

Testing and request log of email backend system

Get all products

The screenshot shows the Postman application interface. In the left sidebar, there is a collection named 'email' containing two items: 'get all product' (GET) and 'add a product' (POST). The 'get all product' item is selected. The main panel displays a GET request to 'http://127.0.0.1:8000/api/products'. The 'Body' tab is selected, showing a JSON response with two product objects:

```
1  [
2   {
3     "product_id": "prod1",
4     "name": "some product",
5     "description": "hdasoidhasoisa",
6     "price": 20,
7     "product_category": "catA",
8     "image1": "/images/prod1_img.png",
9     "image2": "",
10    "image3": "",
11    "discount": 20,
12    "shop_id_fk": "shop1",
13    "date_added": "0000-00-00"
14  },
15  {
16    "product_id": "prod2",
17    "name": "some product",
18    "description": "hdasoidhasoisa",
19    "nrice": 20
20  }
```

The status bar at the bottom indicates: Status: 200 OK Time: 1976 ms Size: 1.67 KB Save Response.

Add a product using post

The screenshot shows the Postman application interface. In the left sidebar, there is a collection named 'email' containing two items: 'get all product' (GET) and 'add a product' (POST). The 'add a product' item is selected. The main panel displays a POST request to 'http://127.0.0.1:8000/api/products?shop_id_fk=shop1&name=Mobile&description=Duel sim&price=5000&product_category=Electronics&discount=0'. The 'Body' tab is selected, showing a form-data key 'image1' with value 'RX8-mini_ID-364x364.jpg'. The status bar at the bottom indicates: Status: 201 Created Time: 750 ms Size: 518 B Save Response.

The Body content is as follows:

```
1  {
2   "product_id": "prod7",
3   "shop_id_fk": "shop1",
4   "name": "Mobile",
5   "description": "Duel sim",
6   "price": "5000",
7   "product_category": "Electronics",
8   "image1": "/images/prod7_img.jpg",
9   "image2": "",
10  "image3": "",
11  "discount": "0"
```

The status bar at the bottom indicates: 28°C Rain showers 09:54 AM 16/08/2021.

Get all categories

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email' containing three items: 'get all products', 'add a product', and 'get all categories'. The 'get all categories' item is selected. The main workspace shows a GET request to 'http://127.0.0.1:8000/api/categories'. The response status is 200 OK, and the response body is a JSON array of five categories:

```
1  [
2    {
3  "category_id": "cat1",
4  "name": "Bags"
5  },
6  {
7  "category_id": "cat2",
8  "name": "Beauty Shop"
9  },
10 {
11 "category_id": "cat3",
12 "name": "Clothing"
13 },
14 {
15 "category_id": "cat4",
16 "name": "Electronics"
17 },
18 {
19 "category_id": "cat5",
```

Add a category using post

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email' containing four items: 'get all products', 'add a product', 'get all categories', and 'GET add a category'. The 'GET add a category' item is selected. The main workspace shows a POST request to 'http://127.0.0.1:8000/api/categories?name=Medicine'. The response status is 201 Created, and the response body is a JSON object representing the new category:

```
1 {
2   "category_id": "cat9",
3   "name": "Medicine"
4 }
```

Get all subcategories

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email' which contains several API endpoints. One endpoint is selected: 'GET / get all subcategories'. The request method is 'GET' and the URL is 'http://127.0.0.1:8000/api/subcategories'. The 'Params' tab is active. In the 'Body' tab, the response is displayed in JSON format:

```
1 {
2   "sub_cat_id": "sub1",
3   "category_id_fk": "cat3",
4   "name": "Man's Clothing"
5 },
6 {
7   "sub_cat_id": "sub10",
8   "category_id_fk": "cat4",
9   "name": "TV"
10 },
11 {
12   "sub_cat_id": "sub11",
13   "category_id_fk": "cat8",
14   "name": "Man's Watch"
15 },
16 {
17   "sub_cat_id": "sub12",
18   "category_id_fk": "cat8",
```

The status bar at the bottom shows: Status: 200 OK Time: 329 ms Size: 2.46 KB Save Response.

Add a subcategory

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email' which contains several API endpoints. One endpoint is selected: 'POST / add a subcategory'. The request method is 'POST' and the URL is 'http://127.0.0.1:8000/api/subcategories?category_id_fk=cat9&name=First Aid'. The 'Params' tab is active. In the 'Body' tab, the request body is defined with two parameters:

KEY	VALUE	DESCRIPTION
category_id_fk	cat9	
name	First Aid	

The status bar at the bottom shows: Status: 201 Created Time: 191 ms Size: 377 B Save Response.

Get all childcategories

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, APIs, Environments, Mock Servers, and Monitors. Under 'Collections', there's a folder named 'email' containing several API endpoints. One endpoint is selected: 'GET get all childcategories'. The main panel shows the request details: method 'GET', URL 'http://127.0.0.1:8000/api/childcategories', and a table for 'Query Params'. Below the table, the 'Body' tab is selected, showing a JSON response with 16 items. The response is a list of objects, each containing 'child_cat_id', 'name', and 'sub_cat_id_fk'. The desktop taskbar at the bottom shows various application icons.

```
1 [ { 2 "child_cat_id": "child1", 3 "name": "T-shirt", 4 "sub_cat_id_fk": "sub1" 5 }, 6 { 7 "child_cat_id": "child10", 8 "name": "Sandals", 9 "sub_cat_id_fk": "sub3" 10 }, 11 { 12 "child_cat_id": "child11", 13 "name": "Sandals", 14 "sub_cat_id_fk": "sub4" 15 }, 16 ]
```

Add a childcategory

The screenshot shows the Postman application interface. The left sidebar is identical to the previous screenshot. A new endpoint is selected: 'POST add a childcategory'. The main panel shows the request details: method 'POST', URL 'http://127.0.0.1:8000/api/childcategories?sub_cat_id_fk=sub34&name=Antiseptic', and a table for 'Query Params'. In the 'Body' tab, there's a JSON body with two fields: 'sub_cat_id_fk' (set to 'sub34') and 'name' (set to 'Antiseptic'). The response status is '201 Created'. The desktop taskbar at the bottom shows various application icons.

```
1 { 2 "sub_cat_id_fk": "sub34", 3 "name": "Antiseptic", 4 "child_cat_id": "child20" 5 }
```

Get all shops

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing a collection named 'email'. Under 'email', there are several API endpoints listed under the 'GET' method. One endpoint, 'get all shops', is selected. The main panel shows a 'GET' request to 'http://127.0.0.1:8000/api/shops'. The 'Body' tab is selected, showing a JSON response with three shop objects:

```
1  [
2    {
3      "shop_id": "shop1",
4      "shop_name": "Walton",
5      "category_id_fk": "cat4",
6      "logo": "/logos/shop1_img.jpg"
7    },
8    {
9      "shop_id": "shop2",
10     "shop_name": "Apple",
11     "category_id_fk": "cat4",
12     "logo": "/logos/shop2_img.jpg"
13   },
14   {
15     "shop_id": "shop3",
16     "shop_name": "pharmacy.com.bd",
17     "category_id_fk": "cat9",
18     "logo": "/logos/shop3_img.jpg"
19   }
]
```

The status bar at the bottom indicates '28°C Rain showers' and the date '18/08/2021'.

Add a shop

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing a collection named 'email'. Under 'email', there are several API endpoints listed under the 'POST' method. One endpoint, 'add a shop', is selected. The main panel shows a 'POST' request to 'http://127.0.0.1:8000/api/shops?category_id_fk=cat9&shop_name=Pharmacy.com.bd'. The 'Body' tab is selected, showing a JSON body with a single key 'logo':

```
1  {
2    "logo": "0e1f04cff3a8a053d89f4f7899f38e1.jpg"
3  }
```

The status bar at the bottom indicates '28°C Rain showers' and the date '18/08/2021'.

Get all shopparents

The screenshot shows the Postman interface with a collection named "email". A GET request is selected with the URL `http://127.0.0.1:8000/api/shopparents`. The response status is 200 OK, and the JSON data returned is:

```

    [
        {
            "email": "iftekhar@gmail.com",
            "dob": "1997-03-02T00:00:00.000000Z",
            "address": "dhaka",
            "NID": 2131233,
            "first_name": "Iftekhar",
            "last_name": "Hyder",
            "number": 1213123,
            "password": "$2y$10$B0ia/126yJiuUev.FgnG7NuTvMb37jt6ubPpgRko54m7AfY.EvKC2",
            "type": "staff",
            "shop_id_fk": "shop1"
        },
        {
            "email": "mazbaul7@gmail.com",
            "dob": "1978-01-01T00:00:00.000000Z",
            "address": "dinajpur",
            "NID": 123456,
            "first_name": "Mazbaul",
            "last_name": "Hoque",
            "number": 123456789,
            "password": "$2y$10$mgb0YAQoXepZ3cW4dAPQh.Mq.K16umZr54nCBEdyI41ft.T8fRwz2",
            "type": "owner"
        }
    ]
  
```

Add a shopparent

The screenshot shows the Postman interface with a collection named "email". A POST request is selected with the URL `http://127.0.0.1:8000/api/shopparents?shop_id_fk=shop3&email=mazbaul7@gmail.com&NID=123456&dob=23/09/1998&address=dinajpur&first_name=Mazbaul&last_name=Hoque&number=0123456789&password=meeem7`. The response status is 201 Created, and the JSON data returned is:

```

    {
        "shop_id_fk": "shop3",
        "email": "mazbaul7@gmail.com",
        "NID": 123456,
        "first_name": "Mazbaul",
        "last_name": "Hoque",
        "number": 123456789,
        "password": "$2y$10$mgb0YAQoXepZ3cW4dAPQh.Mq.K16umZr54nCBEdyI41ft.T8fRwz2",
        "type": "owner"
    }
  
```

Get all users

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing a collection named 'email'. Under 'email', there are several API endpoints listed under the 'GET' method. One endpoint, 'get all users', is selected and highlighted with a grey background. The main workspace shows a 'GET' request to 'http://127.0.0.1:8000/api/users'. The 'Params' tab is active, showing a single query parameter 'key' with the value 'Value'. Below the request, the response status is 'Status: 200 OK' with a response time of '172 ms' and a size of '562 B'. The response body is displayed in JSON format:

```
1
2
3   "email": "iftekhar@gmail.com",
4   "name": "Iftekhar Hyes",
5   "dob": "1997-03-04T00:00:00.000000Z",
6   "address": "Dhaka",
7   "phone": 12321321
8
9
10  "email": "mazbaul@gmail.com",
11  "name": "Mazbaul Hoque",
12  "dob": "1979-01-01T00:00:00.000000Z",
13  "address": "Dinajpur",
14  "phone": 123456789
15
16
```

Add a user

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'My Workspace' containing a collection named 'email'. Under 'email', there are several API endpoints listed under the 'POST' method. One endpoint, 'add a user', is selected and highlighted with a grey background. The main workspace shows a 'POST' request to 'http://127.0.0.1:8000/api/users'. The 'Params' tab is active, showing a query parameter 'email' with the value 'mazbaul@gmail.com', and another query parameter 'name' with the value 'Mazbaul Hoque'. Other parameters like 'dob', 'address', 'phone', and 'password' are also listed with their respective values. Below the request, the response status is 'Status: 201 Created' with a response time of '308 ms' and a size of '439 B'. The response body is displayed in JSON format:

```
1
2
3   "email": "mazbaul@gmail.com",
4   "name": "Mazbaul Hoque",
5   "dob": "1979-01-01T00:00:00.000000Z",
6   "address": "Dinajpur",
7   "phone": 123456789
```

get an image of a product

The screenshot shows the Postman interface with a collection named "email". A GET request is made to `http://127.0.0.1:8000/api/images/prod1/1`. The response details indicate a status of 200 OK, a duration of 196 ms, and a size of 66.32 KB. The response body is a binary file, likely a PNG image, which is displayed as a large block of hexagonal patterns.

get logo of a shop

The screenshot shows the Postman interface with a collection named "email". A GET request is made to `http://127.0.0.1:8000/api/logos/shop3`. The response details indicate a status of 200 OK, a duration of 139 ms, and a size of 6.4 KB. The response body is a binary file, likely a logo image, which is displayed as a small, low-resolution graphic.

get ranged shop list

The screenshot shows the Postman application interface. The left sidebar lists various API endpoints under 'My Workspace'. The main panel displays a successful API call for 'email / get ranged shop list'.

Request Details:

- Method: GET
- URL: <http://127.0.0.1:8000/api/shops/select/2/3>
- Headers: (6)
- Body: (10)
- Pre-request Script: (1)
- Tests: (1)
- Settings: (1)

Query Params:

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body (Pretty JSON):

```
1: [
2:   {
3:     "shop_id": "shop2",
4:     "shop_name": "Apple",
5:     "category_id_fk": "cat4",
6:     "logo": "/logos/shop2_img.jpg"
7:   },
8:   {
9:     "shop_id": "shop3",
10:    "shop_name": "Pharmacy.com.bd",
11:    "category_id_fk": "cat9",
12:    "logo": "/logos/shop3_img.jpg"
13:  }
14: ]
```

Status: 200 OK | Time: 172 ms | Size: 510 B | Save Response

get ranged product list

The screenshot shows the Postman application interface. The left sidebar lists various API endpoints under 'My Workspace'. The main panel displays a successful API call for 'email / get ranged product list'.

Request Details:

- Method: GET
- URL: <http://127.0.0.1:8000/api/products/select/3/7>
- Headers: (6)
- Body: (10)
- Pre-request Script: (1)
- Tests: (1)
- Settings: (1)

Query Params:

KEY	VALUE	DESCRIPTION	
Body	Cookies	Headers (10)	Test Results

Body (Pretty JSON):

```
1: [
2:   {
3:     "product_id": "prod3",
4:     "name": "some product",
5:     "description": "hdasidhasoisa",
6:     "price": 28,
7:     "product_category": "cat4",
8:     "image": "/images/prod3_img.png",
9:     "image2": "",
10:    "image3": "",
11:    "discount": 20,
12:    "shop_id_fk": "shop2",
13:    "date_added": "2021-06-17"
14:  },
15:  {
16:    "product_id": "prod4",
17:    "name": "some product",
18:    "description": "hdasidhasoisa",
19:    "price": 28,
20:    "product_category": "cat4",
21:    "image1": "/images/prod4_img.png",
22:  }
23: ]
```

Status: 200 OK | Time: 167 ms | Size: 1.43 KB | Save Response

get verification of user

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email / get verification of user' containing several requests. One specific request is highlighted: 'GET http://127.0.0.1:8000/api/users/mazbaul@gmail.com/meem?'. The 'Params' tab is selected, showing a single query parameter 'Key' with the value 'Value'. The 'Body' tab is also visible. At the bottom, the response status is shown as 'Status: 200 OK'.

get products under a shop

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email / get products under a shop' containing several requests. One specific request is highlighted: 'GET http://127.0.0.1:8000/api/shops/products/Walton'. The 'Params' tab is selected, showing a single query parameter 'Key' with the value 'Value'. The 'Body' tab is also visible. The response body is displayed in JSON format, showing a list of products under the shop 'Walton'. The JSON output is as follows:

```
1: [
2:   {
3:     "product_id": "prod1",
4:     "name": "some product",
5:     "description": "hdsoildhasoisa",
6:     "price": 28,
7:     "product_category": "cat4",
8:     "image1": "/images/prod1_img.png",
9:     "image2": "",
10:    "image3": "",
11:    "discount": 20,
12:    "shop_id_fk": "shop1",
13:    "date_added": "0000-00-00"
14:  },
15:  {
16:    "product_id": "prod2",
17:    "name": "some product",
18:    "description": "hdsoildhasoisa",
19:    "price": 28,
20:    "product_category": "cat4",
21:    "image1": "/images/prod2_img.png",
22:    "image2": "",
23:    "image3": ""
24:  }
]
```

get a shop info

The screenshot shows the Postman interface with a successful API call. The URL is `http://127.0.0.1:8000/api/shops/get/Pharmacy.com.bd`. The response status is 200 OK, time 175 ms, size 412 B. The response body is:

```
1 "shop_id": "shop3",
2 "shop_name": "Pharmacy.com.bd",
3 "category_id_fk": "cat9",
4 "logo": "/logos/shop3_img.jpg"
```

get some random products

The screenshot shows the Postman interface with a successful API call. The URL is `http://127.0.0.1:8000/api/products/random/3`. The response status is 200 OK, time 182 ms, size 997 B. The response body is:

```
2 [
3   {
4     "product_id": "prod1",
5     "name": "some product",
6     "description": "hdasoidhasoisa",
7     "price": 20,
8     "product_category": "cat4",
9     "image1": "/images/prod1_img.png",
10    "image2": "",
11    "image3": "",
12    "discount": 20,
13    "shop_id_fk": "shop1",
14    "date_added": "0000-00-00"
15  },
16  {
17    "product_id": "prod5",
18    "name": "some product",
19    "description": "hdasoidhasoisa",
20    "price": 20,
21    "product_category": "cat4",
22    "image1": "/images/prod5_img.png",
23    "image2": ""
```

get some recently added products

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email /' containing several API endpoints. One endpoint is highlighted: 'GET /get some recently added products'. The main panel displays the details for this request:

- Method:** GET
- URL:** http://127.0.0.1:8000/api/products/recent/3
- Params:** None
- Authorization:** None
- Headers:** (6)
- Body:** None
- Pre-request Script:** None
- Tests:** None
- Settings:** None

The 'Query Params' section is collapsed. The 'Body' section is also collapsed. The 'Pretty' tab is selected in the preview area. The response status is 200 OK, time 198 ms, size 1005 B. The response body is a JSON array of two product objects:

```
[{"product_id": "prod5", "name": "some product", "description": "hdasoihdasoisa", "price": 20, "product_category": "cat4", "image1": "/images/prod5_img.png", "image2": "", "image3": "", "discount": 20, "shop_id_fk": "shop2", "date_added": "2021-08-31"}, {"product_id": "prod4", "name": "some product", "description": "hdasoihdasoisa", "price": 20, "product_category": "cat4", "image1": "/images/prod4_img.png", "image2": "", "image3": ""}]
```

get product attributes

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'email /' containing several API endpoints. One endpoint is highlighted: 'GET /get product attributes'. The main panel displays the details for this request:

- Method:** GET
- URL:** http://127.0.0.1:8000/api/productattributes
- Params:** None
- Authorization:** None
- Headers:** (6)
- Body:** None
- Pre-request Script:** None
- Tests:** None
- Settings:** None

The 'Query Params' section is collapsed. The 'Body' section is also collapsed. The 'Pretty' tab is selected in the preview area. The response status is 200 OK, time 1049 ms, size 641 B. The response body is a JSON array of three product attribute objects:

```
[{"product_id_fk": "prod1", "color_id_fk": "color1", "size_id_fk": "size1", "child_cat_id_fk": "child1", "quantity": 20}, {"product_id_fk": "prod1", "color_id_fk": "color2", "size_id_fk": "size2", "child_cat_id_fk": "child1", "quantity": 20}, {"product_id_fk": "prod2", "color_id_fk": "color4", "size_id_fk": "size4", "child_cat_id_fk": "child1", "quantity": 20}]
```

add product attributes

The screenshot shows the Postman application interface. In the left sidebar, there's a collection named 'email / add product attributes'. The main area displays a POST request to 'http://127.0.0.1:8000/api/productattributes?product_id_fk=prod2&color_id_fk=color4&size_id_fk=size1&child_cat_id_fk=child1&quantity=30'. The 'Params' tab is selected, showing query parameters: product_id_fk (prod2), color_id_fk (color4), size_id_fk (size1), child_cat_id_fk (child1), and quantity (30). The response status is 201 Created, and the JSON response body is:

```
1 "product_id_fk": "prod2",
2 "color_id_fk": "color4",
3 "size_id_fk": "size1",
4 "child_cat_id_fk": "child1",
5 "quantity": 30
```

get color with value

The screenshot shows the Postman application interface. In the left sidebar, there's a collection named 'email / get color with value'. The main area displays a GET request to 'http://127.0.0.1:8000/api/colors'. The 'Params' tab is selected, showing a single parameter 'Key' with the value 'Value'. The response status is 200 OK, and the JSON response body is:

```
1 [
2   {
3     "color_id": "color1",
4     "value": "c93212"
5   },
6   {
7     "color_id": "color2",
8     "value": "16B444"
9   },
10  {
11    "color_id": "color3",
12    "value": "B01FBF"
13  },
14  {
15    "color_id": "color4",
16    "value": "000000"
17  },
18  {
19    "color_id": "color5",
20
21 }
```

add color with value

The screenshot shows the Postman application interface. On the left, the sidebar lists various API endpoints under 'My Workspace'. In the main area, a POST request is being configured for the URL `http://127.0.0.1:8000/api/colors?value=%23B81FBF`. The 'Params' tab is selected, showing a single parameter named 'value' with the value '#B81FBF'. The 'Body' tab shows a JSON response with the key 'color_id' and value 'color6'. The status bar at the bottom indicates a 201 Created response.

get size with value

The screenshot shows the Postman application interface. On the left, the sidebar lists various API endpoints under 'My Workspace'. In the main area, a GET request is being configured for the URL `http://127.0.0.1:8000/api/sizes`. The 'Params' tab is selected, showing a single parameter named 'value' with the value 'large'. The 'Body' tab shows a JSON response with three size entries: 'size1' (value 'large'), 'size2' (value 'small'), and 'size3' (value 'extra large'). The status bar at the bottom indicates a 200 OK response.

add size with value

The screenshot shows the Postman interface with a collection named "email / add size wth value". A POST request is being made to `http://127.0.0.1:8000/api/sizes?value=extra small`. In the "Query Params" section, there is one entry: "value" with the value "extra small". The response status is 201 Created, and the JSON response body is:

```
1 "size_id": "size5",
2 "value": "extra small"
```

get a product attributes

The screenshot shows the Postman interface with a collection named "email / get a product attributes". A GET request is being made to `http://127.0.0.1:8000/api/getattributes/prod`. In the "Query Params" section, there are no entries. The response status is 200 OK, and the JSON response body is:

```
1 {
2   "product": {
3     "product_id": "prod2",
4     "name": "some product",
5     "description": "hdasdoidhasoisa",
6     "price": 28,
7     "product_category": "cat4",
8     "image1": "/images/prod2_img.png",
9     "image2": "",
10    "image3": "",
11    "discount": 28,
12    "shop_id_fk": "shop1",
13    "date_added": "2021-06-04"
14  },
15  "colors": [
16    {
17      "color_id": "color4",
18      "value": "#000000"
19    }
20  ],
21  "sizes": [
22  ]}
```