

# Deep Dive into fetch()



**Christian Wenz**  
Consultant, Software Architect  
[@chwenz](https://twitter.com/chwenz)



# fetch() Options

```
fetch(  
  'api/events',  
  {  
    method: 'GET',  
    headers: {  
      Accept: 'text/xml'  
    }  
  }  
);
```



```
fetch(  
  'api/cart',  
  {  
    method: 'POST',  
    headers: {  
      'Content-Type': 'application/json'  
    },  
    body: JSON.stringify(  
      { id: 42, quantity: 1 }  
    )  
  }  
);
```

## POST Requests



# HTTP Status Codes

**1xx: Informational**

**2xx: Success**

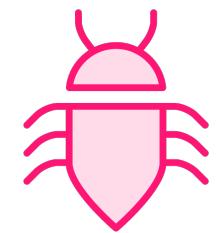
**3xx: Redirection**

**4xx: Client errors**

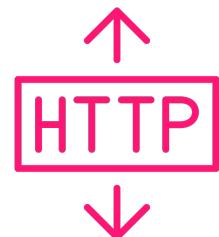
**5xx: Server errors**



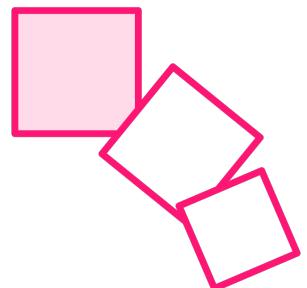
# Error Scenarios



**Error thrown by `fetch()`**



**HTTP status code indicating an error**



**Error while parsing response**



```
const controller =  
  new AbortController();  
  
const signal = controller.signal;  
  
fetch(url, { signal: signal });  
  
controller.abort();
```

◀ Object for aborting requests

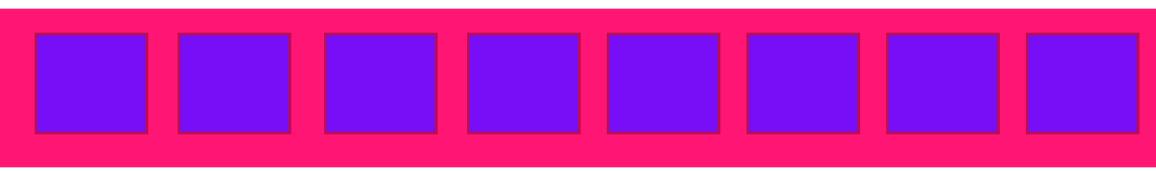
◀ Object for communication with requests

◀ Assign signal (and controller) to request

◀ Abort request



**ReadableStream**



**Stream**

**Data read**



# Cross-Origin Resource Sharing (CORS)



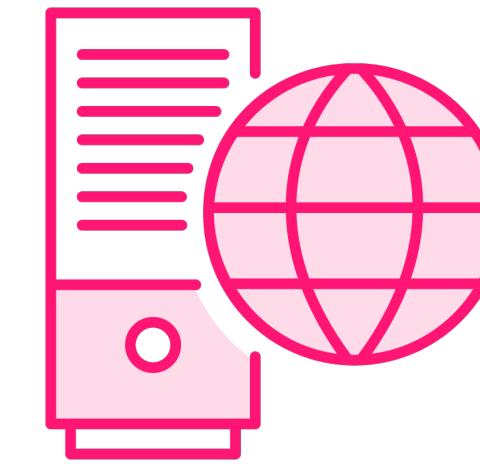
`http://app.example.com`

`GET /api/events HTTP/2  
Host: api.example.com  
Origin: http://app.example.com`



`HTTP/2 200 OK  
Access-Control-Allow-Origin:  
http://app.example.com`

`[{"id":41,"title":"Container  
Enthusiasm",...]`



`http://api.example.com`

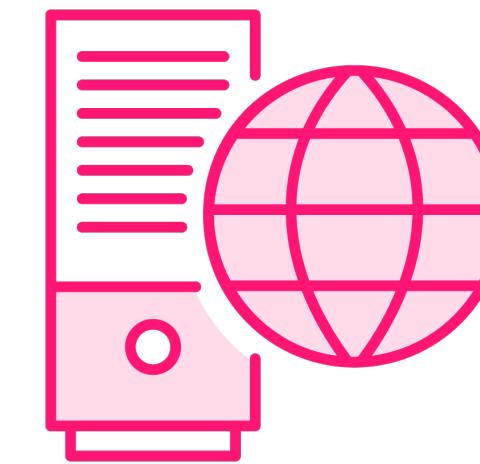


# CORS Preflight Request



<http://app.example.com>

**OPTIONS /api/events HTTP/2**  
**Host: api.example.com**  
**Origin: http://app.example.com**



<http://api.example.com>

**HTTP/2 200 OK**  
**Access-Control-Allow-Origin:**  
**http://app.example.com**  
**Access-Control-Allow-Methods:**  
**GET, POST, OPTIONS**  
**Access-Control-Allow-Headers:**  
**Content-Type, Cookie**



**POST /api/events HTTP/2**  
**Host: api.example.com**



**Up Next:**

# **HTTP Requests with Axios**

---

