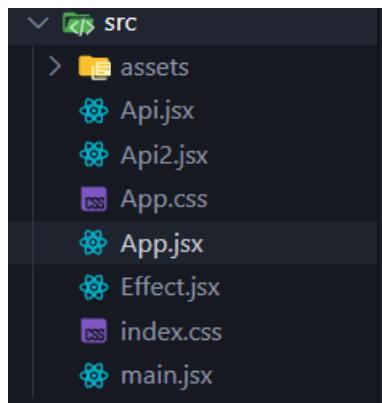


## Archivos



## App.jsx

```
1 import Effect from "./Effect";
2 import Api from "./Api";
3 import Api2 from "./Api2"
4
5 function App() {
6   return (
7     <div>
8       <Effect></Effect>
9       <Api></Api>
10      <Api2></Api2>
11    </div>
12  )
13}
14 export default App
```

## Effect.jsx

```
src > Effect.jsx > default
1 import { useEffect } from "react";
2
3 function Effect(){
4     useEffect(()=>{
5         console.log("Hola UseEffect")
6     },[])
7     return(
8         <div>
9             <h2>Hook UseEffect</h2>
10            </div>
11    )
12 }
13 export default Effect;
```

## Thunder client (Visual Studio Code)

The screenshot shows the Thunder client interface. On the left, there is a request configuration panel with a dropdown menu set to 'GET', a URL input field containing 'https://dog.ceo/api/breeds/image/random', and a 'Send' button. Below the URL are tabs for 'Query', 'Headers 2', 'Auth', 'Body', 'Tests', and 'Pre Run'. Under 'Query Parameters', there is a table with one row: a checkbox labeled 'parameter' and a text input labeled 'value'. On the right, the response details are shown: 'Status: 200 OK', 'Size: 99 Bytes', and 'Time: 1.08 s'. Below this, a 'Response' tab is selected, showing the JSON response content:

```
1 {
2     "message": "https://images.dog.ceo/breeds/terrier-russell/jack-koda-2.jpg",
3     "status": "success"
4 }
```

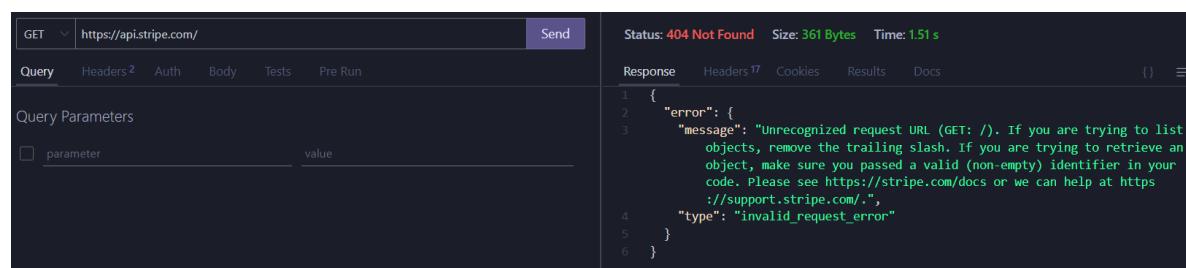
## Api.jsx

```

1 import { useEffect, useState } from "react";
2
3 function Api(){
4     const [perro, setPerro]=useState();
5     const [status, setStatus]=useState();
6
7     useEffect(()=>{
8         listar();
9     },[]);
10
11     async function listar(){
12         const res=await fetch('https://dog.ceo/api/breeds/image/random',{method:'GET'});
13         const respuesta=await res.json();
14         console.log(respuesta);
15         setPerro(respuesta.message);
16         setStatus(respuesta.status);
17     }
18
19     return(
20         <div>
21             <h1>Componente Api</h1>
22             <img src={perro} alt="" />
23             <h2>{status}</h2>
24         </div>
25     )
26 }
27 export default Api;

```

## Thunder client (Visual Studio Code)



The screenshot shows the Thunder client interface with the following details:

- Request URL:** GET https://api.stripe.com/
- Status:** 404 Not Found
- Size:** 361 Bytes
- Time:** 1.51 s
- Response Headers:** Headers, Cookies, Results, Docs
- Query Parameters:** parameter, value
- Response Body:**

```

1 {
2     "error": {
3         "message": "Unrecognized request URL (GET: /). If you are trying to list
4             objects, remove the trailing slash. If you are trying to retrieve an
5             object, make sure you passed a valid (non-empty) identifier in your
6             code. Please see https://stripe.com/docs or we can help at https
7             ://support.stripe.com/.",
8         "type": "invalid_request_error"
9     }
10 }
```

## Api2.jsx

```
1 import { useEffect, useState } from "react";
2
3 function Api2(){
4     const [error, setError]=useState("");
5     const [type, setType]=useState("");
6     useEffect(()=>{
7         traerApi();
8     ,[])
9     async function traerApi(){
10         const res=await fetch('https://api.stripe.com/');
11         const data=await res.json();
12         console.log(data);
13         setError(data.error.message);
14         setType(data.error.type);
15     }
16
17     return(
18         <div>
19             <h1>Componente Api2</h1>
20             <p>{error}</p>
21             <p>{type}</p>
22         </div>
23     )
24 }
25 export default Api2;
```