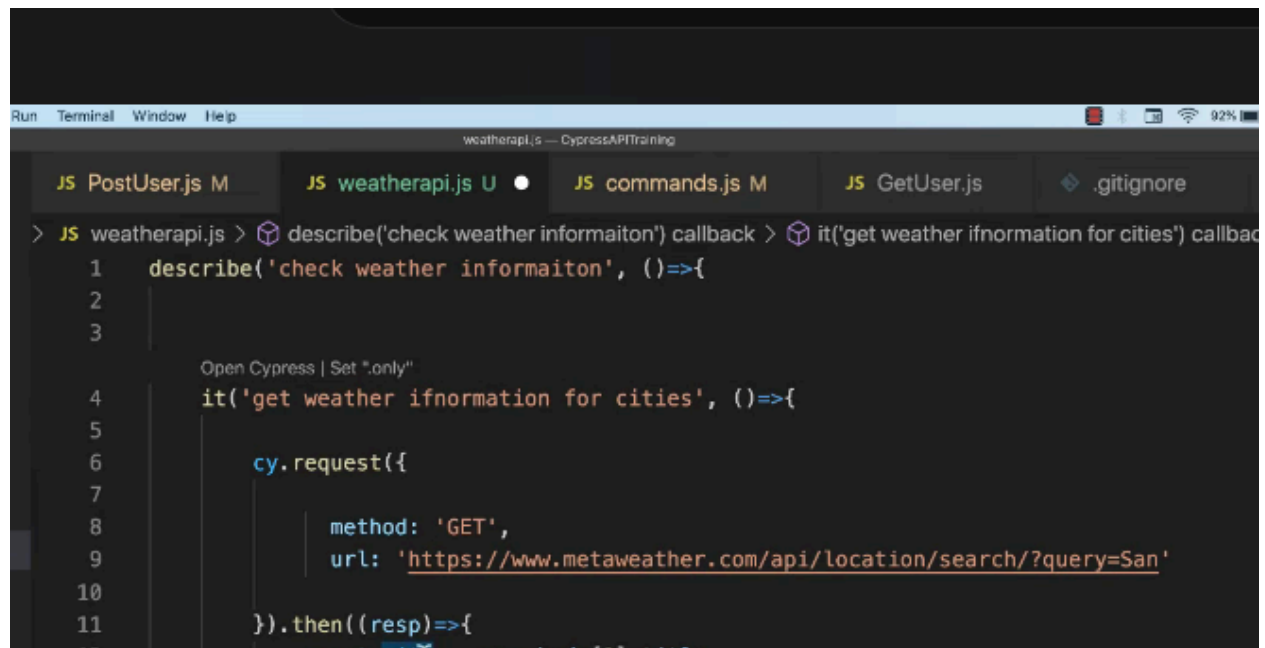


How to pass single value or complete array from one request to another request using chaining in Cypress

Chaining



```
JS PostUser.js M JS weatherapi.js U JS commands.js M JS GetUser.js .gitignore
> JS weatherapi.js > describe('check weather informaiton') callback > it('get weather ifnformation for cities') callback
1 describe('check weather informaiton', ()=>{
2
3
4   it('get weather ifnformation for cities', ()=>{
5
6     cy.request({
7
8       method: 'GET',
9       url: 'https://www.metaweather.com/api/location/search/?query=San'
10
11     }).then((resp)=>{
12       const {city} = resp.body[0].title
```

Using return also we can pass data to next then call

```

9       url: 'https://www.metaweather.com/api/location/search/?query=San'
10
11     }).then((resp) => {
12       const city = resp.body[0].title
13       return city
14     })
15     .then((city) => {
16       cy.request({
17         method: 'GET',
18         url: 'https://www.metaweather.com/api/location/search/?query='+city
19       })

```

```

17       url: 'https://www.metaweather.com/api/location/search/?query='+city
18     }).then((resp) => {
19       expect(resp.status).to.eq(200)
20       expect(resp.body[0]).to.have.property('title', city)
21     })
22
23   })

```

Another chaining

```

27
28
29   it.only('get weather information for all cities', () => {
30     //1st request: GET locations
31     cy.request({
32       method: 'GET',
33       url: 'https://www.metaweather.com/api/location/search/?query=Am'
34

```

```

35     }).then((resp) => {
36       const location = resp.body
37       return location
38     })

```

Using for loop to iterate and apply get request.

```
36     const location = resp.body
37     return location
38   })
39   .then((location)=>{
40
41     for(let i=0; i< location.length; i++){
42       //2nd request for the first location/city
43       cy.request({
44         method: 'GET',
45         url: 'https://www.metaweather.com/api/location/search/?query='+location[i].title
46       }).then((resp)=>{
47         expect(resp.status).to.eq(200)
48         expect(resp.body[0]).to.have.property('title', location[i].title)
49       })
50     }
51   })
52
53 }
```