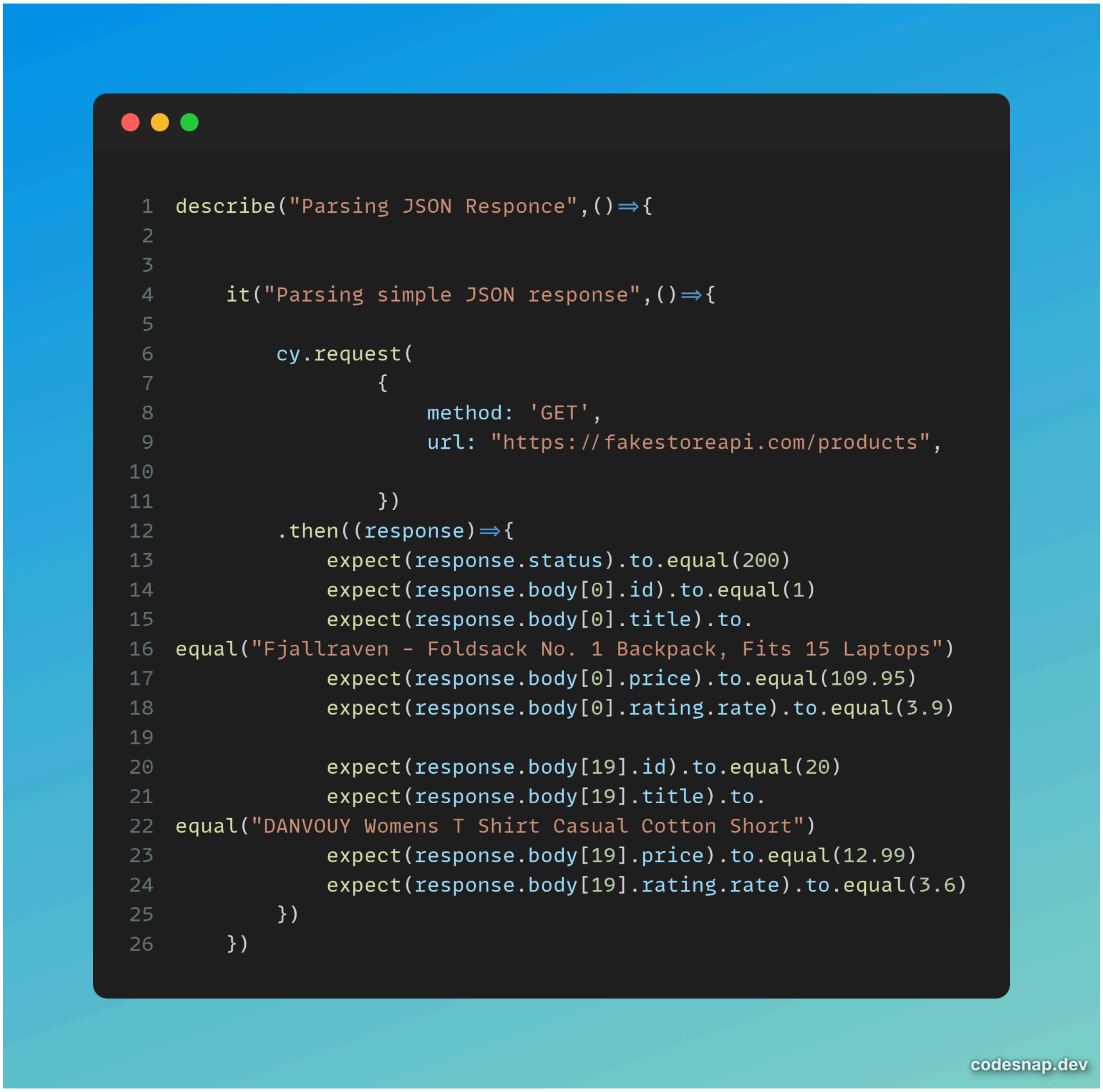
Json path finder good for finding the json path.



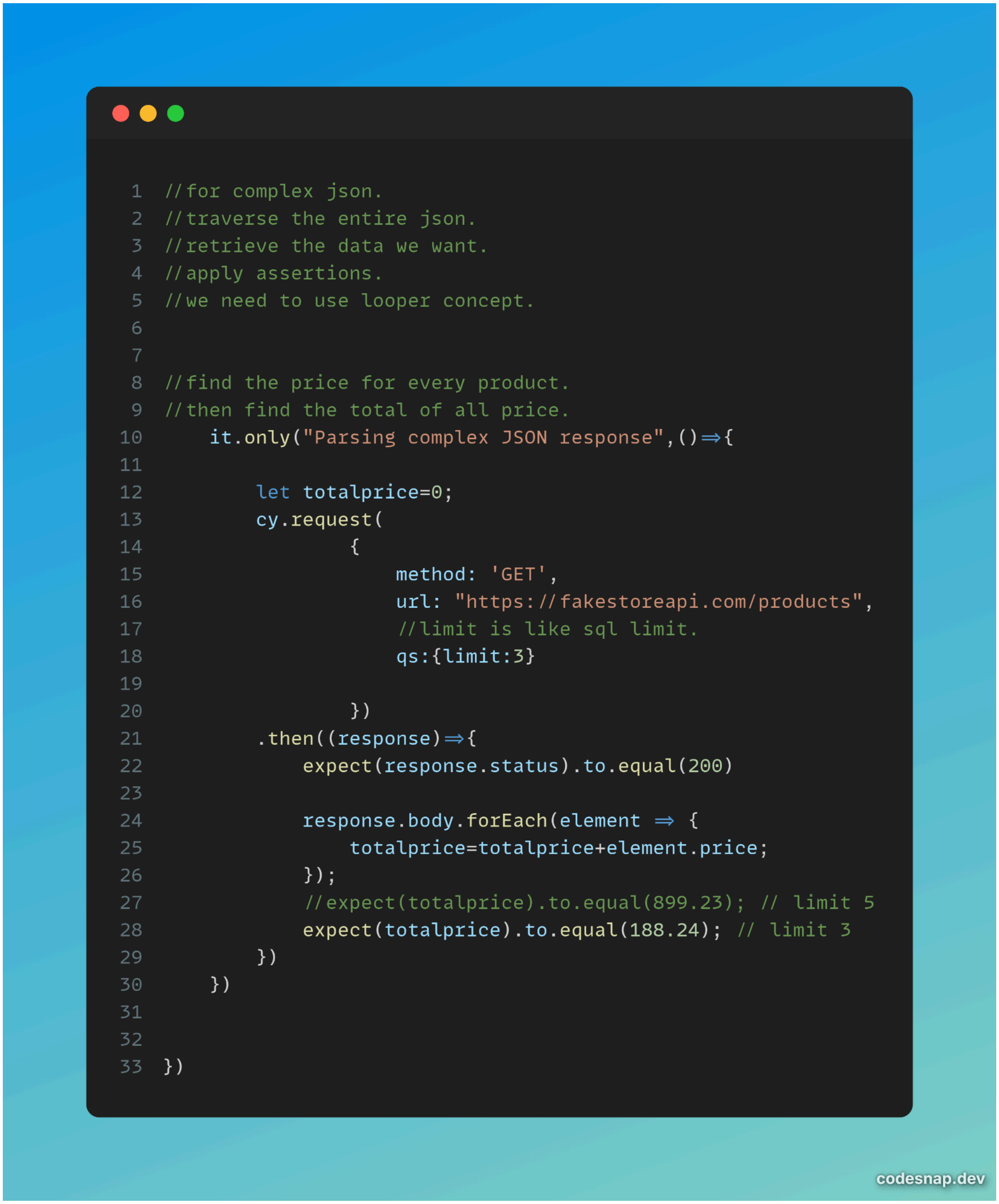
```
describe("Parsing JSON Responce",()=>{


    it("Parsing simple JSON response",()=>{

        cy.request(
                {
                    method: 'GET',
                    url: "https://fakestoreapi.com/products",

                })
        .then((response)=>{
            expect(response.status).to.equal(200)
            expect(response.body[0].id).to.equal(1)
            expect(response.body[0].title).to.
equal("Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops")
            expect(response.body[0].price).to.equal(109.95)
            expect(response.body[0].rating.rate).to.equal(3.9)

            expect(response.body[19].id).to.equal(20)
            expect(response.body[19].title).to.
equal("DANVOUY Womens T Shirt Casual Cotton Short")
            expect(response.body[19].price).to.equal(12.99)
            expect(response.body[19].rating.rate).to.equal(3.6)
        })
    })
```



```
//for complex json.
//traverse the entire json.
//retrieve the data we want.
//apply assertions.
//we need to use looper concept.


//find the price for every product.
//then find the total of all price.
    it.only("Parsing complex JSON response",()=>{

        let totalprice=0;
        cy.request(
                {
                    method: 'GET',
                    url: "https://fakestoreapi.com/products",
                    //limit is like sql limit.
                    qs:{limit:3}

                })
        .then((response)=>{
            expect(response.status).to.equal(200)

            response.body.forEach(element => {
                totalprice=totalprice+element.price;
            });
            //expect(totalprice).to.equal(899.23); // limit 5
            expect(totalprice).to.equal(188.24); // limit 3
        })
    })


})
```

codesnap.dev