

Print Drawing API

Register User → <http://localhost:8080/register/user>

The screenshot displays the Postman interface for the 'Print Drawing API-ScottMichaelAnderson' workspace. The 'REGISTER USER' endpoint is selected, showing a POST request to `http://localhost:8080/register/user` with a JSON body containing `username` and `password` fields. The response is a 200 OK status with a message: 'Registration is successful :)'. The interface includes a sidebar with collections and environments, a top navigation bar with 'Home', 'Workspaces', and 'API Network', and a bottom status bar with various tool icons.

Home Workspaces API Network

Scott_1 New Import

POST REGISTER USER POST AUTHENTICATION GET GET ALL PRINTS

Print Drawing API-ScottMichaelAnderson / REGISTER USER

POST http://localhost:8080/register/user

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "username": "username",
3   "password": "password"
4 }
```

Body Cookies (1) Headers (14) Test Results

200 OK 507 ms 452 B Save as example

Pretty Raw Preview Visualize Text

1. Registration is successful :)

Body Cookies (1) Headers (14) Test Results

200 OK 507 ms 452 B Save as example

Pretty Raw Preview Visualize Text

1. Registration is successful :)

Online Find and replace Console Import Complete

Postbot Runner Start Proxy Cookies Vault Trash

Print Drawing API

Authenticate User → <http://localhost:8080/api/authenticate>

The screenshot displays the Postman API client interface. On the left, a sidebar shows the 'Print Drawing API-ScottMichaelAnderson' collection with various endpoints. The 'POST AUTHENTICATION' endpoint is selected. The main panel shows the request details for the URL `http://localhost:8080/api/authenticate` with a `POST` method. The request body is in JSON format, containing `username` and `password` fields. The response is shown in the 'Body' tab, displaying a long JWT token. The status bar at the bottom indicates the request was successful with a `200 OK` status.

Print Drawing API-ScottMichaelAnderson / AUTHENTICATION

POST `http://localhost:8080/api/authenticate` Send

Params Authorization Headers (9) Body Scripts Settings Cookies Beautify

`{`
 `"username": "username",`
 `"password": "password"`
`}`

Body Cookies (1) Headers (14) Test Results 200 OK 342 ms 655 B Save as example

Pretty Raw Preview Visualize Text `1 eyJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRwczovL3N1Y3VyZS5nZW51aW51Y29kZXIuY29tIiwic3ViIjoIdXN1cm5hbWU1LCJpYXQ1OjE3MjEzNTYyODMsImV4cCI6MTcyMTM1ODA4M30.1tPpioSn2PPoKMgn1U8Xb-VNY6mt6Db0idb36fiSQIViSKPMXN4t20TPhPwzHnv00K4j7AzvfPOblp6m8jMTIA`

body COOKIES (1) HEADERS (14) TEST RESULTS 200 OK 342 ms 655 B Save as example

Pretty Raw Preview Visualize Text `1 eyJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRwczovL3N1Y3VyZS5nZW51aW51Y29kZXIuY29tIiwic3ViIjoIdXN1cm5hbWU1LCJpYXQ1OjE3MjEzNTYyODMsImV4cCI6MTcyMTM1ODA4M30.1tPpioSn2PPoKMgn1U8Xb-VNY6mt6Db0idb36fiSQIViSKPMXN4t20TPhPwzHnv00K4j7AzvfPOblp6m8jMTIA`

Online Find and replace Console Import Complete Postbot Runner Start Proxy Cookies Vault Trash

Print Drawing API

GET ALL PRINTS → <http://localhost:8080/api/print?pageNo=0&pageSize=10>

The screenshot displays the Postman interface with a REST client request configured for the endpoint `http://localhost:8080/api/print?pageNo=0&pageSize=10`. The request method is `GET`. The `Authorization` tab is active, showing a `Bearer Token` type. A red box highlights the text `Enter Bearer Token` with an arrow pointing to the token input field. The `Body` tab shows the JSON response of the API call, which includes a list of print details such as `id`, `drawingName`, `customer`, `oem`, `diameter`, `faceLength`, `bearingMin`, `bearingMax`, `customerPin`, `customerRevision`, `date`, `dateCreated`, `dmgDrawingPath`, `newBasePrice`, and `originatingCustomer`.

Print Drawing API

GET ALL PRINTS → <http://localhost:8080/api/print?pageNo=0&pageSize=10>

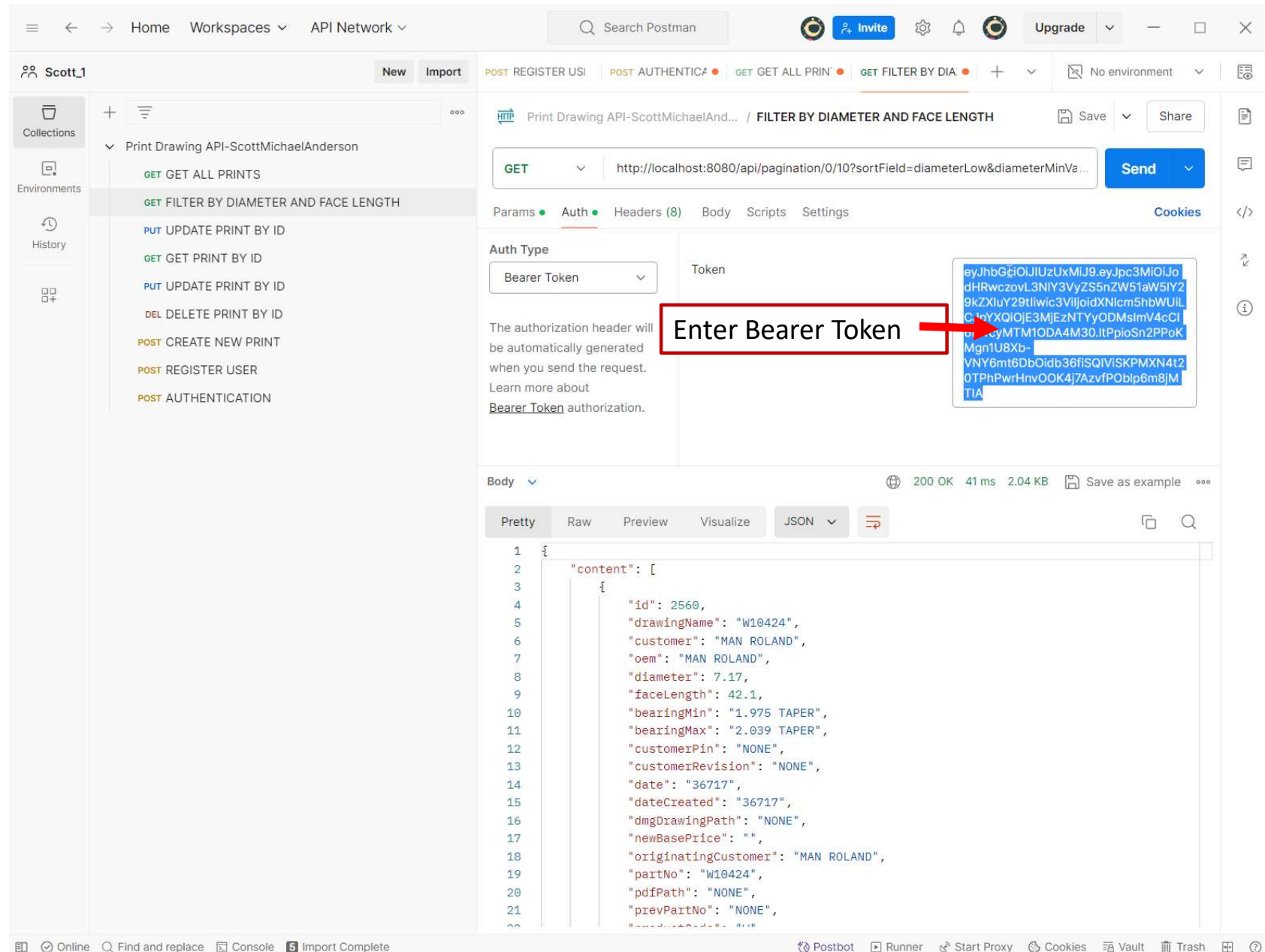
The screenshot shows the Postman interface for the 'Print Drawing API-ScottMichaelAnderson' collection. The 'GET ALL PRINTS' endpoint is selected, with the URL `http://localhost:8080/api/print?pageNo=0&pageSize=10` and the method 'GET'. The 'Query Params' tab is active, showing a table with the following data:

Key	Value	Description
pageNo	0	
pageSize	10	

A red arrow points from the 'Enter Query Params' text box to the 'Query Params' tab. The 'Body' tab shows the JSON response in 'Pretty' format:

```
1 {
2   "content": [
3     {
4       "id": 1,
5       "drawingName": "M90101",
6       "customer": "PRINTING USA",
7       "oem": "PRINTING USA",
8       "diameter": 5.14,
9       "faceLength": 35.0,
10      "bearingMin": "MANDREL",
11      "bearingMax": "MANDREL",
12      "customerPin": "NONE",
13      "customerRevision": "NONE",
14      "date": "36622",
15      "dateCreated": "36622",
16      "dmgDrawingPath": "NONE",
17      "newBasePrice": "",
18      "originatingCustomer": "SOSSINI",
```

The status bar at the bottom indicates a 200 OK response with a 139 ms response time and 5.68 KB of data.



Print Drawing API

FIND BY DIAMETER AND FACE LENGTH:

<http://localhost:8080/api/pagination/0/10?sortField=diameterLow&diameterMinValue=7.1&diameterMaxValue=7.2&faceLengthMinValue=40&faceLengthMaxValue=50>

The screenshot shows the Postman interface for the 'Print Drawing API-ScottMichaelAnderson' collection. The selected endpoint is 'GET FILTER BY DIAMETER AND FACE LENGTH'. The URL is `http://localhost:8080/api/pagination/0/10?sortField=diameterLow&diameterMinValue=7.1&diameterMaxValue=7.2&faceLengthMinValue=40&faceLengthMaxValue=50`. The 'Query Params' section is expanded, showing a table of parameters:

Key	Value	Description
sortField	diameterLow	
diameterMinValue	7.1	
diameterMaxValue	7.2	
faceLengthMinValue	40	
faceLengthMaxValue	50	

The 'Body' section shows the response in JSON format:

```
{
  "content": [
    {
      "id": 2560,
      "drawingName": "W10424",
      "customer": "MAN ROLAND",
      "oem": "MAN ROLAND",
      "diameter": 7.17,
      "faceLength": 42.1,
      "bearingMin": "1.975 TAPER",
      "bearingMax": "2.039 TAPER",
      "customerPin": "NONE",
      "customerRevision": "NONE",
      "date": "36717",
      "dateCreated": "36717",
      "dmgDrawingPath": "NONE",
      "newBasePrice": "",
      "originatingCustomer": "MAN ROLAND",
      "partNo": "W10424",
      "pdfPath": "NONE",
      "prevPartNo": "NONE",
      "..."
    }
  ]
}
```

A red arrow points from the 'Enter Query Params' text box to the 'Query Params' section of the Postman interface.

Enter Query Params

Print Drawing API

UPDATE PRINT BY ID → <http://localhost:8080/api/print/update/1>

The screenshot displays the Postman application interface. On the left sidebar, the 'Collections' panel is open, showing a collection named 'Print Drawing API-ScottMichaelAnderson'. The 'PUT UPDATE PRINT BY ID' endpoint is selected. The main panel shows the request configuration for this endpoint. The method is 'PUT' and the URL is 'http://localhost:8080/api/print/update/10'. The 'Authorization' tab is active, showing 'Auth Type' as 'Bearer Token' and a 'Token' field containing a long alphanumeric string. A red box with the text 'Enter Bearer Token' and a red arrow points to the token field. The 'Response' section is empty, showing a cartoon character and the text 'Click Send to get a response'.

Print Drawing API-ScottMichaelAnderson / UPDATE PRINT BY ID

PUT http://localhost:8080/api/print/update/10

Params Authorization Headers (10) Body Scripts Settings Cookies

Auth Type: Bearer Token

Token: eyJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRv...

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Response

Click Send to get a response

Print Drawing API

UPDATE PRINT BY ID → <http://localhost:8080/api/print/update/1>

The screenshot shows a Postman interface for a PUT request to the endpoint `http://localhost:8080/api/print/update/1`. The request body is a JSON object with the following fields:

```
1 {
2   "bearingMax": "value_bearingMax_UPDATED",
3   "bearingMin": "value_bearingMin_UPDATED",
4   "cust": "value_cust_UPDATED",
5   "custPin": "value_custPin_UPDATED",
6   "custRev": "value_custRev_UPDATED",
7   "date": "value_date_UPDATED",
8   "dateCreated": "value_dateCreated_UPDATED",
9   "dia1": "value_dia1_UPDATED",
10  "dia2": "value_dia2_UPDATED",
11  "dmgDrawingPath": "value_dmgDrawingPath_UPDATED",
12  "drawingName": "value_drawingName_UPDATED",
13  "face1": "value_face1_UPDATED",
14  "face2": "value_face2_UPDATED",
15  "newBasePrice": "value_newBasePrice_UPDATED",
16  "oem": "value_oem_UPDATED",
17  "originatingCustomer": "value_originatingCustomer_UPDATED",
18  "partNo": "value_partNo_UPDATED",
19  "pdfPath": "value_pdfPath_UPDATED",
20  "prevPartNo": "value_prevPartNo_UPDATED",
21  "productCode": "value_productCode_UPDATED",
22  "revNumber": "value_revNumber_UPDATED",
23  "scannedPath": "value_scannedPath_UPDATED",
24  "steps": "value_steps_UPDATED",
25  "subcontractor": "value_subcontractor_UPDATED",
26  "type": "value_type_UPDATED",
27  "xlsmPath": "value_xlsmPath_UPDATED",
28  "xlsxPath": "value_xlsxPath_UPDATED"
29 }
30
```

A red arrow points to the `newBasePrice` field in the request body, which is labeled "Updated Print Object Values".

The response body is a JSON object with the following fields:

```
1 {
2   "id": 1,
3   "drawingName": "value_drawingName_UPDATED",
4   "customer": "",
5   "oem": "value_oem_UPDATED",
6   "diameter": 0.0,
7   "faceLength": 0.0,
8   "bearingMin": "value_bearingMin_UPDATED",
9   "bearingMax": "value_bearingMax_UPDATED",
10  "customerPin": "",
11  "customerRevision": "",
12  "date": "value_date_UPDATED",
13  "dateCreated": "value_dateCreated_UPDATED",
14  "dmgDrawingPath": "value_dmgDrawingPath_UPDATED",
15  "newBasePrice": "",
16  "originatingCustomer": "value_originatingCustomer_UPDATED",
17  "partNo": "value_partNo_UPDATED",
18  "pdfPath": "value_pdfPath_UPDATED",
19  "prevPartNo": "value_prevPartNo_UPDATED",

```


Print Drawing API

GET PRINT BY ID → <http://localhost:8080/api/print/1>

The screenshot shows the Postman interface with a collection named "Print Drawing API-ScottMichaelAnderson". The selected endpoint is "GET PRINT BY ID". The request is configured with the following details:

- Method: GET
- URL: <http://localhost:8080/api/print/1>
- Auth Type: Bearer Token
- Token: eyJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJodHRv...

The response is a 200 OK status with a response time of 17 ms and a body size of 1.23 KB. The response body is displayed in JSON format:

```
1 {
2   "id": 1,
3   "drawingName": "value_drawingName_UPDATED",
4   "customer": "",
5   "oem": "value_oem_UPDATED",
6   "diameter": 0.0,
7   "faceLength": 0.0,
8   "bearingMin": "value_bearingMin_UPDATED",
9   "bearingMax": "value_bearingMax_UPDATED".
```

A red arrow points to the "Enter Bearer Token" text box, indicating where the token should be entered.

Print Drawing API

GET PRINT BY ID → <http://localhost:8080/api/print/update/1>

The screenshot shows the Postman interface with a GET request configured. The URL is `http://localhost:8080/api/print/1`. The Authorization tab is selected, showing a Bearer Token. The response body is displayed in the Pretty view, showing a JSON object with various fields. Two red boxes with arrows highlight specific parts of the interface: one points to the 'id' field in the JSON response, and the other points to the 'Enter Bearer Token' text in the Authorization tab.

Print Object with id = 1

Enter Bearer Token

```
{
  "id": 1,
  "drawingName": "v",
  "customer": "",
  "oem": "value_oem_UPDATED",
  "diameter": 0.0,
  "faceLength": 0.0,
  "bearingMin": "value_bearingMin_UPDATED",
  "bearingMax": "value_bearingMax_UPDATED",
  "customerPin": "",
  "customerRevision": "",
  "date": "value_date_UPDATED",
  "dateCreated": "value_dateCreated_UPDATED",
  "dmgDrawingPath": "value_dmgDrawingPath_UPDATED",
  "newBasePrice": "",
  "originatingCustomer": "value_originatingCustomer_UPDATED",
  "partNo": "value_partNo_UPDATED",
  "oem": "value_oem_UPDATED",
  "diameter": 0.0,
  "faceLength": 0.0,
  "bearingMin": "value_bearingMin_UPDATED",
  "bearingMax": "value_bearingMax_UPDATED",
  "customerPin": "",
  "customerRevision": "",
  "date": "value_date_UPDATED",
  "dateCreated": "value_dateCreated_UPDATED",
  "dmgDrawingPath": "value_dmgDrawingPath_UPDATED",
  "newBasePrice": "",
  "originatingCustomer": "value_originatingCustomer_UPDATED",
  "partNo": "value_partNo_UPDATED",
  "pdfPath": "value_pdfPath_UPDATED",
  "prevPartNo": "value_prevPartNo_UPDATED",
  "productCode": "value_productCode_UPDATED",
  "revNumber": "value_revNumber_UPDATED",
  "scannedPath": "value_scannedPath_UPDATED",
  "steps": "value_steps_UPDATED",
  "subcontractor": "value_subcontractor_UPDATED",
  "type": "value_type_UPDATED",
  "xlsmPath": "value_xlsmPath_UPDATED",
  "xlsxPath": "value_xlsxPath_UPDATED"
}
```

Print Drawing API

DELETE PRINT BY ID → <http://localhost:8080/api/print/delete/1>

The screenshot shows the Postman interface for a REST client. The left sidebar displays a collection named 'Print Drawing API-ScottMichaelAnderson' with several endpoints. The 'DEL DELETE PRINT BY ID' endpoint is selected. The main panel shows the configuration for this endpoint: the method is 'DELETE', the URL is 'http://localhost:8080/api/print/delete/1', and the authorization is set to 'Bearer Token'. A red box labeled 'Enter Bearer Token' points to the token input field, which contains a long alphanumeric string. The bottom panel shows the response body in 'Pretty' format, displaying a single line: '1 Successfully deleted print drawing id = 1'. A red box labeled 'Deleted Print Object with id=1' points to this response line. The status bar at the bottom indicates a '200 OK' response with a 31 ms latency and 464 B of data.

Deleted Print Object with id=1

Enter Bearer Token

1 Successfully deleted print drawing id = 1

Print Drawing API

CREATE NEW PRINT → <http://localhost:8080/api/print/create>

The screenshot displays the Postman interface for a REST client. On the left, the 'Collections' panel shows a collection named 'Print Drawing API-ScottMichaelAnderson' with several endpoints. The 'POST CREATE NEW PRINT' endpoint is selected. The main panel shows the details of this endpoint, including the URL 'http://localhost:8080/api/print/create', the method 'POST', and the 'Bearer Token' authorization type. A red box with an arrow points to the 'Token' field, which contains the value 'eyJhbGciOiJIUzUxMi9.eyJpc3MiOiJodHRv...'. Below the authorization section, the 'Body' tab is active, showing a JSON response in 'Pretty' format. The response is a JSON object with the following structure:

```
1 {
2   "id": 5000,
3   "drawingName": "value_drawingName",
4   "customer": "",
5   "oem": "value_oem",
6   "diameter": 0.0,
7   "faceLength": 0.0,
8   "bearingMin": "value_bearingMin",
9   "bearingMax": "value_bearingMax",
10  "customerPin": "",
11  "customerRevision": "",
12  "date": "value_date",
13  "dateCreated": "value_dateCreated",
14  "dmgDrawingPath": "value_dmgDrawingPath",
15  "newBasePrice": "",
16  "originatingCustomer": "value_originatingCustomer",
17  "partNo": "value_partNo",
18  "pdfPath": "value_pdfPath",
19  "prevPartNo": "value_prevPartNo",
```

The status bar at the bottom indicates a successful response with a 201 status code, 44 ms response time, and 1.09 KB body size. The bottom of the interface shows various utility buttons like 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and 'Help'.

Print Drawing API

CREATE NEW PRINT → <http://localhost:8080/api/print/create>

The screenshot shows the Postman interface with a POST request to `http://localhost:8080/api/print/create`. The request body is a JSON object with various fields. A red box labeled "New Print Object" points to the request body. The response is a JSON object with an `id` of 5000. A red box labeled "New Print Object with id=5000" points to the response body.

Request Body (JSON):

```
1 {
2   "bearingMax": "value_bearingMax",
3   "bearingMin": "value_bearingMin",
4   "cust": "value_cust",
5   "custPin": "value_custPin",
6   "custRev": "value_custRev",
7   "date": "value_date",
8   "dateCreated": "value_dateCreated",
9   "dia1": "value_dia1",
10  "dia2": "value_dia2",
11  "drawingName": "value_drawingName",
12  "dmgDrawingPath": "value_dmgDrawingPath",
13  "face1": "value_face1",
14  "face2": "value_face2",
15  "newBasePrice": "value_newBasePrice",
16  "oem": "value_oem",
17  "originatingCustomer": "value_originatingCustomer",
18  "partNo": "value_partNo",
19  "pdfPath": "value_pdfPath",
20  "prevPartNo": "value_prevPartNo",
21  "productCode": "value_productCode",
22  "revNumber": "value_revNumber",
23  "scannedPath": "value_scannedPath",
24  "steps": "value_steps",
25  "subcontractor": "value_subcontractor",
26  "type": "value_type",
27  "xlsPath": "value_xlsPath",
28  "xlsxPath": "value_xlsxPath"
29 }
```

Response Body (JSON):

```
1 {
2   "id": 5000,
3   "drawingName": "value_drawingName",
4   "customer": "",
5   "oem": "value_oem",
6   "diameter": 0.0,
7   "faceLength": 0.0,
8   "bearingMin": "value_bearingMin",
9   "bearingMax": "value_bearingMax",
10  "customerPin": "",
11  "customerRevision": "",
12  "date": "value_date",
13  "dateCreated": "value_dateCreated",
14  "dmgDrawingPath": "value_dmgDrawingPath",
15  "newBasePrice": "",
16  "originatingCustomer": "value_originatingCustomer",
17  "partNo": "value_partNo",
18  "pdfPath": "value_pdfPath",
19  "prevPartNo": "value_prevPartNo",
20 }
```