

1. Download the Swagger app from the Git
URL: <https://github.com/Srirambala2k/SwaggerTestApp>

2. Install node modules using "npm install" command

3. Populate the sample data into the Mongoose DB

Execute the following js files

1. node /populate/populatebike.js

2. node /populate/populatedress.js

3. node /populate/populatehome.js

4. Run the application: node index.js

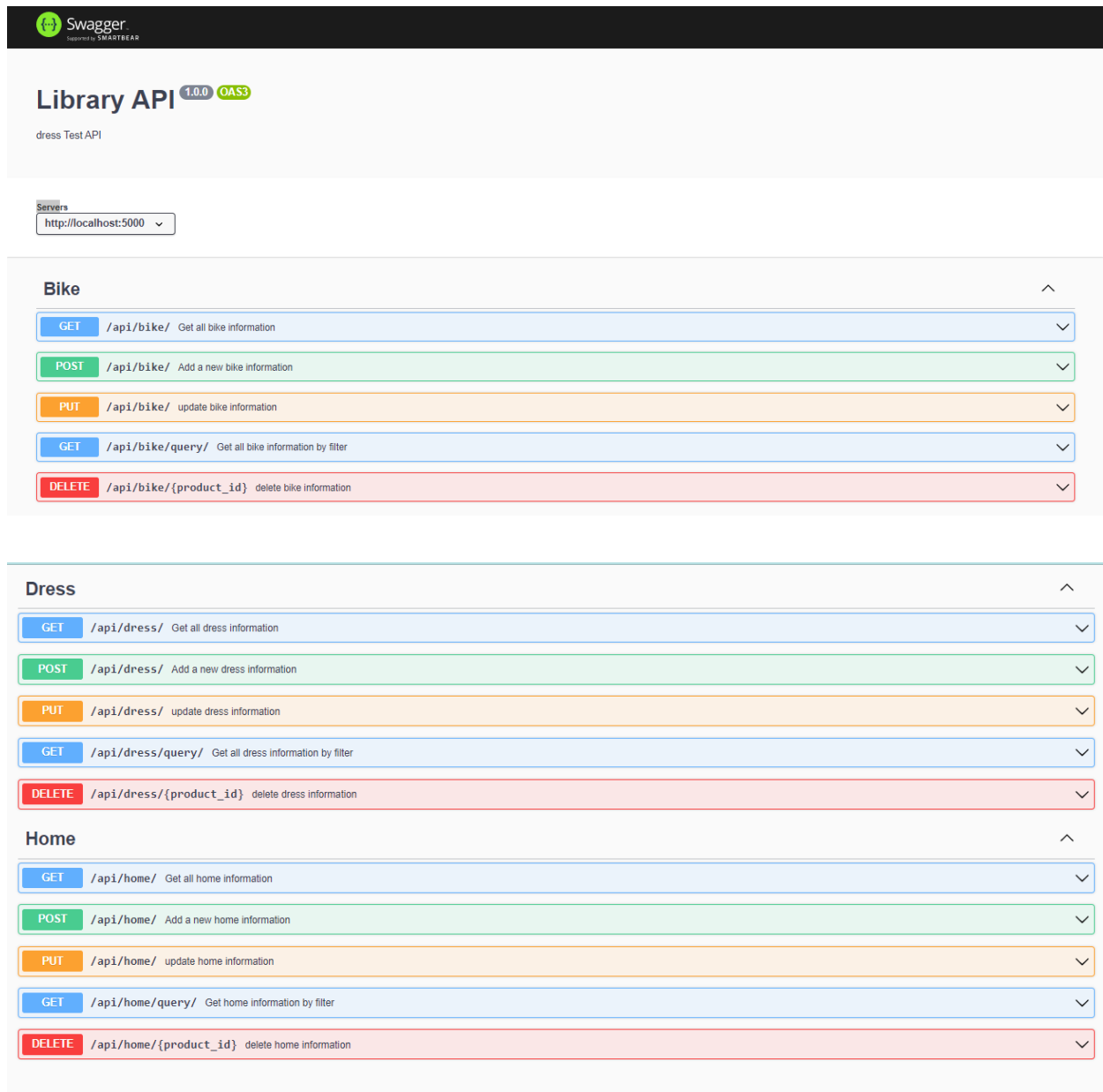
Schemas

```
bike ▾ {  
  product_id      integer  
  name            string  
  brand           string  
  cc              integer  
  topspeed        integer  
  price           integer  
}
```

```
dress ▾ {  
  product_id      integer  
  name            string  
  size            string  
  colour          string  
  sex             string  
  price           integer  
  stocks          integer  
}
```

```
home ▾ {  
  product_id      integer  
  ownerName       string  
  sqfeet          integer  
  beds            integer  
  bath            integer  
  price           integer  
  place           string  
  contactNumber   integer  
}
```

APIs



The image shows a Swagger UI interface for a 'Library API'. At the top, the Swagger logo is visible. Below it, the title 'Library API' is displayed with version '1.0.0' and 'OAS3' tags. A 'Servers' dropdown menu is set to 'http://localhost:5000'. The API is organized into three sections: 'Bike', 'Dress', and 'Home'. Each section contains a list of endpoints with their respective HTTP methods (GET, POST, PUT, DELETE) and descriptions. For example, the 'Bike' section includes endpoints for getting all bike information, adding a new bike, updating bike information, getting bike information by filter, and deleting bike information by product_id. The 'Dress' and 'Home' sections follow a similar pattern with their respective endpoints.

Library API 1.0.0 OAS3
dress Test API

Servers
http://localhost:5000

Bike

- GET /api/bike/ Get all bike information
- POST /api/bike/ Add a new bike information
- PUT /api/bike/ update bike information
- GET /api/bike/query/ Get all bike information by filter
- DELETE /api/bike/{product_id} delete bike information

Dress

- GET /api/dress/ Get all dress information
- POST /api/dress/ Add a new dress information
- PUT /api/dress/ update dress information
- GET /api/dress/query/ Get all dress information by filter
- DELETE /api/dress/{product_id} delete dress information

Home

- GET /api/home/ Get all home information
- POST /api/home/ Add a new home information
- PUT /api/home/ update home information
- GET /api/home/query/ Get home information by filter
- DELETE /api/home/{product_id} delete home information

References:

Swagger Editor: <https://editor.swagger.io/>

Swagger Components: <https://swagger.io/docs/specification/components/>

Videos: NodeJS Swagger API Documentation Tutorial Using Swagger JSDoc
<https://www.youtube.com/watch?v=S8kmHtQeflo>

