

Query parameter –

```
1 describe("API testing", () => {
2
3   //store query param in a variable.
4   const queryParams = { page: 2 };
5
6
7   it("Passing Query parameters", () => {
8
9     cy.request({
10       method: 'GET',
11       url: 'https://reqres.in/api/users',
12       // qs: {page: 2}
13       qs: queryParams
14     })
15     .then((response) => {
16       expect(response.status).to.eq(200);
17       expect(response.status).equal(200);
18       expect(response.body.page).to.eq(2);
19       expect(response.body.data).toHaveLength(6);
20       expect(response.body.data[0]).have.property('id', 7);
21       expect(response.body.data[0]).have.property('first_name', 'Michael')
22     })
23   })
24
25 })
26
27
28
29
30 })
```

codesnap.dev

Token –

```
1 describe("API testing",()=>{
2
3     let authToken=null;
4
5     //before hook for generating access token.
6     //when multiple before blocks written, then the order in which
7     //they are written, in the same order they run.
8
9     before("creating access token",()=>{
10
11         cy.request({
12             method: 'POST',
13             url: 'https://simple-books-api.glitch.me/api-clients/',
14             failOnStatusCode: false,
15             headers: {
16                 'Content-Type': 'application/json'
17             },
18             body:{
19                 clientName: 'ABC',
20                 clientEmail: Math.random().
21 toString(5).substring(2)+"@gmail.com"
22             }
23         }).then((response) =>{
24             authToken=response.body.accessToken;
25         });
26     });
```

```
1 before("creating new order",()=>{
2
3     cy.request({
4         method: 'POST',
5         url: 'https://simple-books-api.glitch.me/orders/',
6         failOnStatusCode: false,
7         headers: {
8             'Content-Type': 'application/json',
9             'Authorization': 'Bearer '+authToken
10        },
11        body:{
12            "bookId": 1,
13            "customerName": "xyzabc"
14        }
15    }).then((response) =>{
16
17        expect(response.status).to.eq(201);
18        expect(response.body.created).to.eq(true);
19    });
20 });
```

```
1  it("Fetching the orders",()=>{
2
3      cy.request(
4          { method: 'GET',
5            url: 'https://simple-books-api.glitch.me/orders/',
6            failOnStatusCode: false,
7            headers:{
8                'Content-Type': 'application/json',
9                'Authorization': 'Bearer '+authToken
10           },
11           cookies:{
12               'cookieName': 'mycookie'
13           }
14      }). then((response)=>{
15          expect(response.status).to.eq(200);
16          //to check the length of json array.
17          expect(response.body).has.length(1);
18      })
19  })
20 })
```