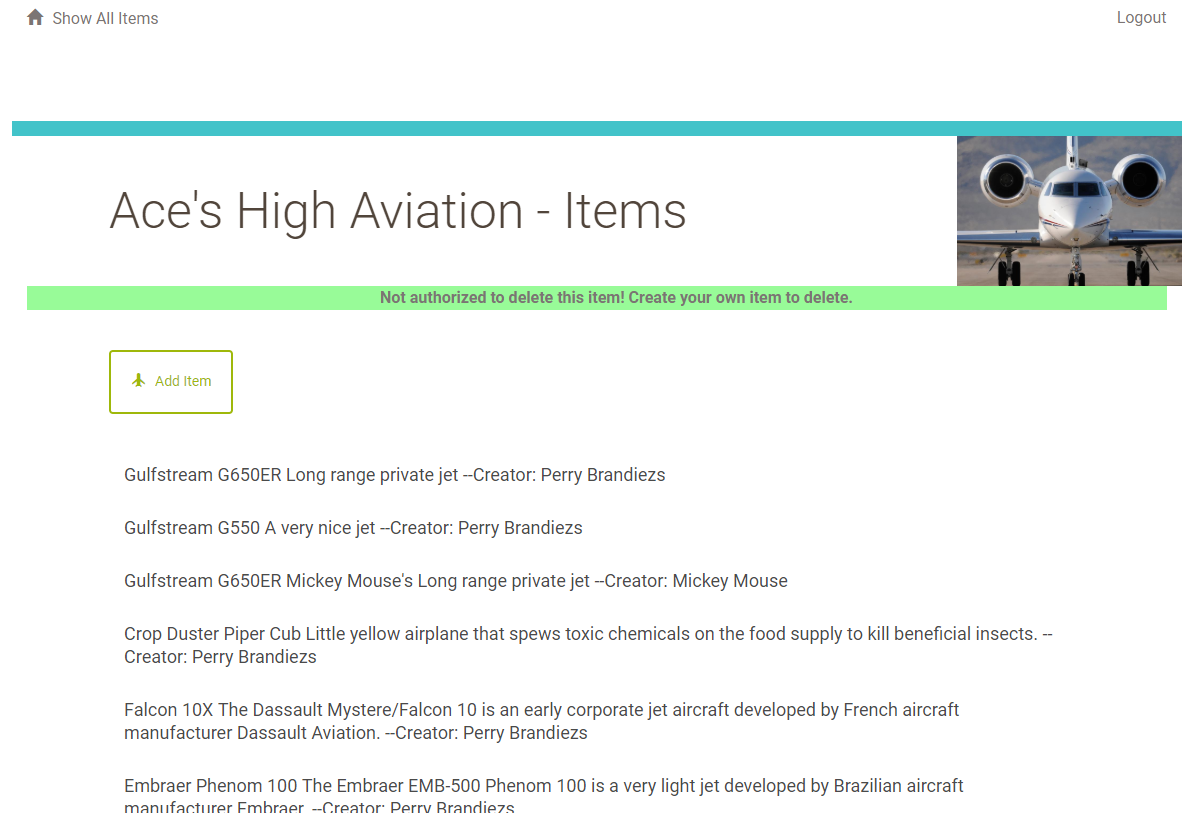


## Delete item:

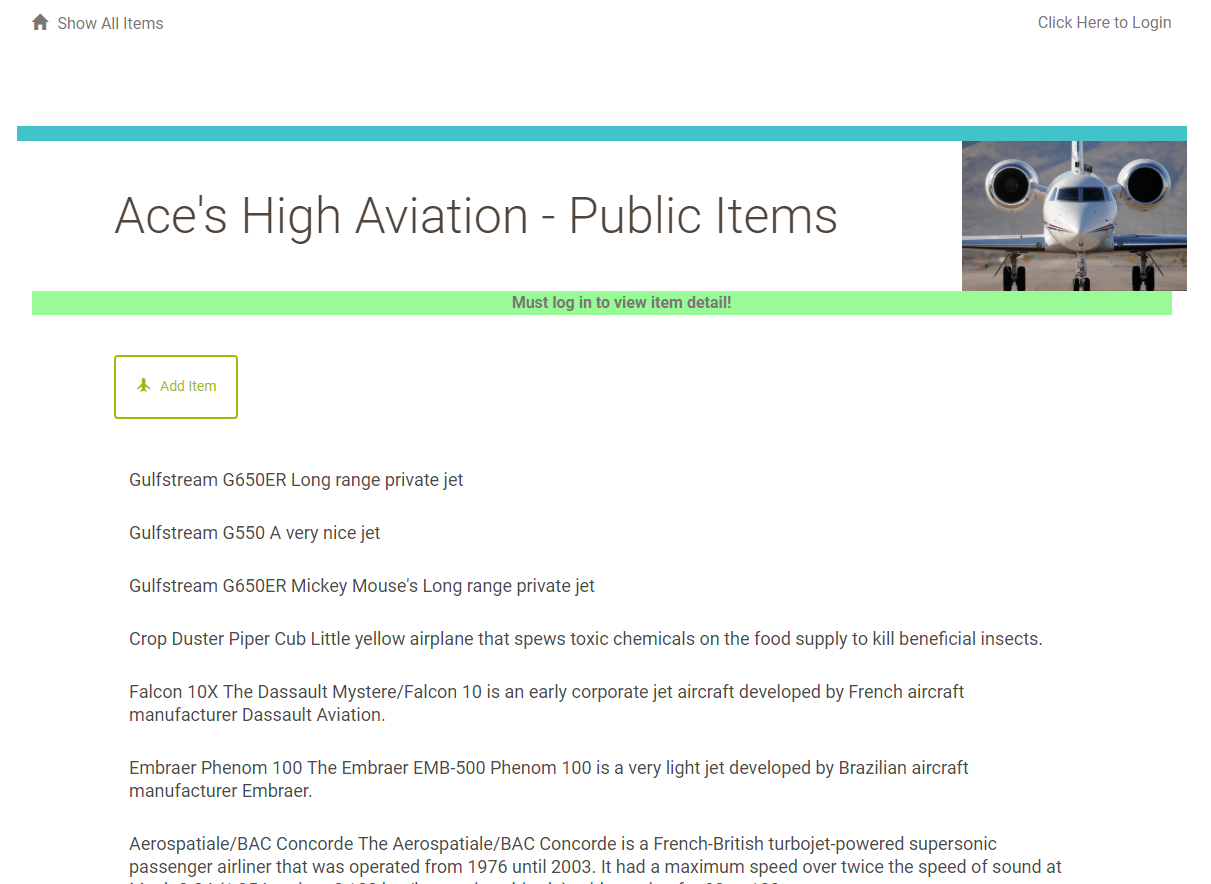
<http://localhost:8000/item/2/delete/>



## Attempt to delete an item, which you have not created:



## Attempt to view item detail when not logged in:



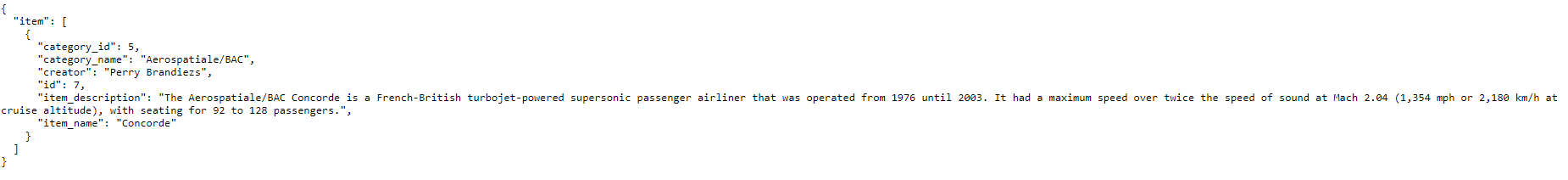
## API endpoint – all items

<http://localhost:8000/items/JSON>



## API endpoint – specific item:

<http://localhost:8000/item/7/JSON>



## API endpoint – all users:



## API endpoint – all categories:

