HTTP REQUEST & RESPONSE

CLIENT INITIATES REQUEST

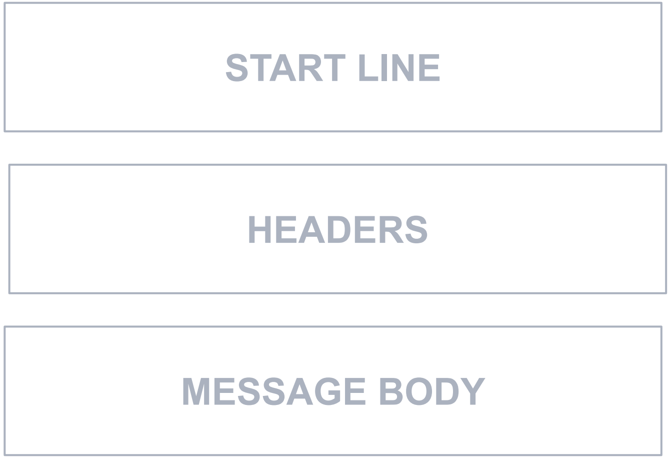
We can initiate client requests with javascript using

XMLHttpRequest

API FETCH API

AXIOS

REQUEST MESSAGE



START LINE

HEADERS

MESSAGE BODY

START LINE CONTAINS

HTTP METHOD

TARGET URL

HTTP VERSION

```
GET / HTTP/1.1
Host: localhost:1234
```

START LINE CONTAINS

HTTP METHOD

TARGET URL

HTTP VERSION

```
GET / HTTP/1.1
Host: localhost:1234
```

START LINE CONTAINS

HTTP METHOD

TARGET URL

HTTP VERSION

```
GET / HTTP/1.1
Host: api.thecatapi.com
```

START LINE CONTAINS

HTTP METHOD

TARGET URL

HTTP VERSION

```
GET / HTTP/1.1
Host: localhost:1234
```

FORM REQUEST

TYPE POST

```
<form id="formdata" action="http://domain.com/products" method="post">
```

TARGET URL

url to the resource that will handle the request

```
<form id="formdata" action="http://domain.com/products" method="post">
```

HTTP Version

url to the resource that will handle the request

```
<form id="formdata" action="http://domain.com/products" method="post">


      <input id=″userName″ type=″text″ name=″userName″ value=″″/>
```

HEADERS

GENERAL HEADERS

REQUEST HEADERS

RESPONSE HEADERS

CONTENT HEADERS

GENERAL HEADERS

Used in by both the request and response

access-control-allow-origin: `*`

cache-control: `post-check=0, pre-check=0`

server: `nginx/1.18.0`

content-type: `application/json; charset=utf-8`

REQUEST HEADERS

information about the resource being fetched.

Accept: `text/html`

Accept-Encoding: `gzip, deflate, br`

Accept-Language: `en-US,en;q=0.9`

Connection: `keep-alive`

Host: `localhost:5000`

RESPONSE HEADERS

Information about the response from the server.

access-control-allow-origin: `*`

content-length: `4616`

content-type: `application/json; charset=utf-8`

date: `Mon, 18 Jan 2021 17:50:58 GMT`

expires: `Tue, 03 Jul 2001 06:00:00 GMT`

CONTENT HEADERS

Describe the encoding and format of the message body

Content-Length: `128`

Content-Language: `en-CA`

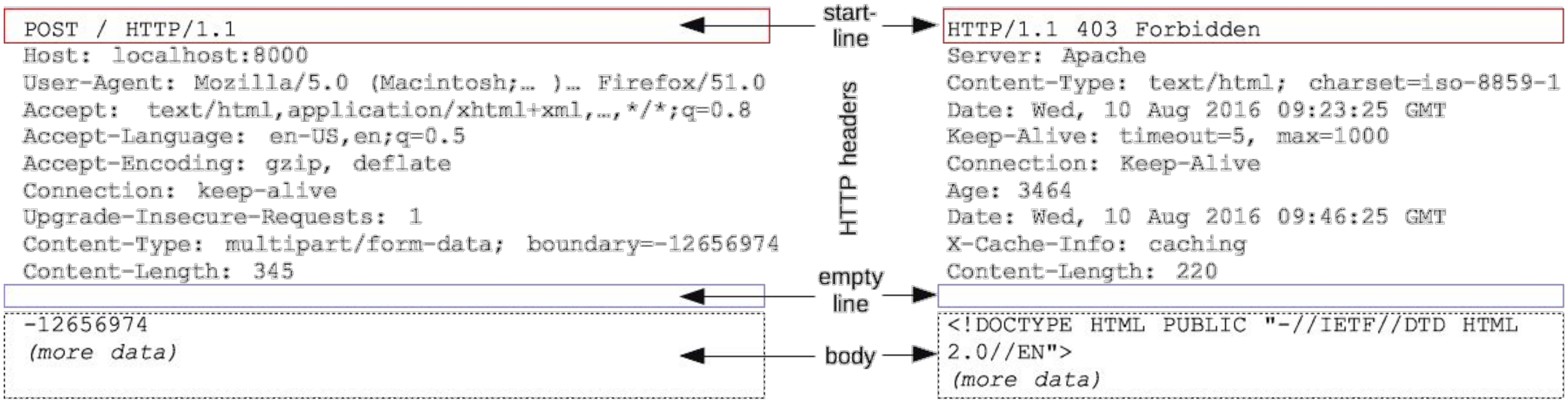
Content-Encoding: `gzip`

Content-Type: `text/html; charset=UTF-8`

HEADER BODY

contains the data to send or data received

REQUEST RESPONSE



```
POST / HTTP/1.1
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;… )… Firefox/51.0
Accept: text/html,application/xhtml+xml,…,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345

-12656974
(more data)
```

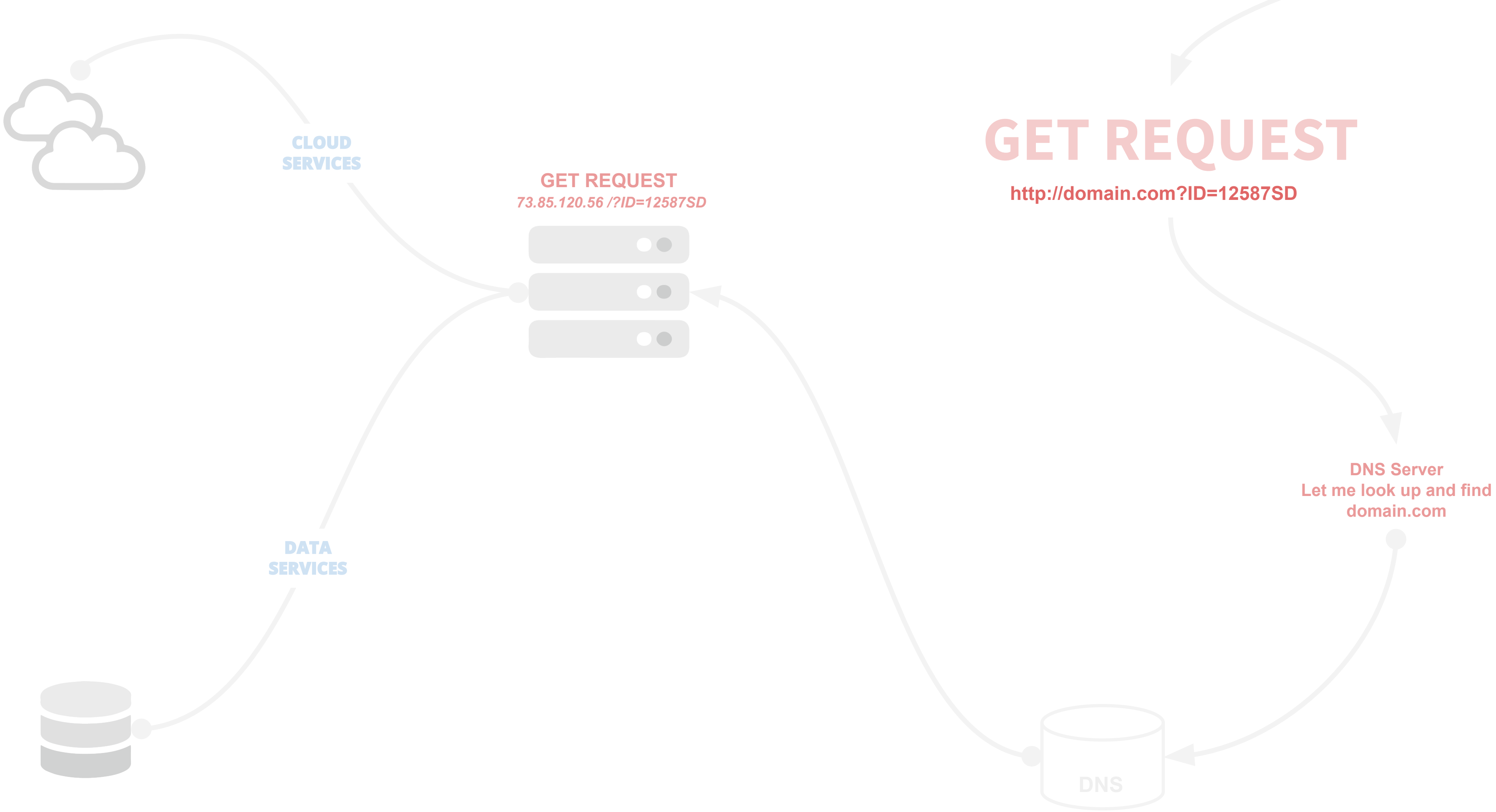
```
HTTP/1.1 403 Forbidden
Server: Apache
Content-Type: text/html; charset=iso-8859-1
Date: Wed, 10 Aug 2016 09:23:25 GMT
Keep-Alive: timeout=5, max=1000
Connection: Keep-Alive
Age: 3464
Date: Wed, 10 Aug 2016 09:46:25 GMT
X-Cache-Info: caching
Content-Length: 220

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML
2.0//EN">
(more data)
```

GET & DELETE

Do not have a message body.

MAKING A GET REQUEST



GET REQUEST

http://domain.com?ID=12587SD

DNS Server
Let me look up and find
domain.com

DNS

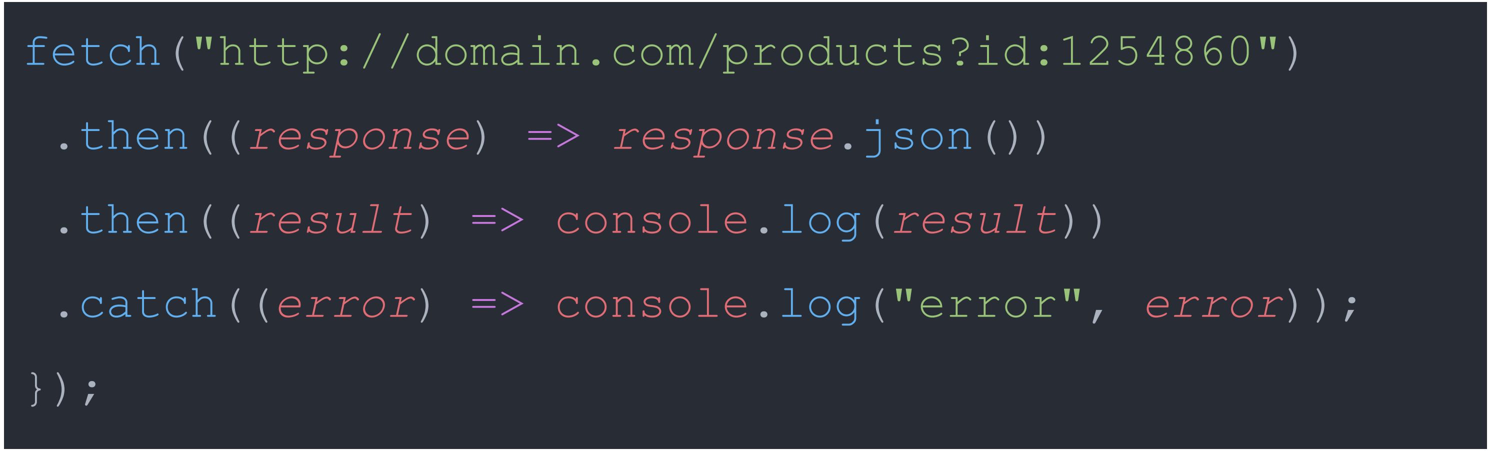
GET REQUEST
73.85.120.56 /?ID=12587SD

CLOUD
SERVICES

DATA
SERVICES

CLIENT REQUEST

made using the fetch api



```
fetch("http://domain.com/products?id:1254860")
 .then((response) => response.json())
 .then((result) => console.log(result))
 .catch((error) => console.log("error", error));
});
```

GET REQUEST

use it to retrieve data.

http://domain.com/api/v2/employees?id=2578412AC

API URL

Path to your application server

http://domain.com/api/v2/employees?id=2578412AC

ENDPOINT

It’s a path on the server that will handle your data request.

http://domain.com/api/v2/employees?id=2578412AC

DATA START FLAG

add ? after the endpoint signals the start of the data name value pairs

http://domain.com/api/v2/employees?id=2578412AC

QUERY STRING

The data your sending appended to the url as name value pairs.

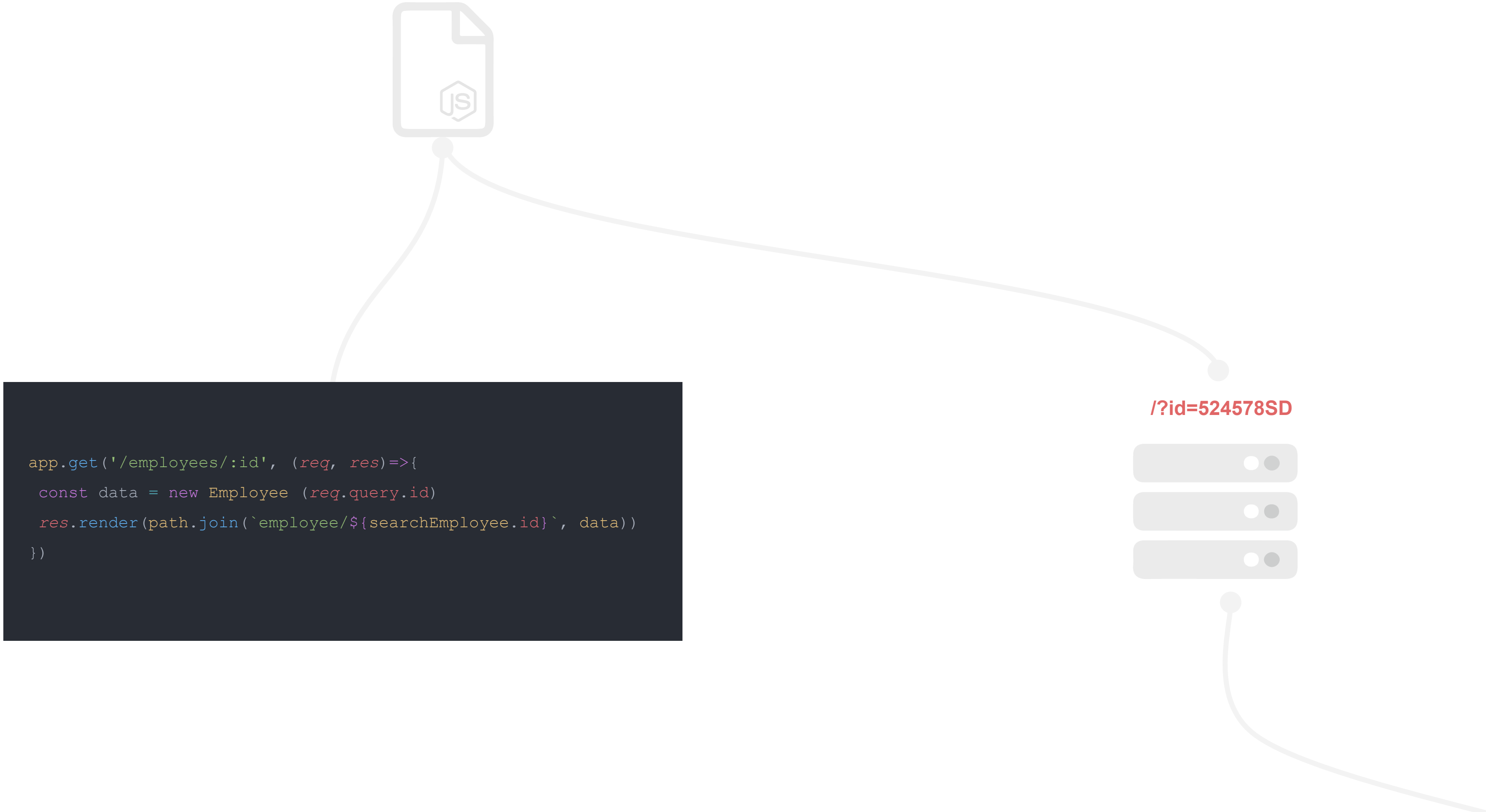
http://domain.com/api/v2/employees?id=2578412AC

NAME VALUE PAIRS

You can send any number of parameters separated by &

/api/v2/employees?department=finance&role=manager

WHAT THE ENDPOINT

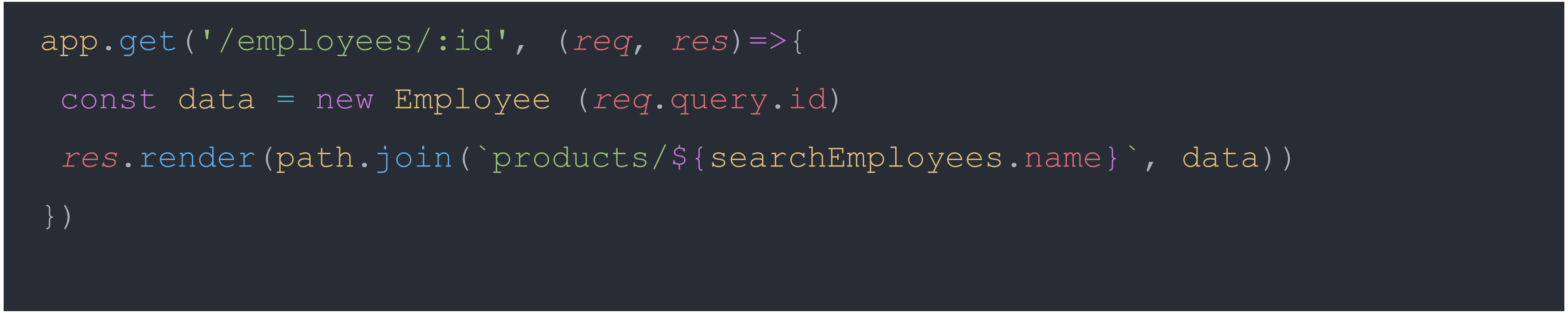


```
app.get('/employees/:id', (req, res)=>{
 const data = new Employee (req.query.id)
 res.render(path.join(`employee/${searchEmployee.id}`, data))
})
```

/?id=524578SD

APPLICATION SERVER

node and express



```
app.get('/employees/:id', (req, res)=>{
 const data = new Employee (req.query.id)
 res.render(path.join(`products/${searchEmployees.name}`, data))
})
```

LET'S WRITE SOME CODE