Capstone Project Part-3 (API Testing using Bruno)

TESTING YOUR REFUNDS API USING BRUNO

===

1. Create a New Collection

===

1. Click **New Collection**

2. Name it: **Refunds API**

3. Choose a folder to store it

Bruno creates a local folder for your API tests.

===

2. Create a New Request

===

1. Click **New Request**

2. Set:

   o **Request Name**: Get All Refunds

   o **Method**: GET

   o **URL**:

http://localhost:8080/refunds

======================================

3. Add Bearer Token Authentication

======================================

Your API requires:

Authorization: Bearer my-secret-token

In Bruno:

1. Click **Headers** tab

2. Add a new header:

Key	Value
Authorization	Bearer my-secret-token

That's it!

======================================

4. Test the Request

======================================

1. Click **Send**

2. Expected output:

[]

(because no refunds yet)

If you **forget the token**, Bruno will return:

401 Unauthorized

Which matches your ApiKeyFilter.

==

5. Create More Requests

==

Below are all API calls with correct Bruno configurations.

5.1 Create Refund — POST /refunds

URL:

```
http://localhost:8080/refunds
```

Method:

POST

Headers:

Key	Value
Authorization	Bearer my-secret-token
Content-Type	application/json

Body (JSON):

```
{
  "amount": 500,
  "currency": "USD",
  "reason": "Product Return"
}
```

Expected Response:

```
{

  "id": "generated-uuid",

  "amount": 500,

  "currency": "USD",

  "reason": "Product Return",

  "status": "PENDING"

}
```

5.2 Get Refund By ID — GET /refunds/{id}

Paste ID from previous response.

URL:

```
http://localhost:8080/refunds/<REFUND_ID>
```

Headers:

Key	Value
Authorization	Bearer my-secret-token

5.3 Check Status — POST /refunds/status

URL:

http://localhost:8080/refunds/status

Method:

POST

Body:

```
{
  "refundId": "<REFUND_ID>"
}
```

Headers:

Key	Value
Authorization	Bearer my-secret-token
Content-Type	application/json

==

7. Create a Bruno Environment (Optional but recommended)

==

Helps avoid repeating token & base URL.

Step 7.1 — Create Environment

1. Click **Environments → New Environment**

2. Name it: Local

Add variables:

Variable	Value
baseUrl	http://localhost:8080
token	my-secret-token

Step 7.2 — Use Variables in Requests

Replace:

```
http://localhost:8080/refunds
```

With:

```
{{baseUrl}}/refunds
```

Replace header value:

```
Bearer my-secret-token
```

With:

```
Bearer {{token}}
```

Now all requests automatically use the environment.

==

8. Re-run Tests with Environment

==

To activate the environment:

1. Click top-right **Environment dropdown**

2. Select Local

Click **Send**

Everything works with dynamic values.

==

9. Expected Status Codes

==

Scenario	Expected
Missing Authorization header	401 Unauthorized
Wrong token	401 Unauthorized
Valid token	Normal JSON response
Refund ID not found	500 RuntimeException (custom error)