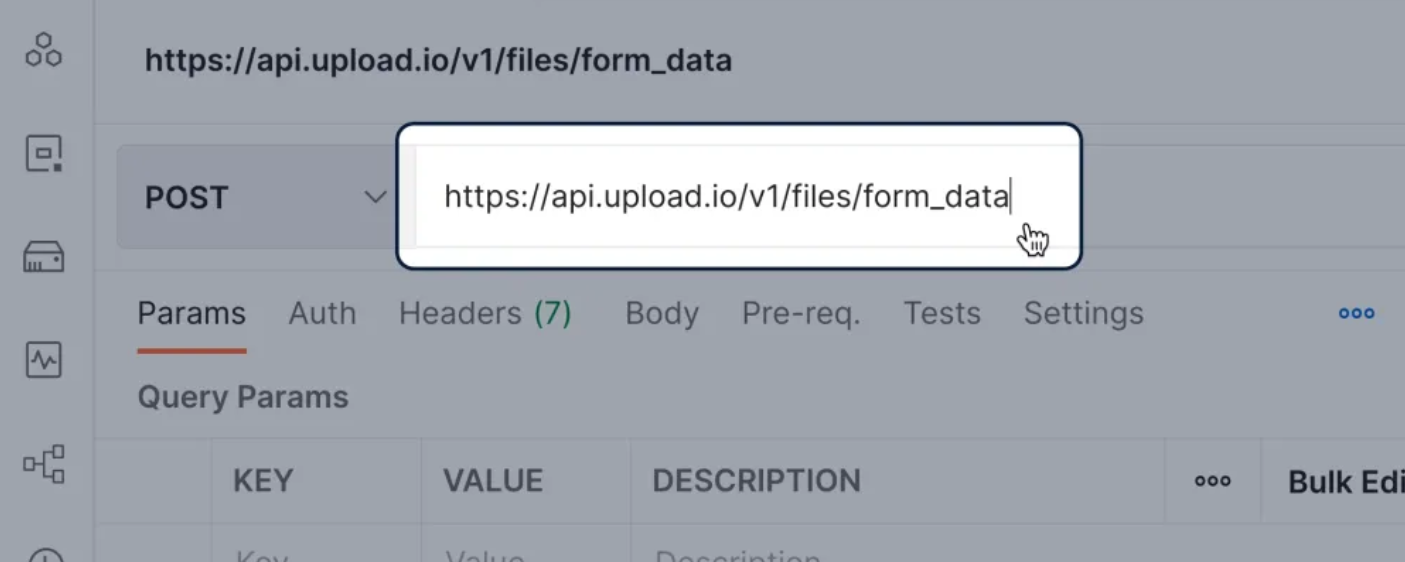
Sending and testing multipart/form-data requests

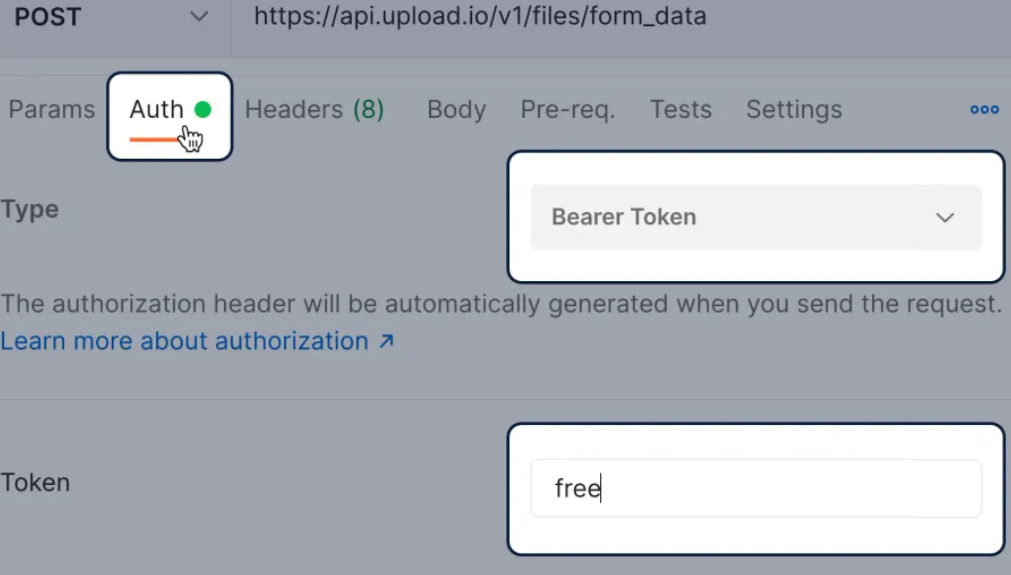
# Steps

1. Create a new POST request
2. Provide the end point url

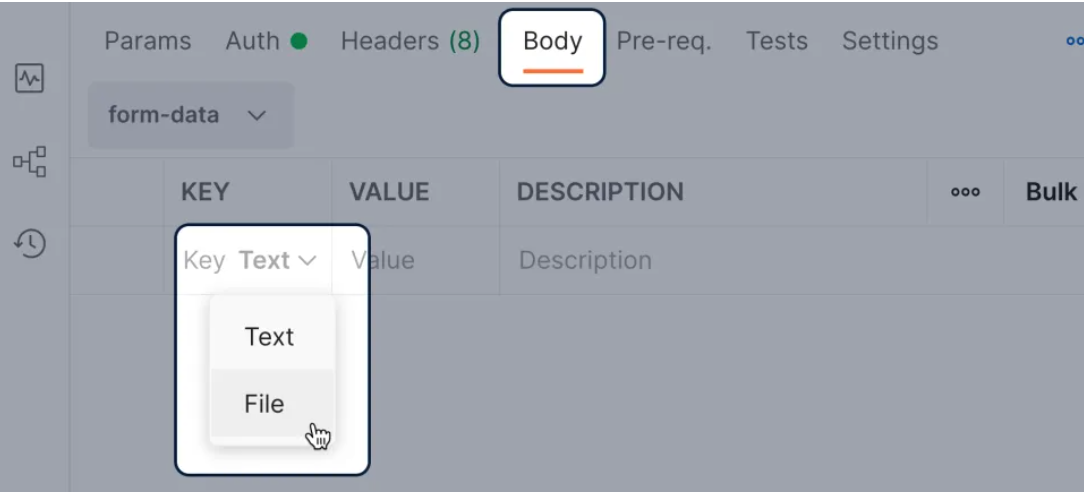
[**https://api.upload.io/v1/files/form\_data**](https://api.upload.io/v1/files/form_data)

****

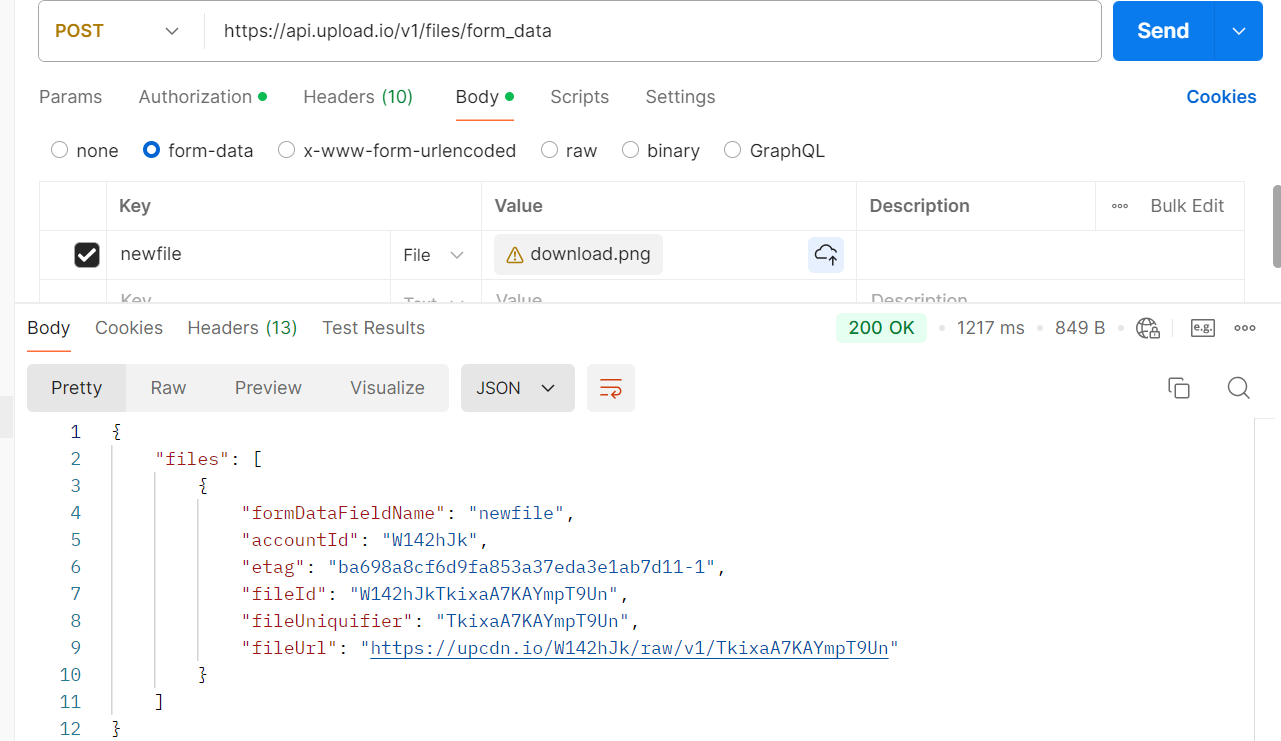
1. Set the authorization header

****

1. **Set the body to "form-data" and Select "File" from the dropdown**

****

1. Click on Send
2. Validate the response

****

1. Add assertions for status code and file name

pm.test("Testing upload", ()=>{

    pm.response.to.have.status(200)

})

**const** jsonData= pm.response.json();

pm.test("Testing upload file", ()=>{

    pm.expect(jsonData.files[0].formDataFieldName).to.eql("newfile")

})

1. Export the collection and run using Newman

# Selecting binary option

In the previous example we used multipart/form-data for our request body.

But what about using Postman to perform a binary file upload instead?

First, set the body to **"binary"**

Select the file and send the request.

