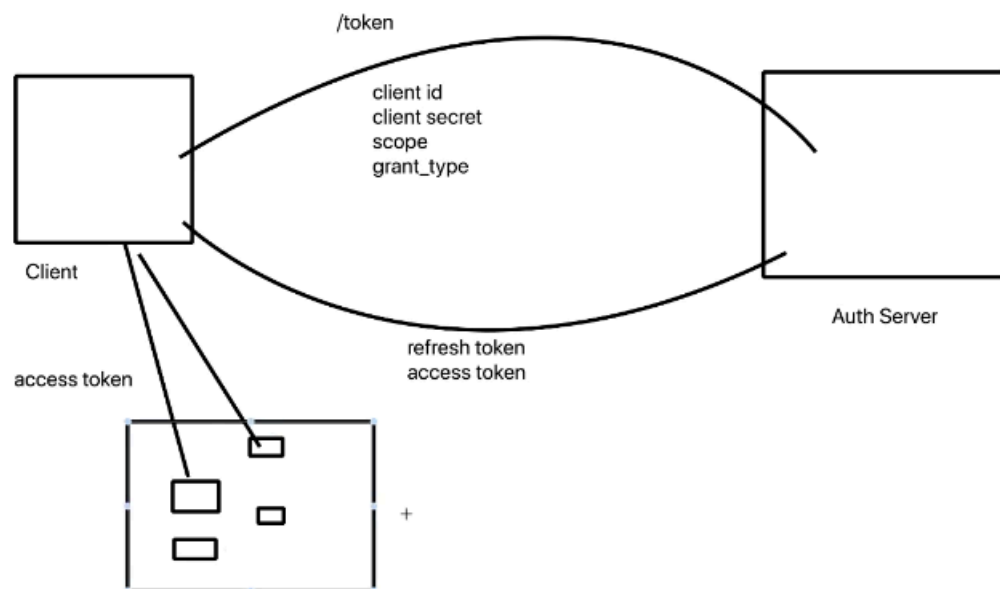




OAuth 2.0 Flow



Application to create for oauth 2.0-

coop.apps.symfonycasts.com

Not Secure | coop.apps.symfonycasts.com/application

AppsJava Wrapper Cla...Top 30 Patterns In...2021\_March\_Eve...2021\_July\_Mamin...2021\_June\_V

# The COOP

## Let's get you started

Application Name

CypressApp

Redirect URI

http://www.naveenautomationlabs.com

Scope

What is it you want to do?

☒ Unlock the Barn

☒ Put the Toilet Seat Down

☒ Feed Your Chickens

☒ Collect Eggs from Your Chickens

☒ Get the Number of Eggs Collected Today

☒ Access Your Profile Data

Cancel

Submit

coop.apps.symfonycasts.com/ X +

← → ↻ Not Secure | coop.apps.symfonycasts.com/application

Apps Java Wrapper Cla... M Top 30 Patterns In... 2021\_March\_Eve... 2021\_July\_Momih... 2021\_June\_Videos 2021\_July\_Code Set Up a Local Sol... Why We Sho

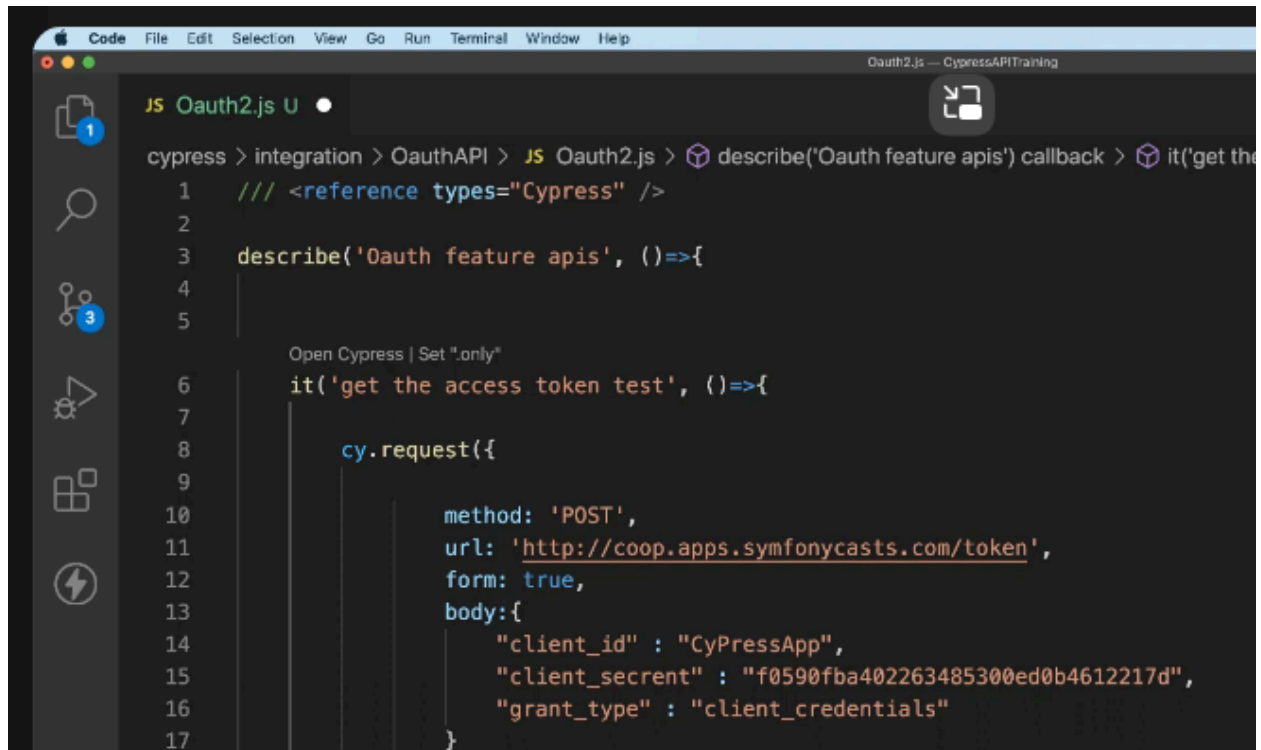
**The COOP** Logout API Built

Congratulations! You've created your application!

*CypressApp*

Client ID	CypressApp
Client Secret	f0590fba402263485300ed0b4612217d
Redirect URI	http://www.naveenautomationlabs.com
Scope	<ul style="list-style-type: none"><li>• barn-unlock</li><li>• toiletseat-down</li><li>• chickens-feed</li><li>• eggs-collect</li><li>• eggs-count</li><li>• profile</li></ul>

```
Code File Edit Selection View Go Run Terminal Window Help
1 JS Oauth2.js 2, U
cypress > integration > OauthAPI > JS Oauth2.js > de
1  /// <reference types="Cypress" />
2
3  describe('Oauth feature apis', ()=>{
4
```

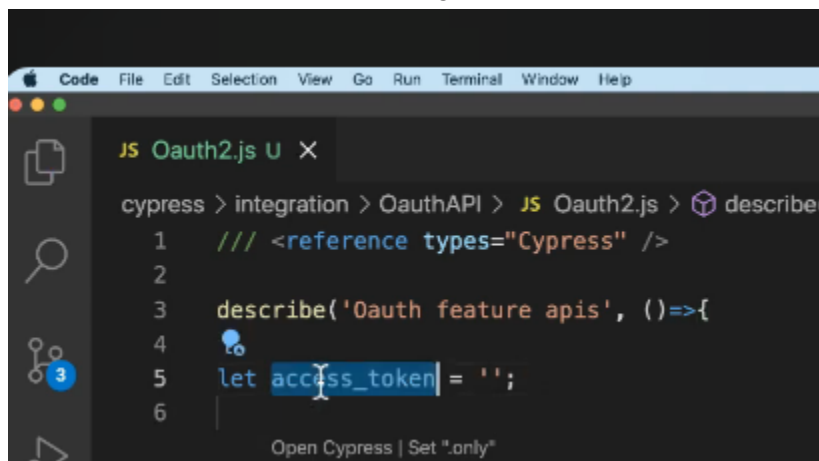


The screenshot shows a VS Code editor window with a file named 'Oauth2.js'. The code is a Cypress test suite for an OAuth2 API. It starts with a 'describe' block for 'OAuth feature apis' and an 'it' block for 'get the access token test'. Inside the 'it' block, there is a 'cy.request' call with a POST method to the URL 'http://coop.apps.symfonycasts.com/token'. The request body contains 'client\_id', 'client\_secret', and 'grant\_type' set to 'client\_credentials'.

```
1  /// <reference types="Cypress" />
2
3  describe('OAuth feature apis', ()=>{
4
5      // Open Cypress | Set ".only"
6      it('get the access token test', ()=>{
7
8          cy.request({
9
10             method: 'POST',
11             url: 'http://coop.apps.symfonycasts.com/token',
12             form: true,
13             body:{
14                 "client_id" : "CyPressApp",
15                 "client_secret" : "f0590fba402263485300ed0b4612217d",
16                 "grant_type" : "client_credentials"
17             }
18         })
19     })
20 }
```

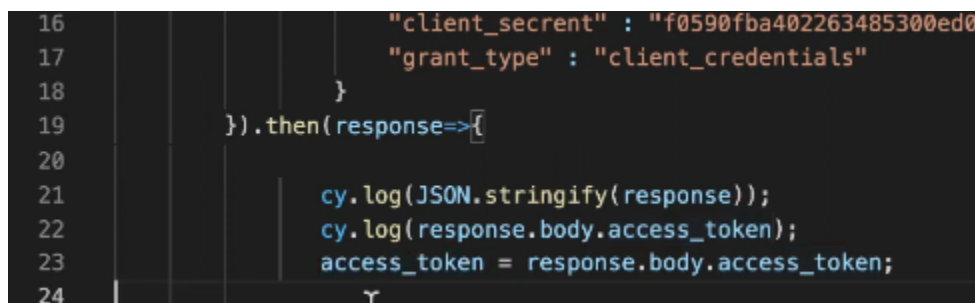
Added5

Access token variable created at global level.



This screenshot shows the same VS Code editor window, but now with a global variable 'access\_token' declared at the top of the test suite, before the 'describe' block. The variable is initialized as an empty string.

```
1  /// <reference types="Cypress" />
2
3  describe('OAuth feature apis', ()=>{
4
5      let access_token = '';
6
7      // Open Cypress | Set ".only"
8      it('get the access token test', ()=>{
9
10         cy.request({
11
12             method: 'POST',
13             url: 'http://coop.apps.symfonycasts.com/token',
14             form: true,
15             body:{
16                 "client_id" : "CyPressApp",
17                 "client_secret" : "f0590fba402263485300ed0b4612217d",
18                 "grant_type" : "client_credentials"
19             }
20         })
21     })
22 }
```



This is a close-up view of the bottom part of the code from the previous screenshot. It shows the '.then' callback of the 'cy.request' call. The response is logged using 'cy.log', and the 'access\_token' variable is updated with the value from 'response.body.access\_token'.

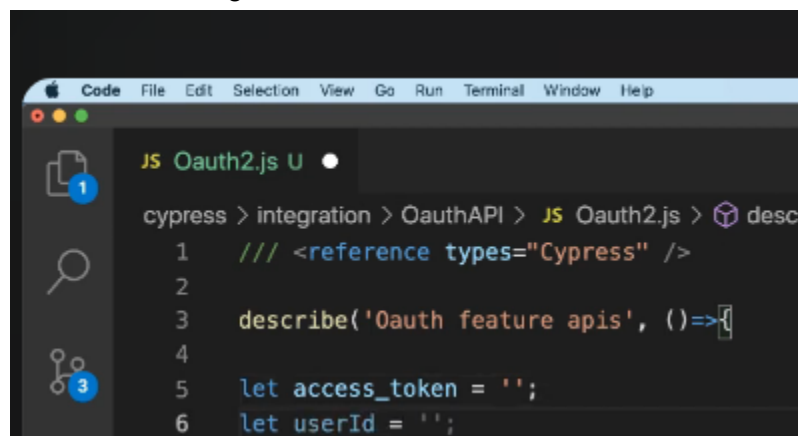
```
16         "client_secret" : "f0590fba402263485300ed0b4612217d",
17         "grant_type" : "client_credentials"
18     }
19     }).then(response=>{
20
21         cy.log(JSON.stringify(response));
22         cy.log(response.body.access_token);
23         access_token = response.body.access_token;
24     })
25 }
```

Use get call with the access token to get the user ids.

```
23     access_token = response.body.access_token;
24     cy.request({
25       method: 'GET',
26       url: 'http://coop.apps.symfonycasts.com/api/me',
27       headers: {
28         'Authorization' : 'Bearer ' + access_token
29       }
30     })
```

Line6

Added user id at global level.



```
JS Oauth2.js U
cypress > integration > OauthAPI > JS Oauth2.js > desc
1  /// <reference types="Cypress" />
2
3  describe('Oauth feature apis', ()=>{
4
5    let access_token = '';
6    let userId = '';
```

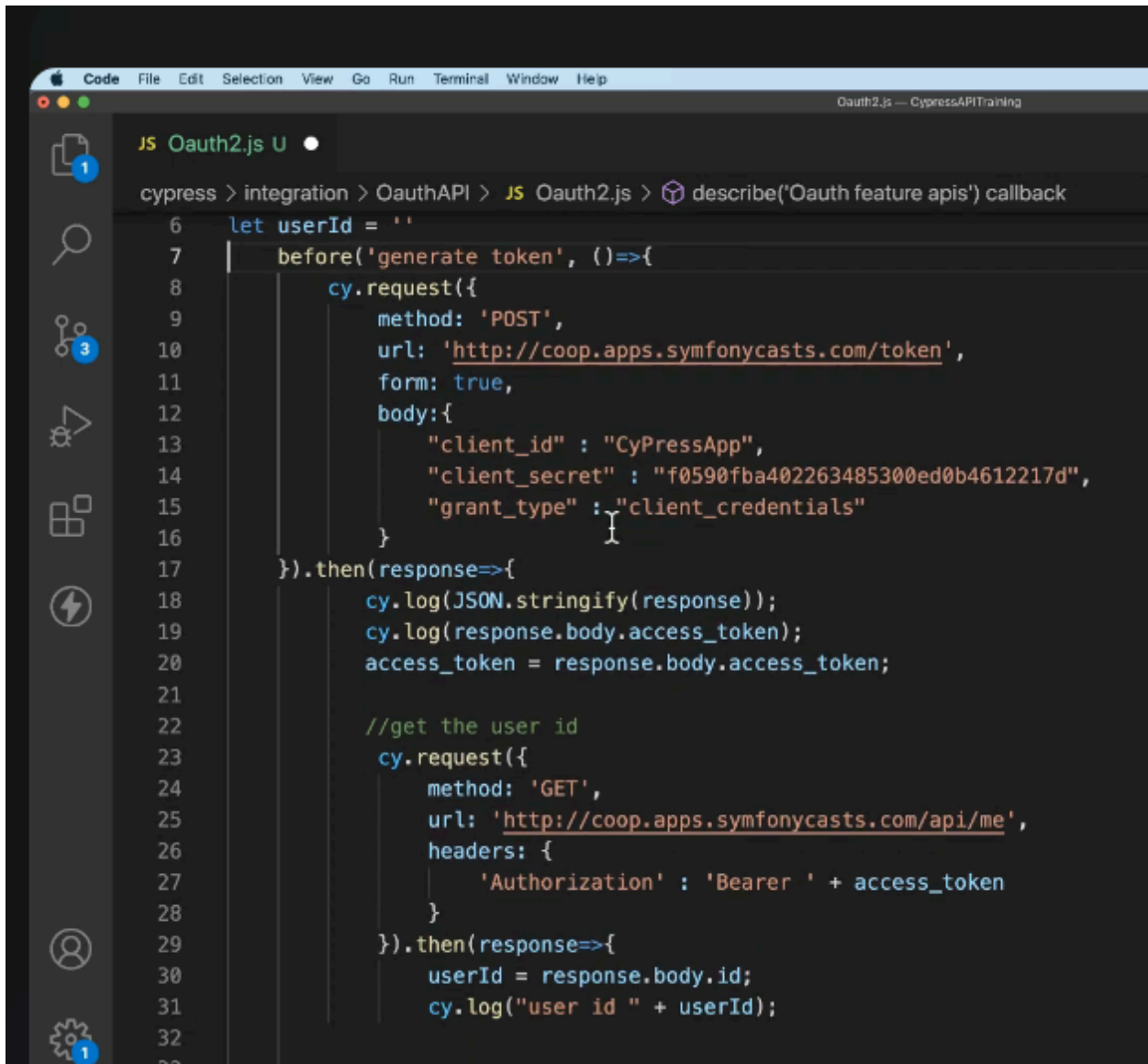
Use post call with the user id.

```
29       'Authorization' : 'Bearer ' + access_token
30     }
31   }).then(response=>{
32     userId = response.body.id;
33     cy.log("user id " + userId);
34     cy.request({
35       method: 'POST',
36       url: 'http://coop.apps.symfonycasts.com/api/'+userId+'/chickens-feed',
37       headers: {
38         'Authorization' : 'Bearer ' + access_token
39     }
```

```
38       'Authorization' : 'Bearer ' + access_token
39     }
40   }).then(response=>{
41     cy.log(JSON.stringify(response));
42     expect(response.body.status).toEqual(200);
43   })
```

Before call

To run before all the tests. Upto user id generation.



```
JS Oauth2.js U ●
cypress > integration > OauthAPI > JS Oauth2.js > describe('Oauth feature apis') callback
6   let userId = ''
7   before('generate token', ()=>{
8     cy.request({
9       method: 'POST',
10      url: 'http://coop.apps.symfonycasts.com/token',
11      form: true,
12      body:{
13        "client_id" : "CyPressApp",
14        "client_secret" : "f0590fba402263485300ed0b4612217d",
15        "grant_type" : "client_credentials"
16      }
17    }).then(response=>{
18      cy.log(JSON.stringify(response));
19      cy.log(response.body.access_token);
20      access_token = response.body.access_token;
21
22      //get the user id
23      cy.request({
24        method: 'GET',
25        url: 'http://coop.apps.symfonycasts.com/api/me',
26        headers: {
27          'Authorization' : 'Bearer ' + access_token
28        }
29      }).then(response=>{
30        userId = response.body.id;
31        cy.log("user id " + userId);
32      });
33    });
```

Then you can use different it blocks for different scenarios.

```

30         userId = response.body.id;
31         cy.log("user id " + userId);
32     })
33
34     Open Cypress | Set "only"
35     it('Unlock the Barn Test', ()=>{
36         cy.request({
37             method: 'POST',
38             url: 'http://coop.apps.symfonycasts.com/api/'+userId+'/barn-unlock',
39             headers: {
40                 'Authorization': 'Bearer ' + access_token
41             }
42         }).then(response=>{
43             cy.log(JSON.stringify(response));
44             expect(response.status).to.equal(200);
45         })
46     })
47

```

```

42         cy.log(JSON.stringify(response));
43         expect(response.status).to.equal(200);
44     })
45
46     })
47
48     Open Cypress | Set "only"
49     it('Put the Toilet Seat Down Test', ()=>{
50         cy.request({
51             method: 'POST',
52             url: 'http://coop.apps.symfonycasts.com/api/'+userId+'/toiletseat-down',
53             headers: {
54                 'Authorization' : 'Bearer ' + access_token
55             }
56         }).then(response=>{
57             cy.log(JSON.stringify(response));
58             expect(response.status).to.equal(200);
59         })
60     })
61

```

```

56     cy.log(JSON.stringify(response));
57     expect(response.status).to.equal(200);
58   })
59 })
60
61   Open Cypress | Set ".only"
62   it('Chicekn Feed Test', ()=>{
63     cy.request({
64       method: 'POST',
65       url: 'http://coop.apps.symfonycasts.com/api/'+userId+'/hickens-feed',
66       headers: {
67         'Authorization' : 'Bearer ' + access_token
68       }
69     }).then(response=>{
70       cy.log(JSON.stringify(response));
71       expect(response.status).to.equal(200);
72     })
73   })

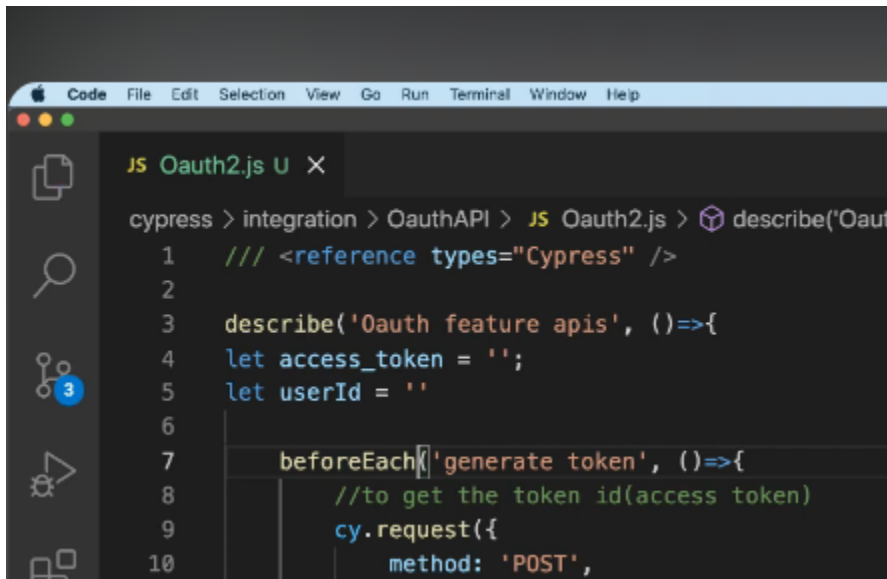
```

Changed7

Before each.

This will run before each test.

Good practice as token will never expire unnecessarily in test environment.



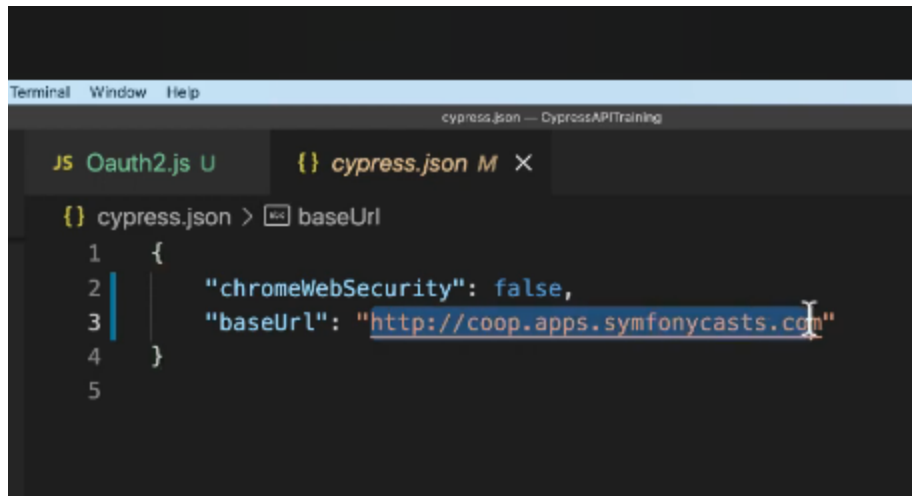
```

VS Oauth2.js U X
cypress > integration > OauthAPI > JS Oauth2.js > describe("Oauth
1  /// <reference types="Cypress" />
2
3  describe('Oauth feature apis', ()=>{
4    let access_token = '';
5    let userId = ''
6
7    beforeEach('generate token', ()=>{
8      //to get the token id(access token)
9      cy.request({
10       method: 'POST',

```

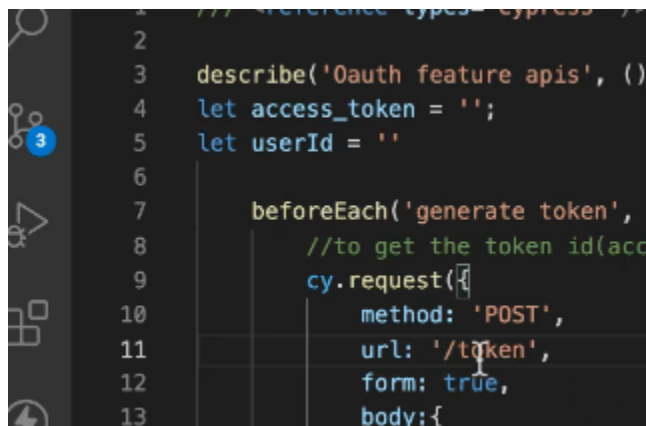
Maintain base url in cypress json file





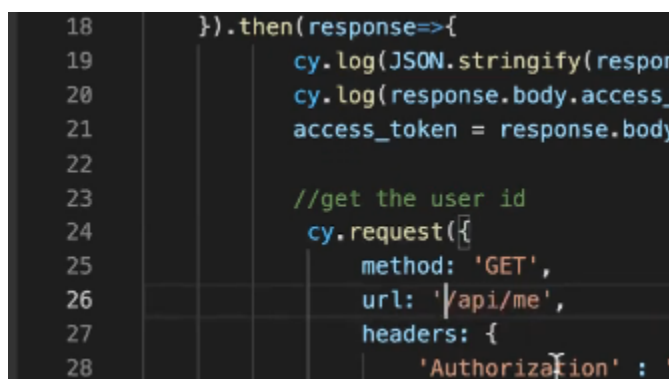
Changed11

How to use the base url in class.



changed

26. how to use the base url in class.



Change39.

Same like above, changed base url.

```

31     userId = response.body.id;
32     cy.log("user id " + userId);
33   })
34 })
35
36   Open Cypress | Set "only"
37   it('Unlock the Barn Test', ()=>{
38     cy.request({
39       method: 'POST',
40       url: 'http://coop.apps.symfonycasts.com/api/'+userId+'/barn-unlock',
41       headers: {
42         'Authorization' : 'Bearer ' + access_token

```

Change66  
Changed base url.

```

58     cy.log(JSON.stringify(response));
59     expect(response.status).to.equal(200);
60   })
61 })
62
63   Open Cypress | Set "only"
64   it('Chicekn Feed Test', ()=>{
65     cy.request({
66       method: 'POST',
67       url: '/api/'+userId+'/chickens-feed',
68       headers: {
69         'Authorization' : 'Bearer ' + access_token
70     }, then(response=>{

```