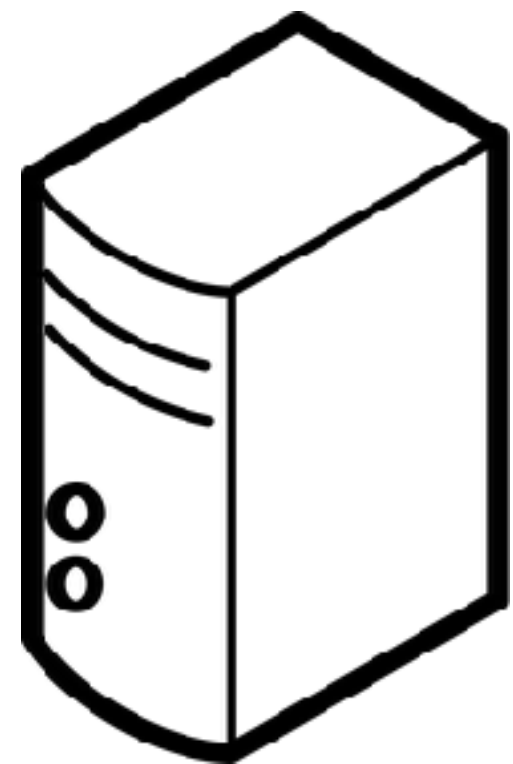
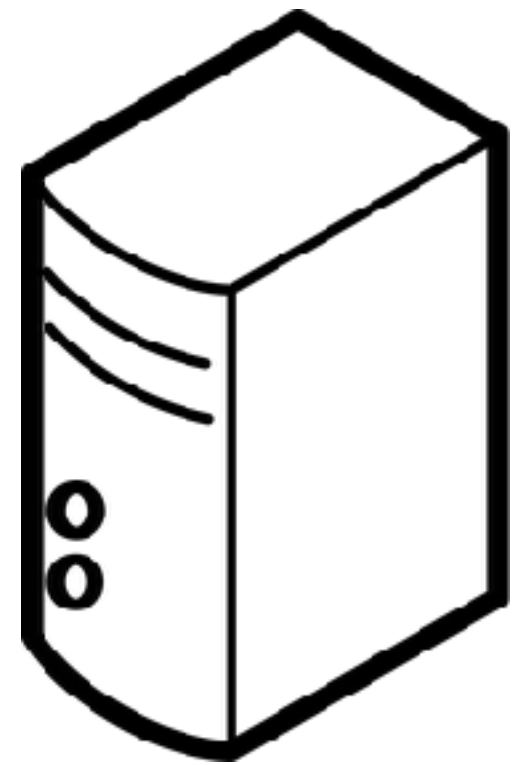


FULL STACK DATA FLOW

There and back again

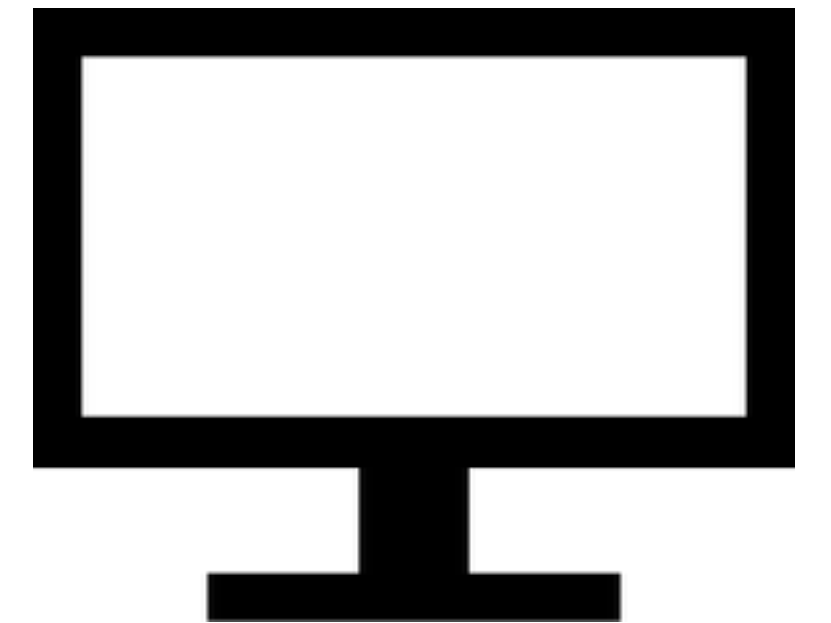




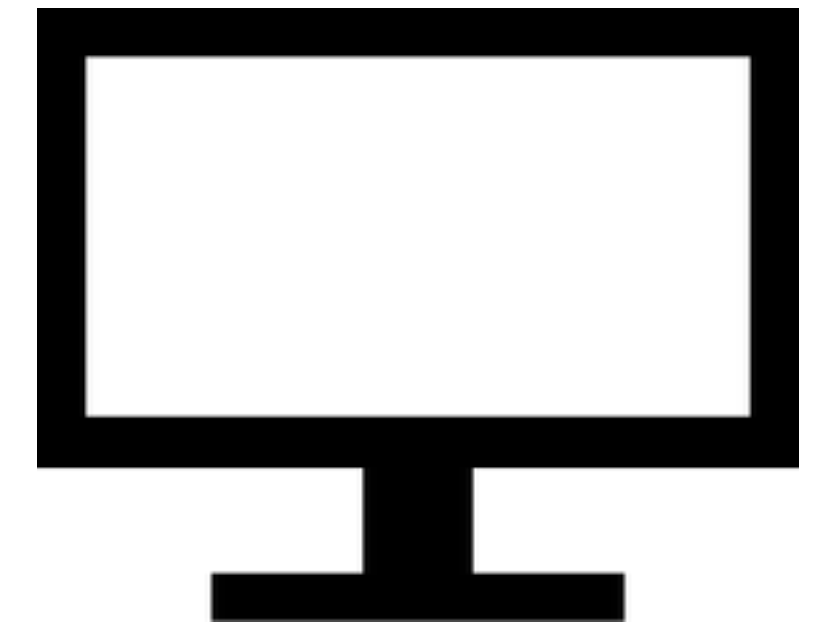
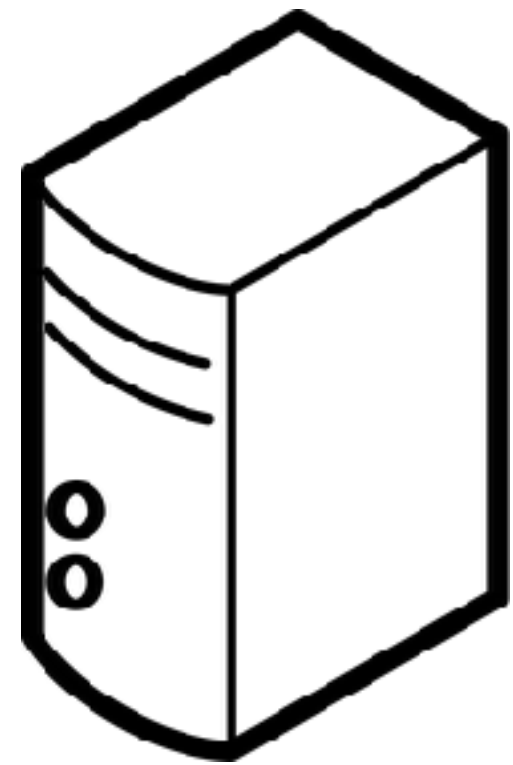
Server



“Internet”



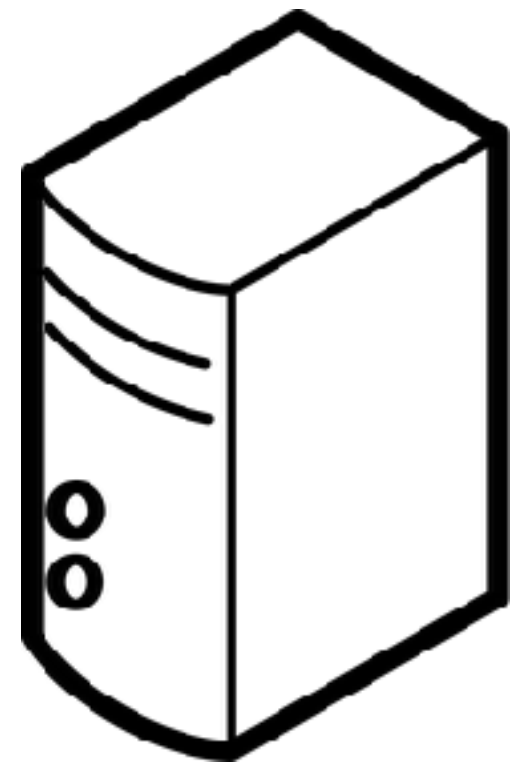
Client



Server

“Internet”

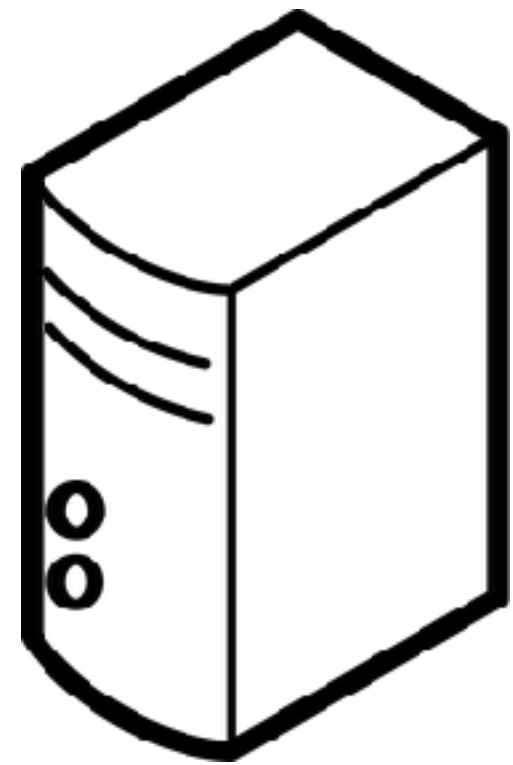
Client



Server

“Internet”

Client

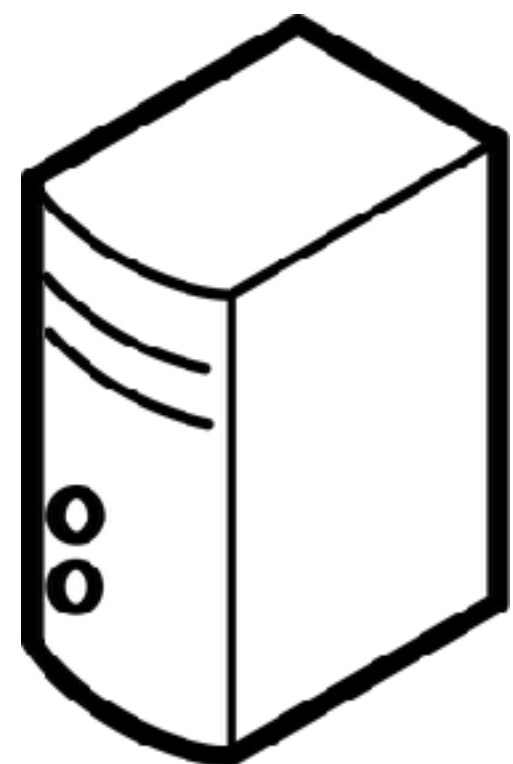


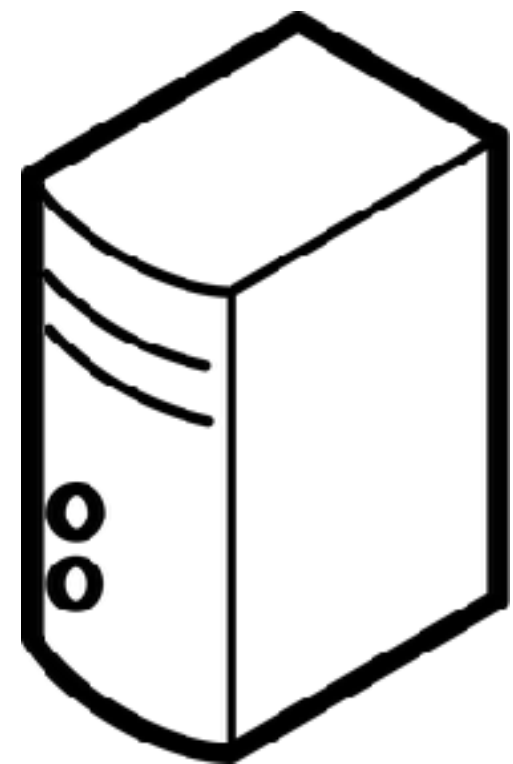
Server



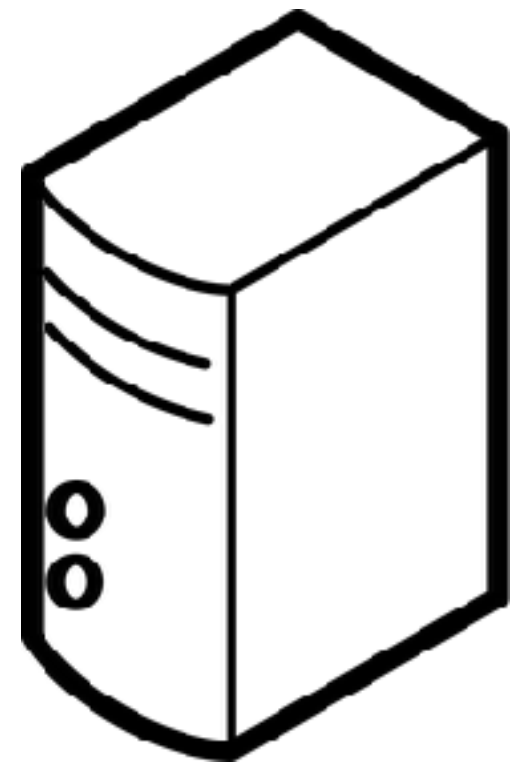
Client

“Internet”



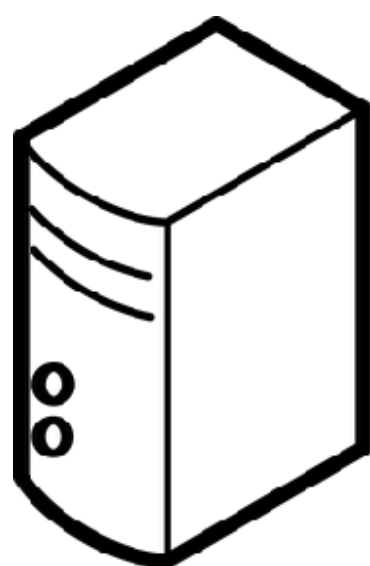


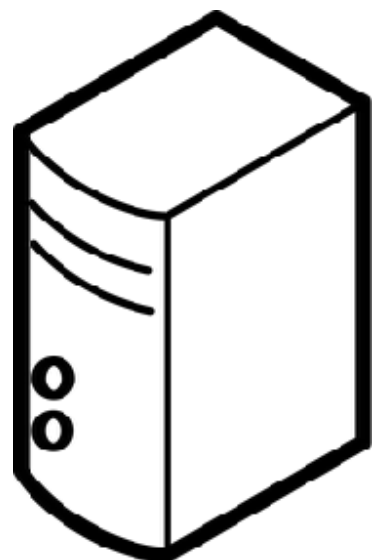
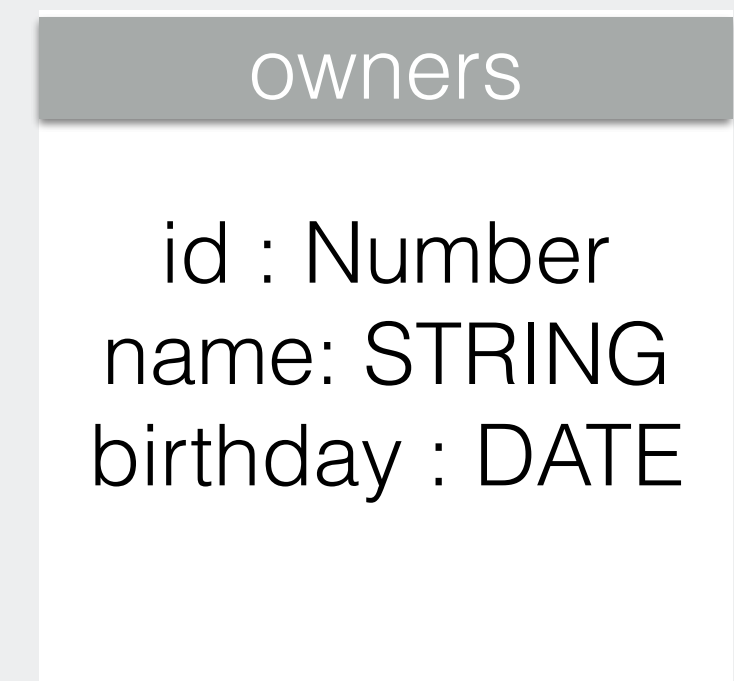
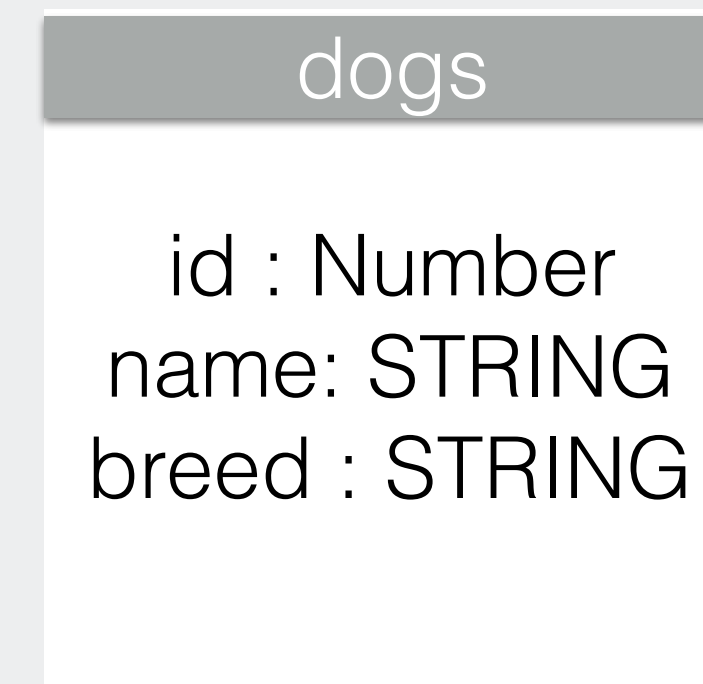
- **A JavaScript runtime built on Chrome's V8 JS engine**
- **Allows execution of JS on an operating system**

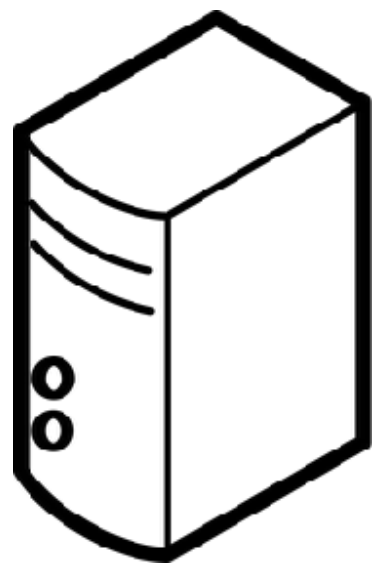


- **An open source, object-relational database system**

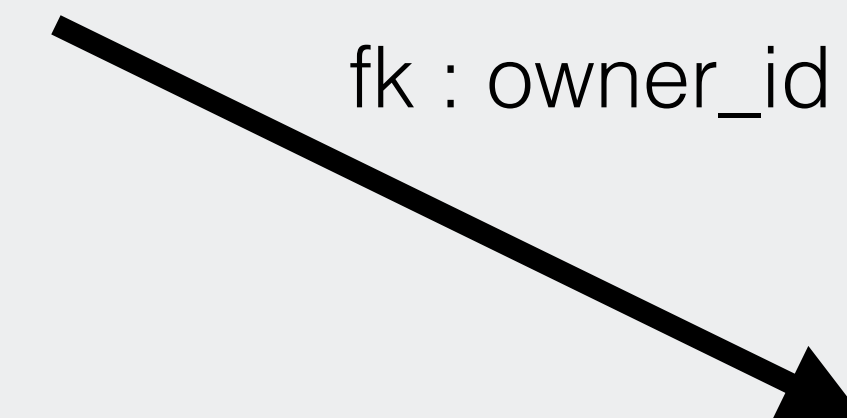








dogs
id : Number name: STRING breed : STRING



owners
id : Number name: STRING birthday : DATE

listening on port 5432



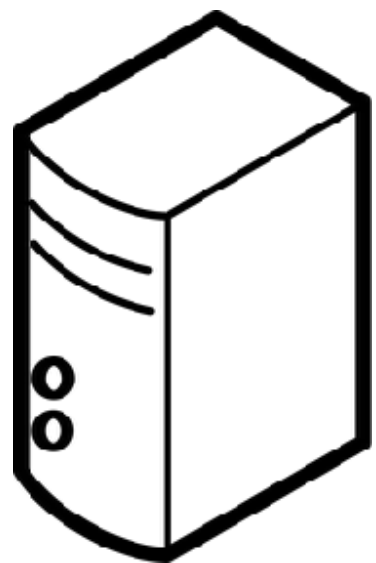
dogs
id : Number name: STRING breed : STRING

owners
id : Number name: STRING birthday : DATE

fk : owner_id

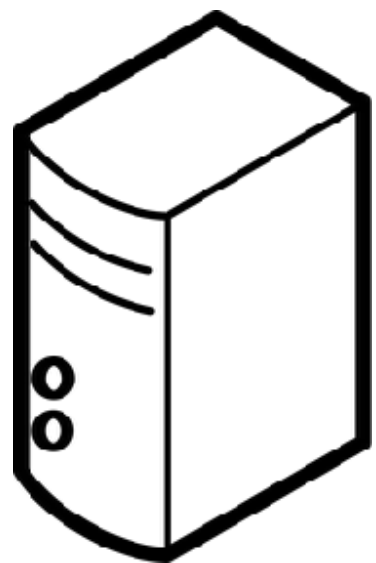
postgres://

listening on port 5432





express



dogs

id : Number
name: STRING
breed : STRING

fk : owner_id

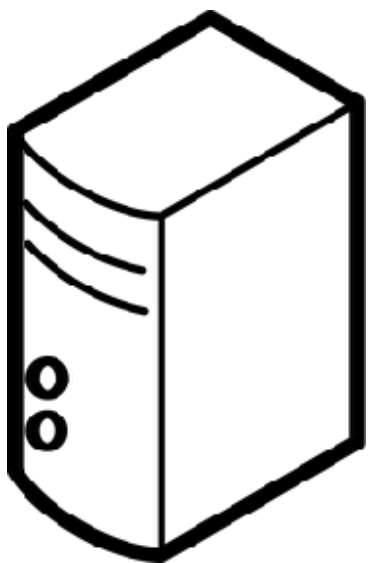
owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

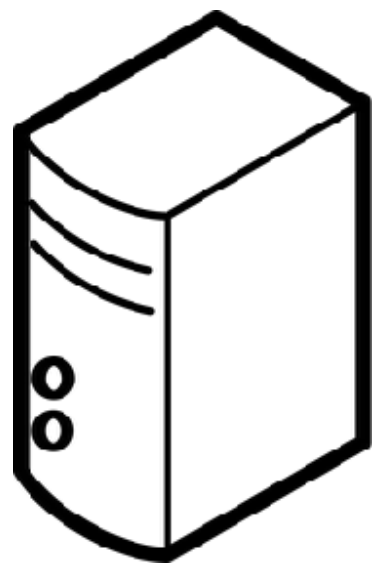
express

- **A application written with Node.js**
- **Provides a framework for writing web servers**





express



dogs

id : Number
name: STRING
breed : STRING

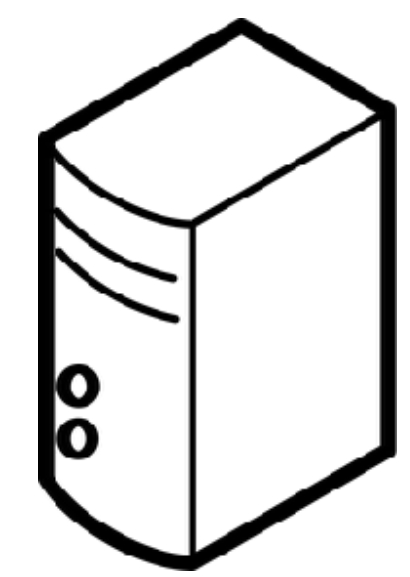
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

server/
index.js



express



dogs

id : Number
name: STRING
breed : STRING

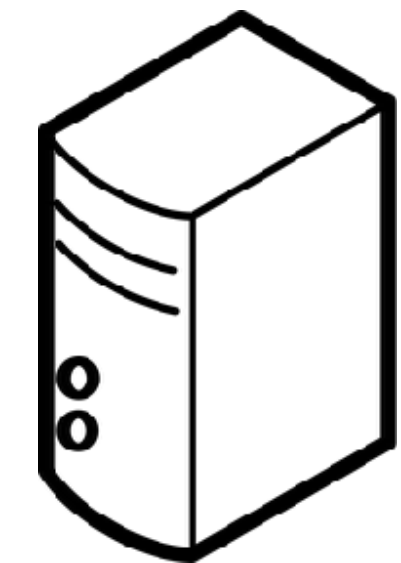


owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

server/
index.js

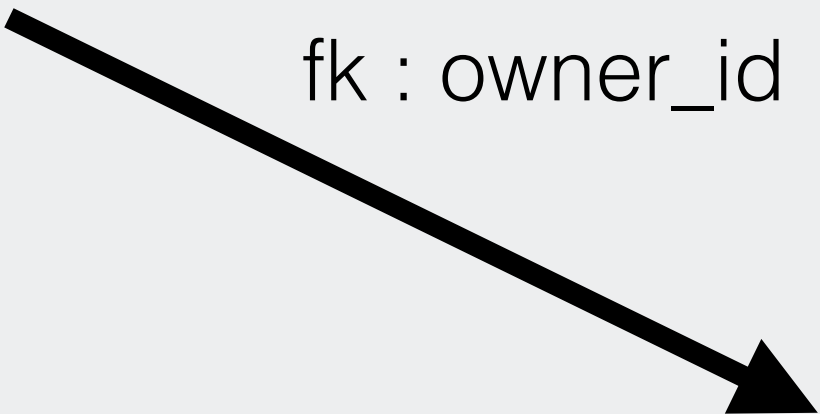


express

listening on port 8080



dogs
id : Number name: STRING breed : STRING



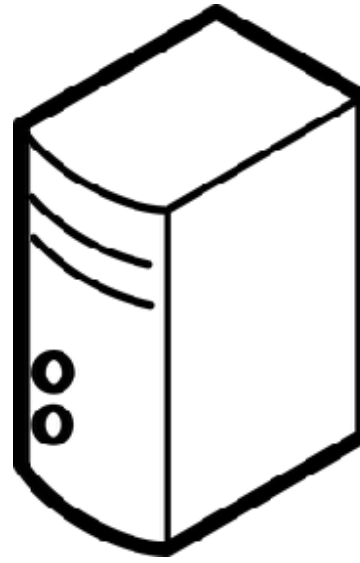
owners
id : Number name: STRING birthday : DATE

listening on port 5432

server/
index.js



express



http://

listening on port 8080



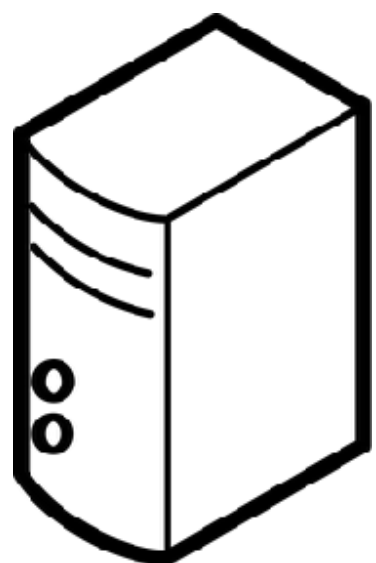
dogs
id : Number name: STRING breed : STRING



owners
id : Number name: STRING birthday : DATE

listening on port 5432

server/
 index.js
public/
 index.html
 style.css



express

listening on port 8080



dogs

id : Number
name: STRING
breed : STRING

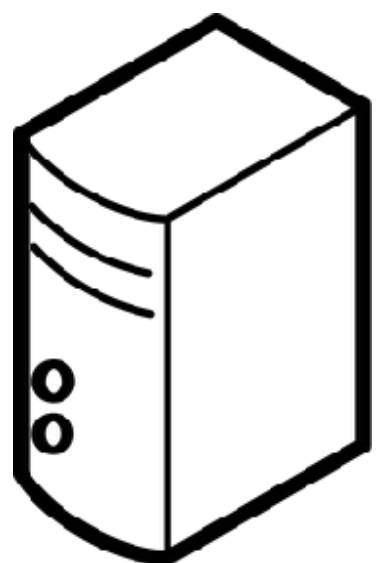
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

server/
index.js
public/
index.html
style.css



express



listening on port 8080



dogs

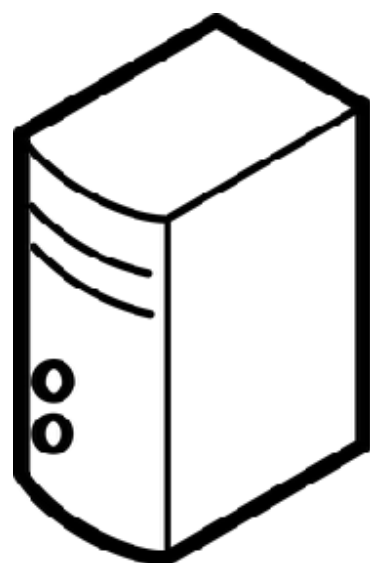
id : Number
name: STRING
breed : STRING

fk : owner_id

owners

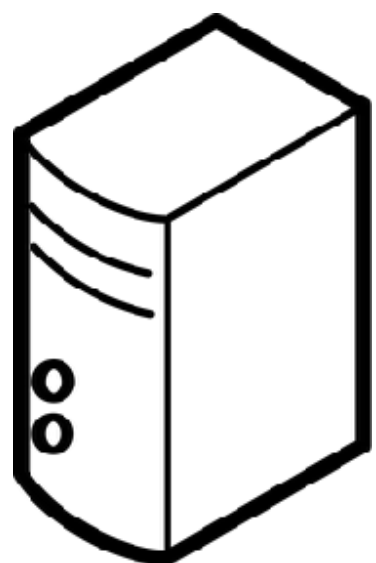
id : Number
name: STRING
birthday : DATE

listening on port 5432



- **A application written with Node.js**
- **Provides a framework for interacting with a database from a Node application**

server/
index.js
public/
index.html
style.css



express



listening on port 8080



dogs

id : Number
name: STRING
breed : STRING

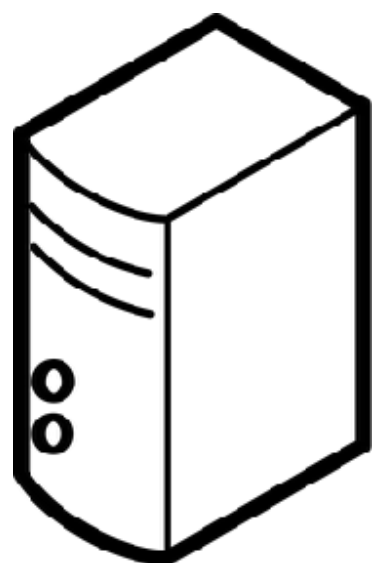
fk : owner_id

owners

id : Number
name: STRING
birthday : DATE

listening on port 5432

server/
 index.js
 db.js
public/
 index.html
 style.css



express



listening on port 8080



dogs

id : Number
name: STRING
breed : STRING

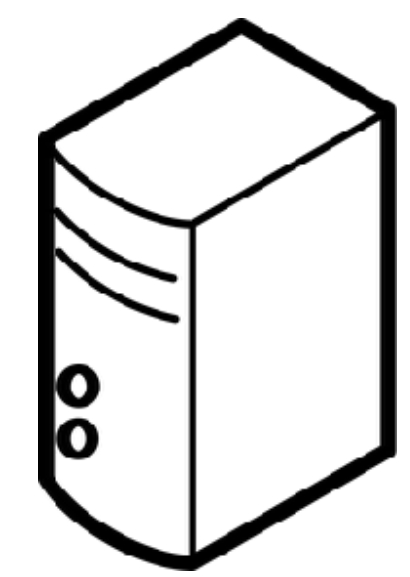
fk : owner_id



owners

id : Number
name: STRING
birthday : DATE


listening on port 5432

server/
index.js
db.js
public/
index.html
style.css





listening on port 8080



dogs
id : Number
name: STRING
breed : STRING

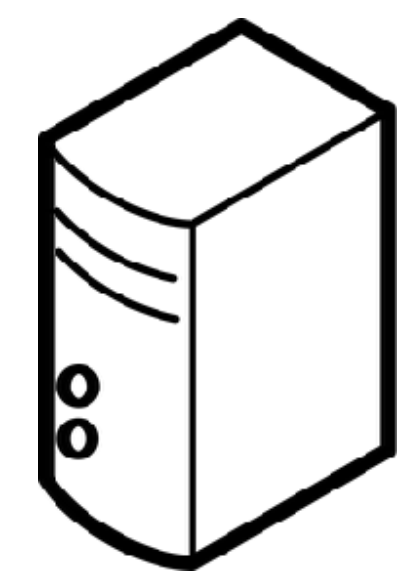
owners
id : Number
name: STRING
birthday : DATE



fk : owner_id

listening on port 5432


server/
index.js
db.js
public/
index.html
style.css

client/
Footer.js
Header.js





listening on port 8080



dogs
id : Number
name: STRING
breed : STRING

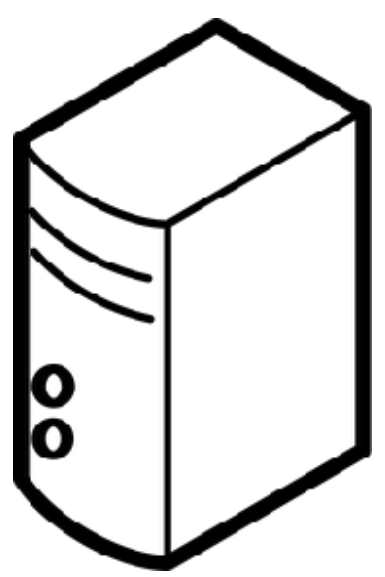
owners
id : Number
name: STRING
birthday : DATE



fk : owner_id

listening on port 5432


server/
index.js
db.js
public/
index.html
style.css

client/
Footer.js
Header.js
 React





listening on port 8080

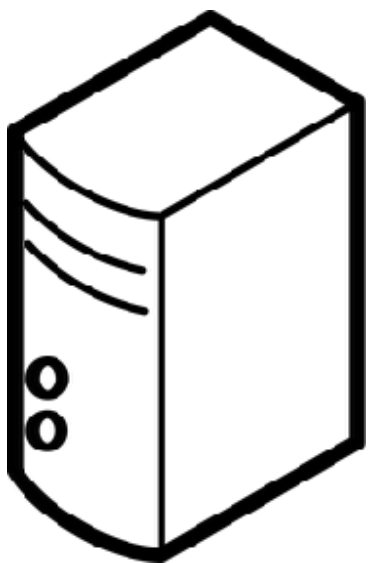


dogs
id : Number
name: STRING
breed : STRING

owners
id : Number
name: STRING
birthday : DATE

fk : owner_id

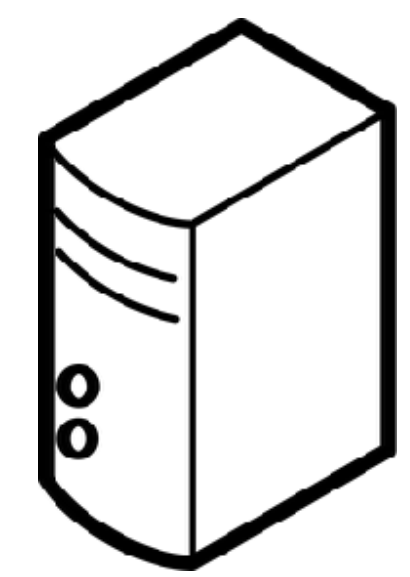
listening on port 5432





- **A JavaScript library for building user interfaces**
- **Executed on a browser (usually)**
- **...But avails itself of special syntax not recognized by browsers (JSX)**


server/
index.js
db.js
public/
index.html
style.css

client/
Footer.js
Header.js
 React





listening on port 8080



dogs
id : Number
name: STRING
breed : STRING

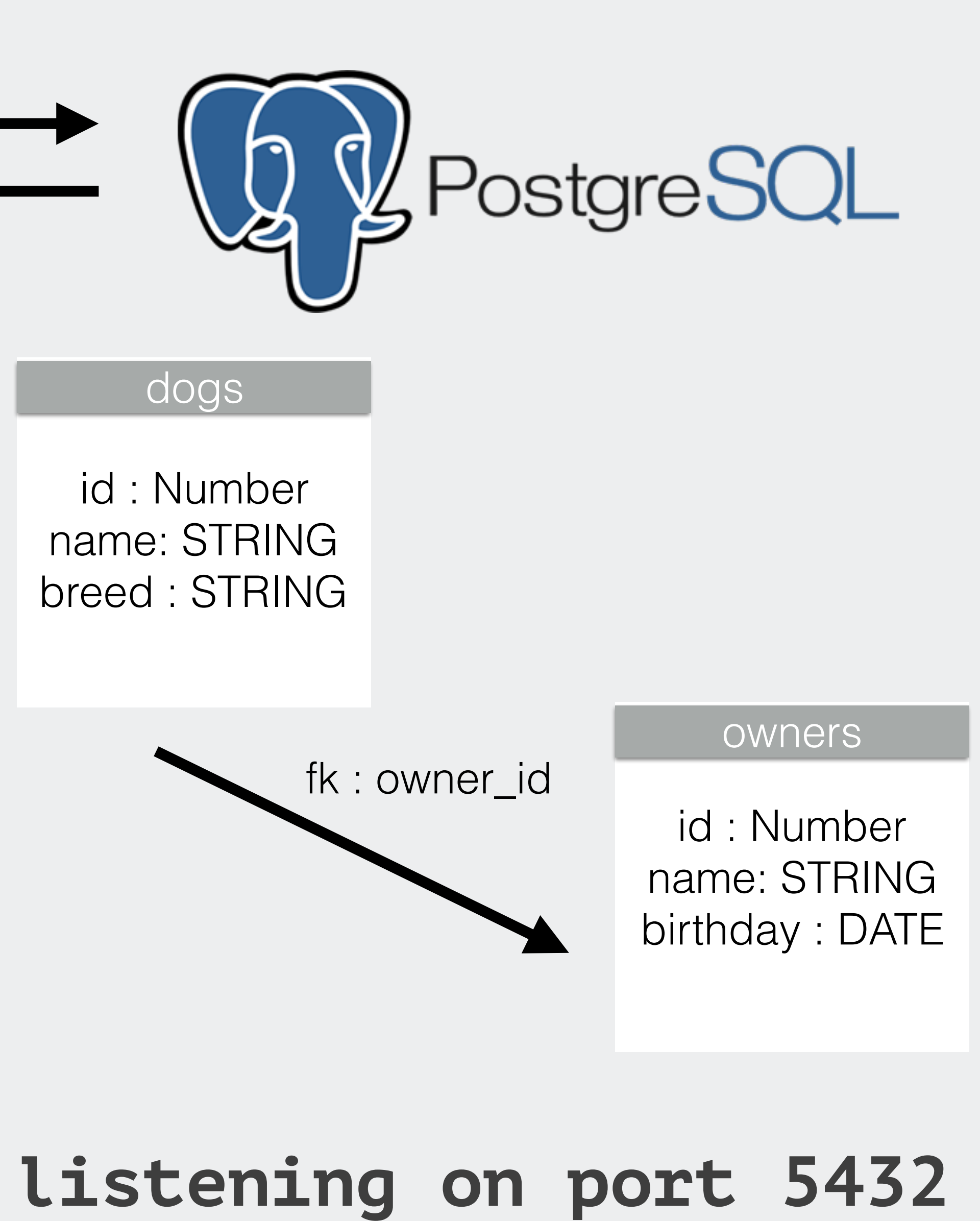
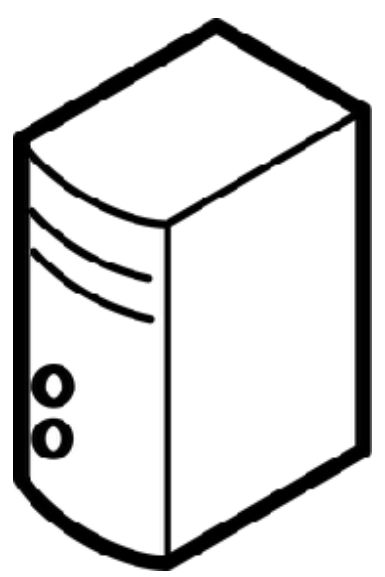
owners
id : Number
name: STRING
birthday : DATE

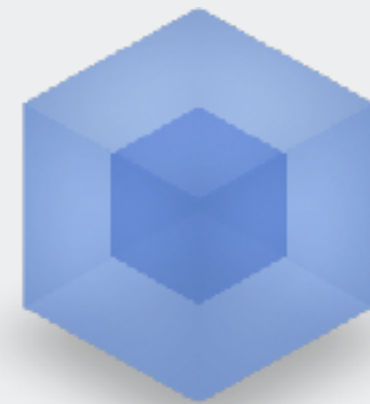
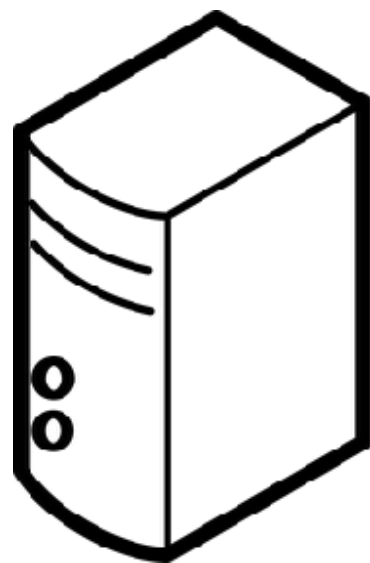
fk : owner_id

listening on port 5432

server/
index.js
db.js
public/
index.html
style.css

client/
Footer.js
Header.js
 React



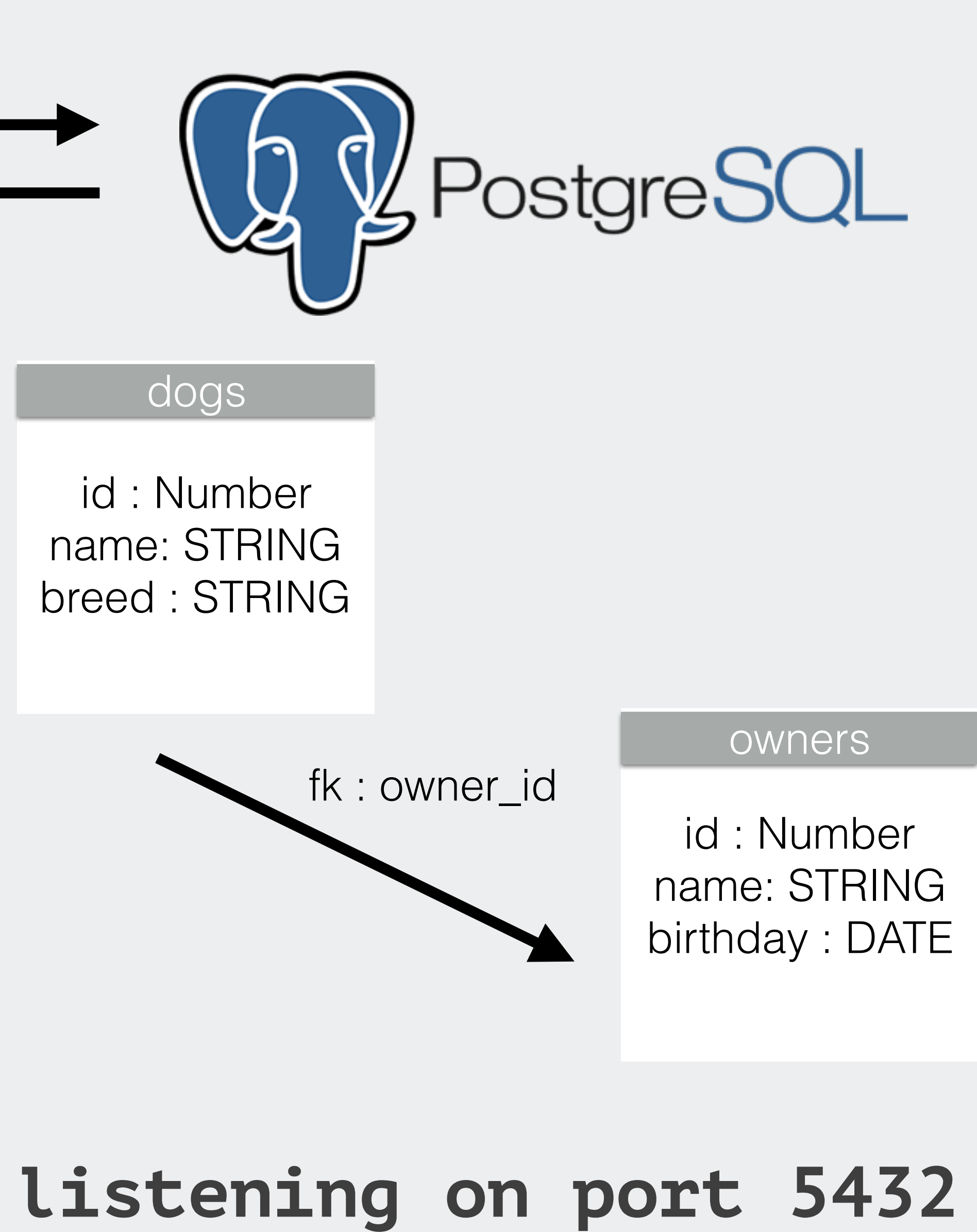
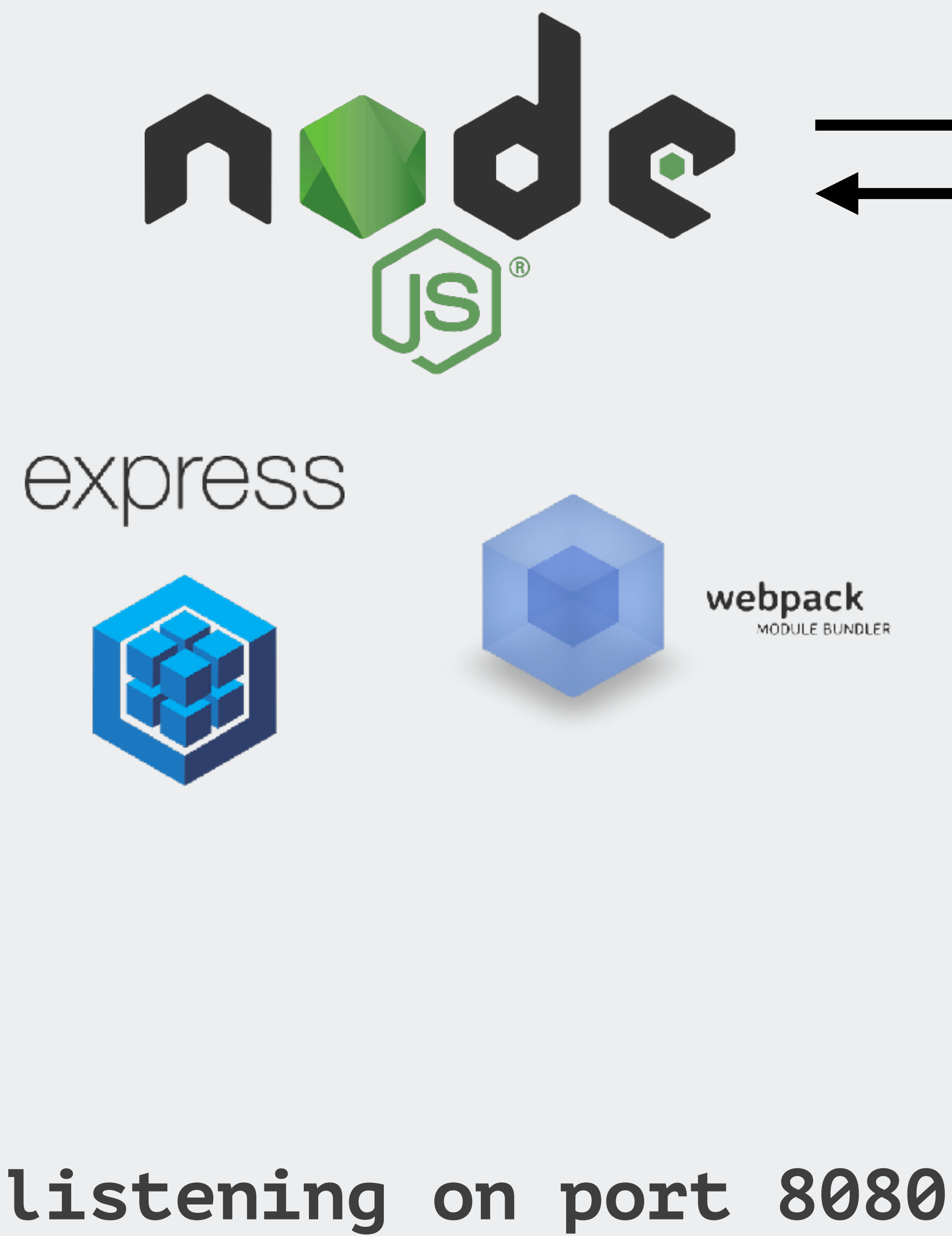
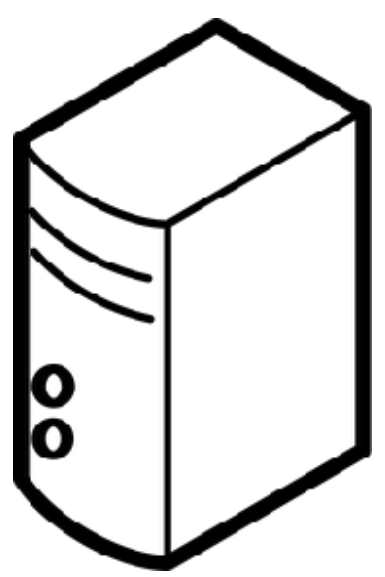


webpack
MODULE BUNDLER

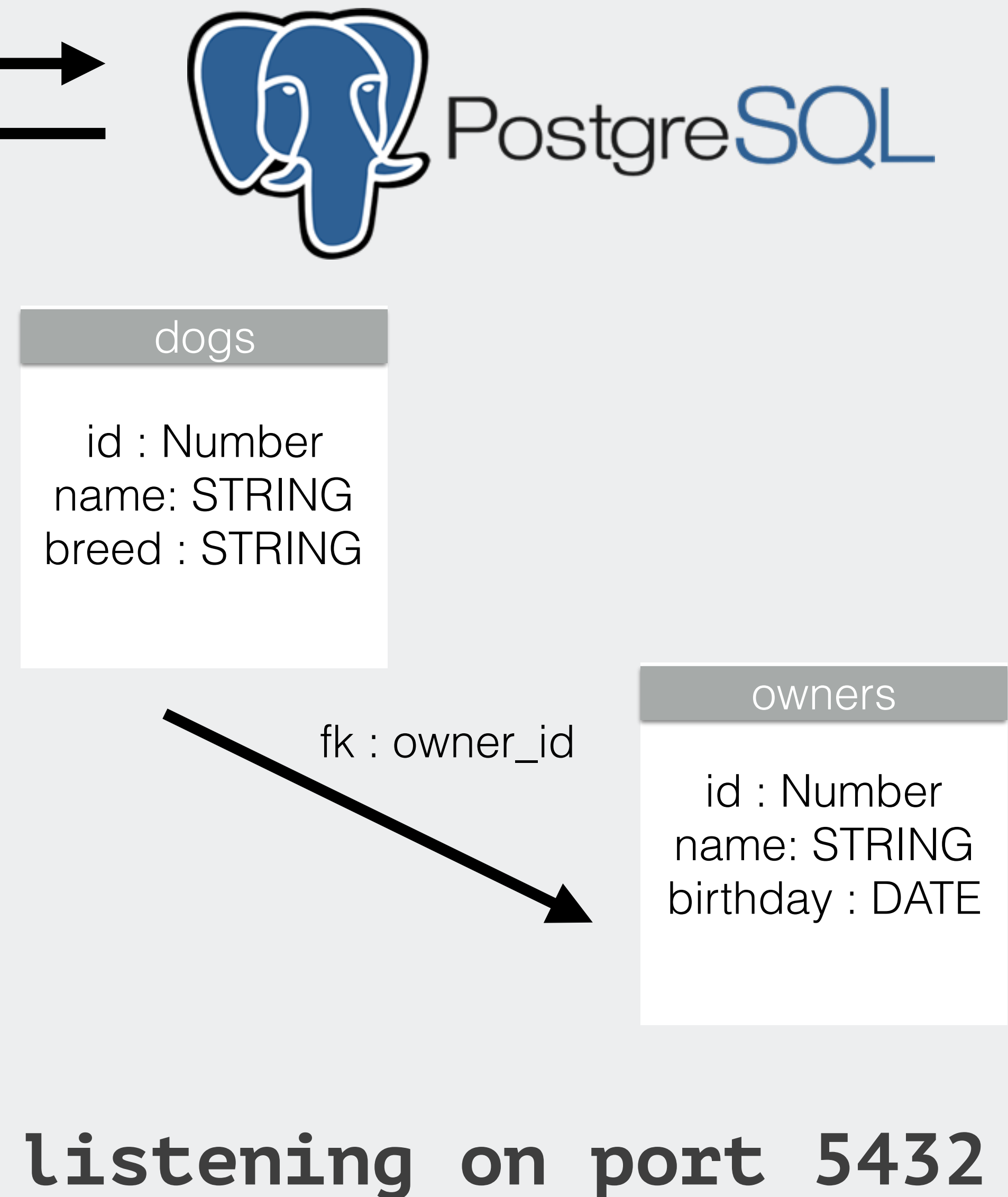
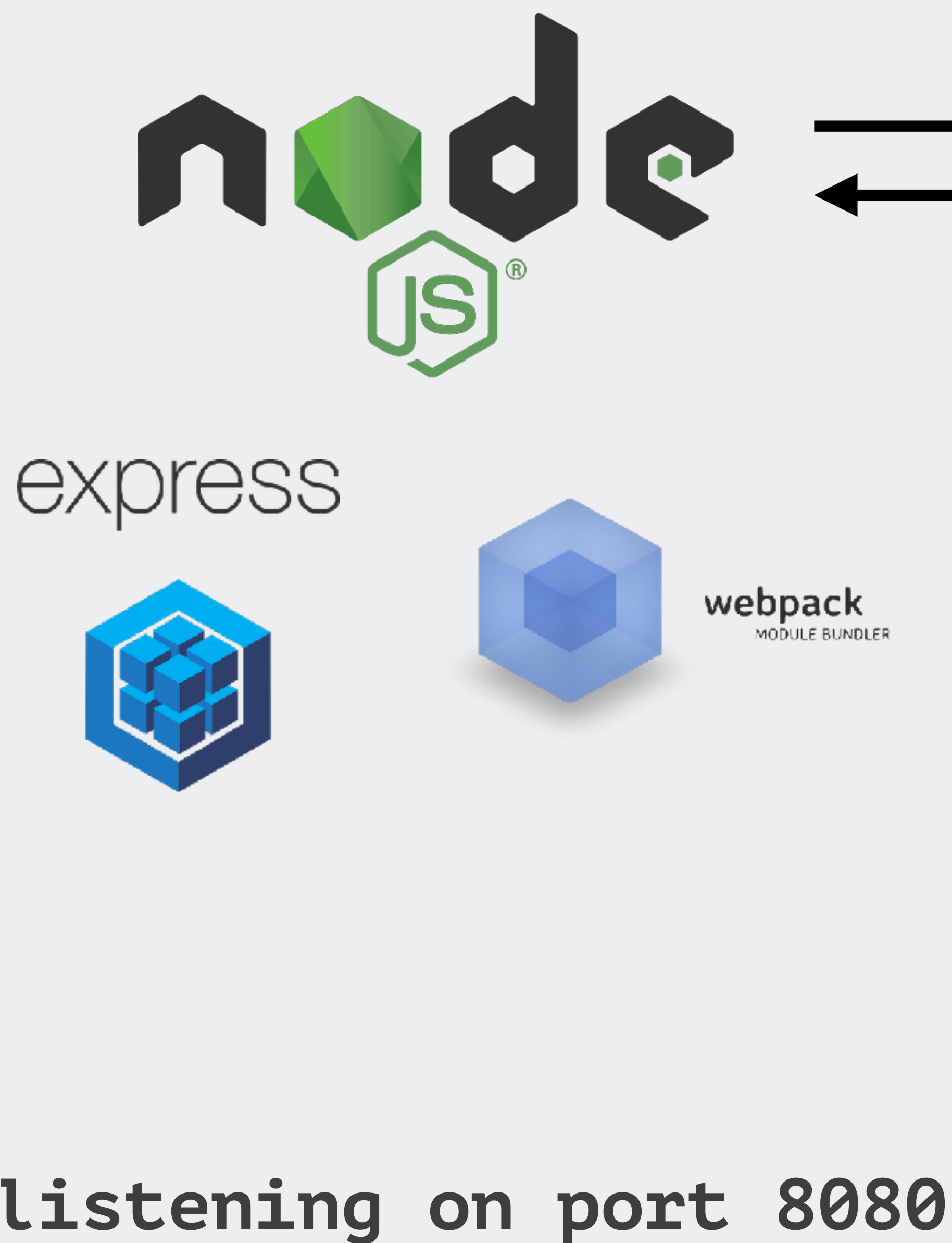
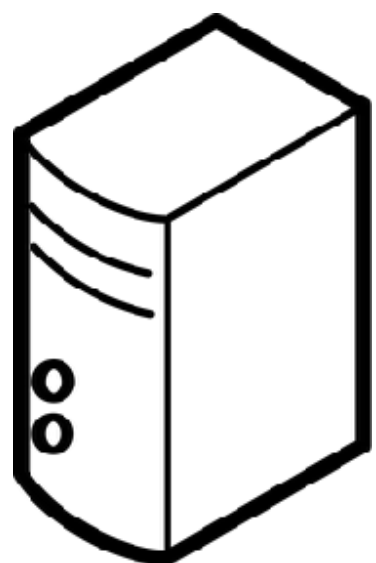
- **A application written with Node.js**
- **Takes many .js files and “bundles” them into one**
- **Compiles special syntax like JSX into browser-safe JS**

server/
index.js
db.js
public/
index.html
style.css

client/
Footer.js
Header.js
 React



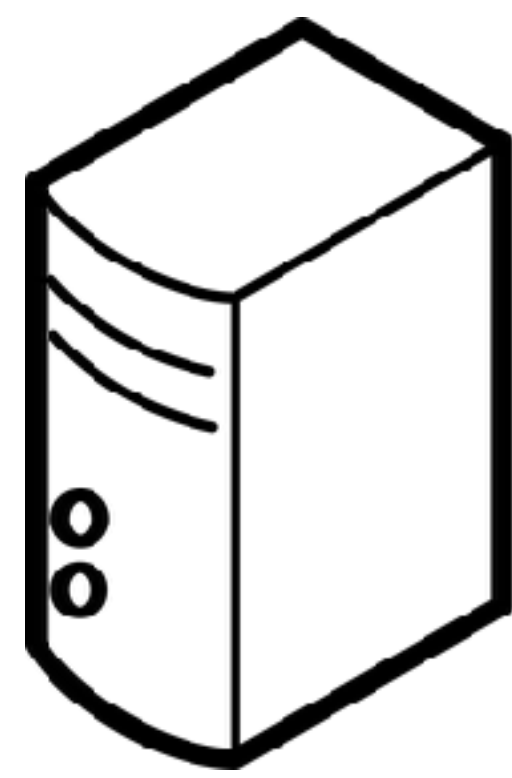
server/
 index.js
 db.js
public/
 index.html
 style.css
 bundle.js
client/
 Footer.js
 Header.js
  React



index.html 

style.css 

bundle.js 



Server

“Internet”



Client

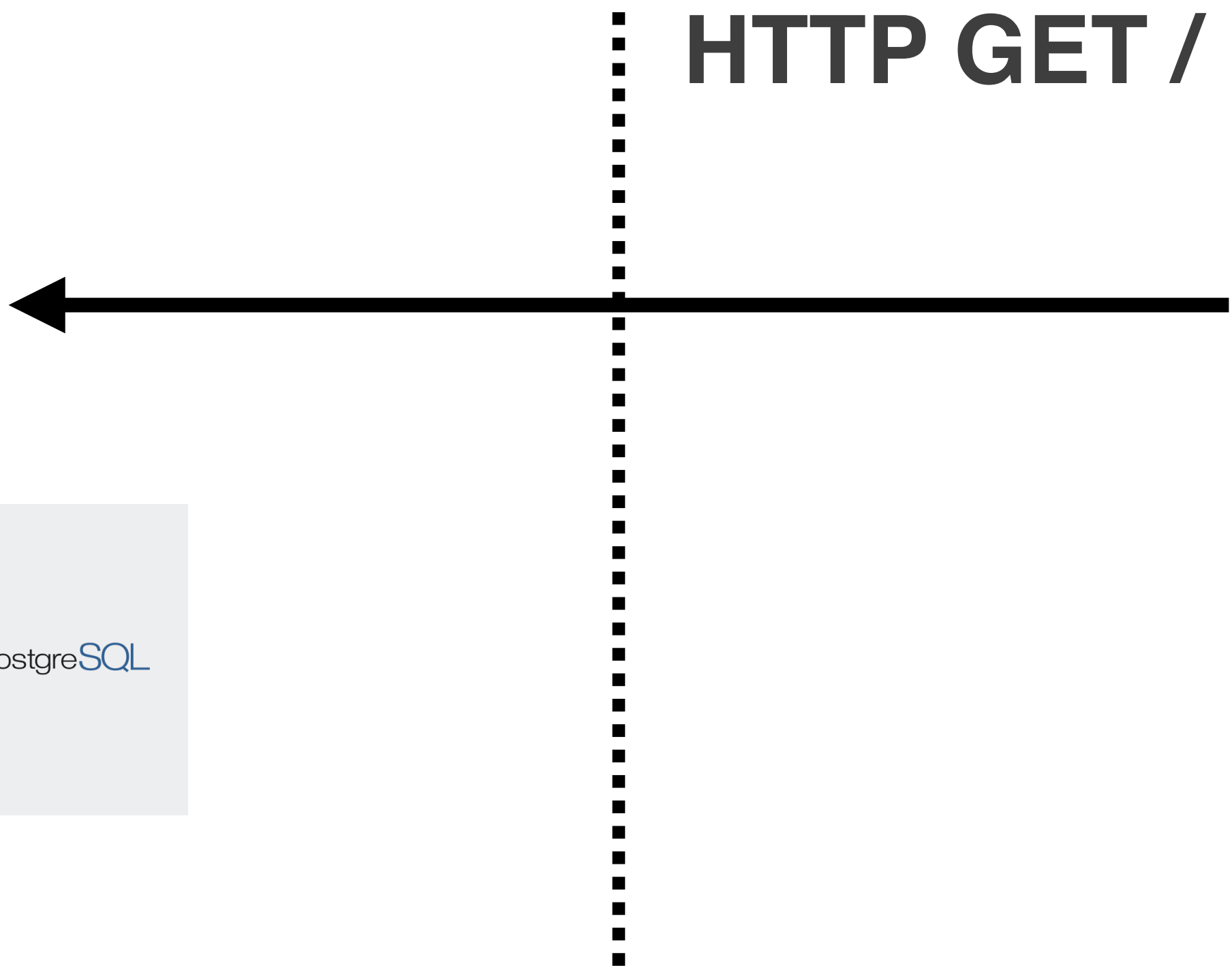
index.html

style.css

bundle.js



Server



Client

index.html

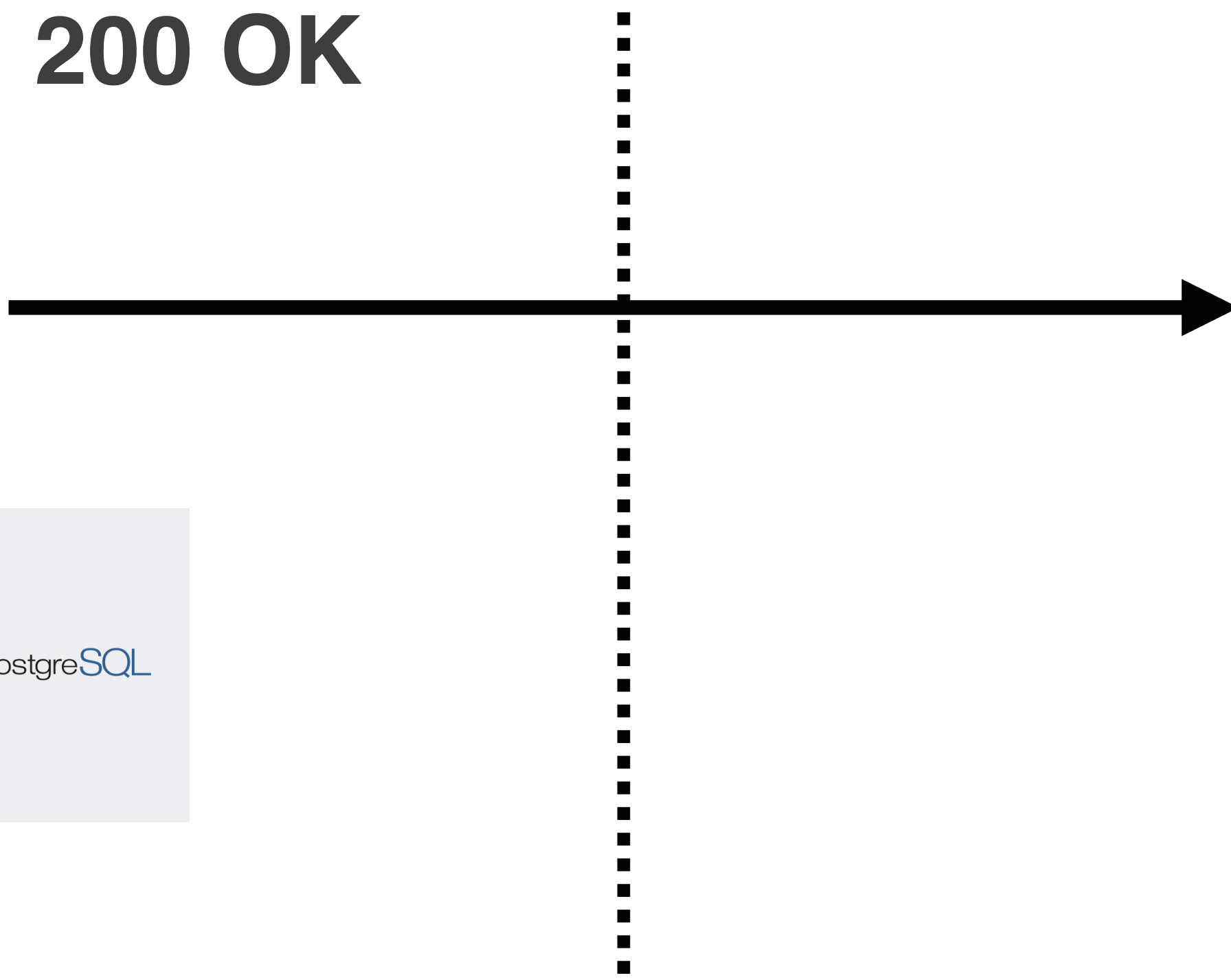
style.css

bundle.js



Server

200 OK



“Internet”



Client



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

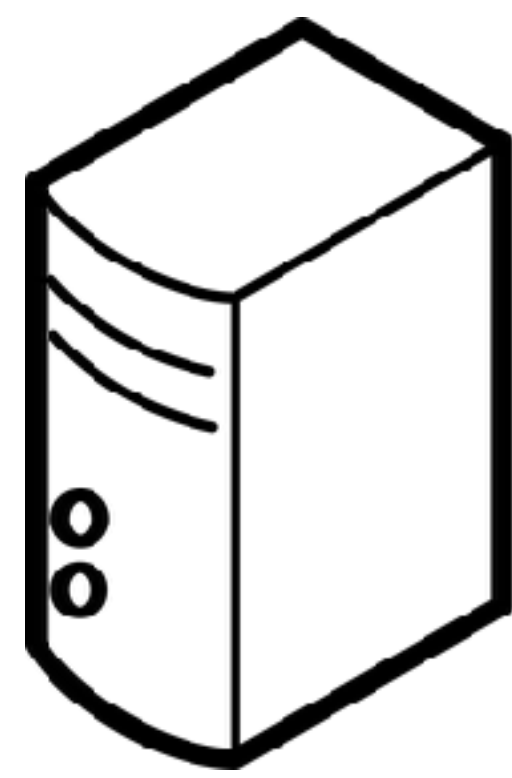



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

index.html 

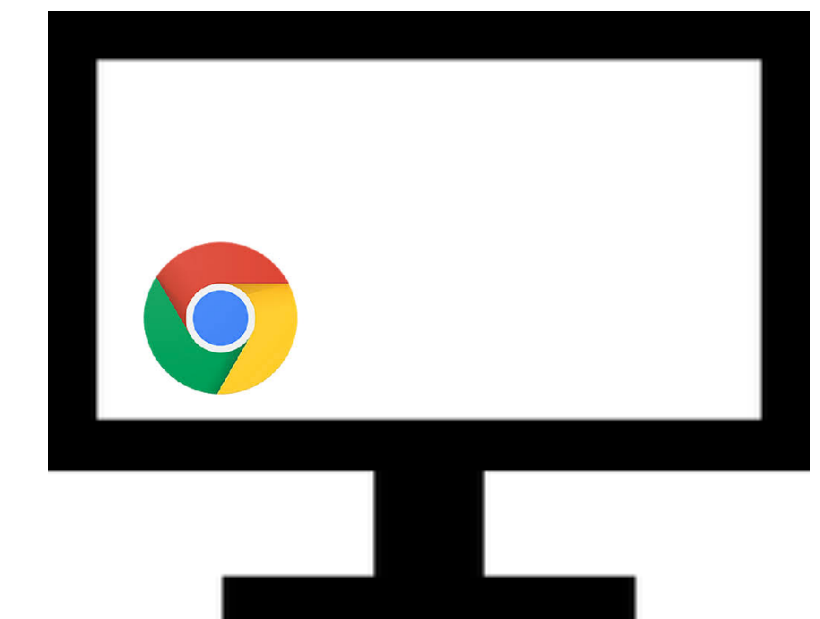
style.css 

bundle.js 



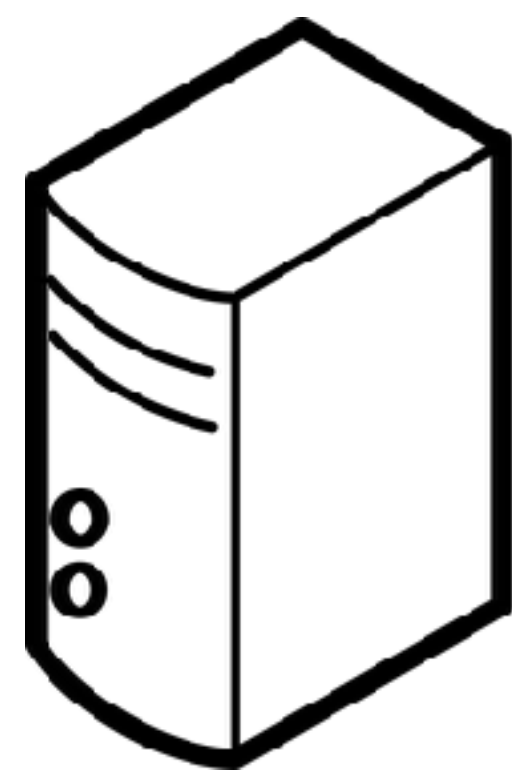
Server

“Internet”



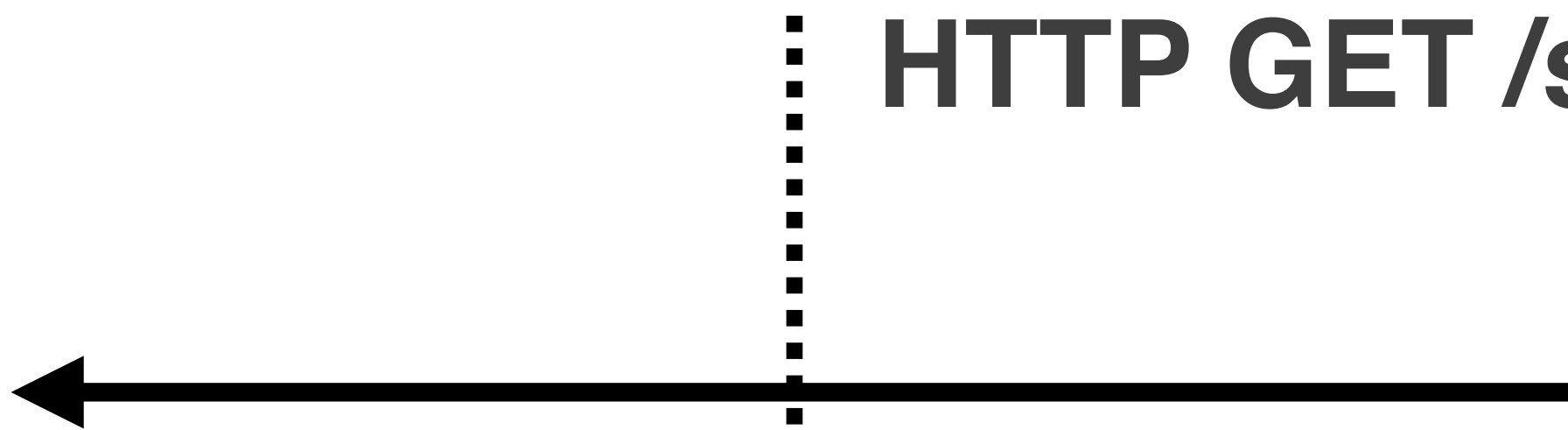
Client

index.html 
style.css 
bundle.js 

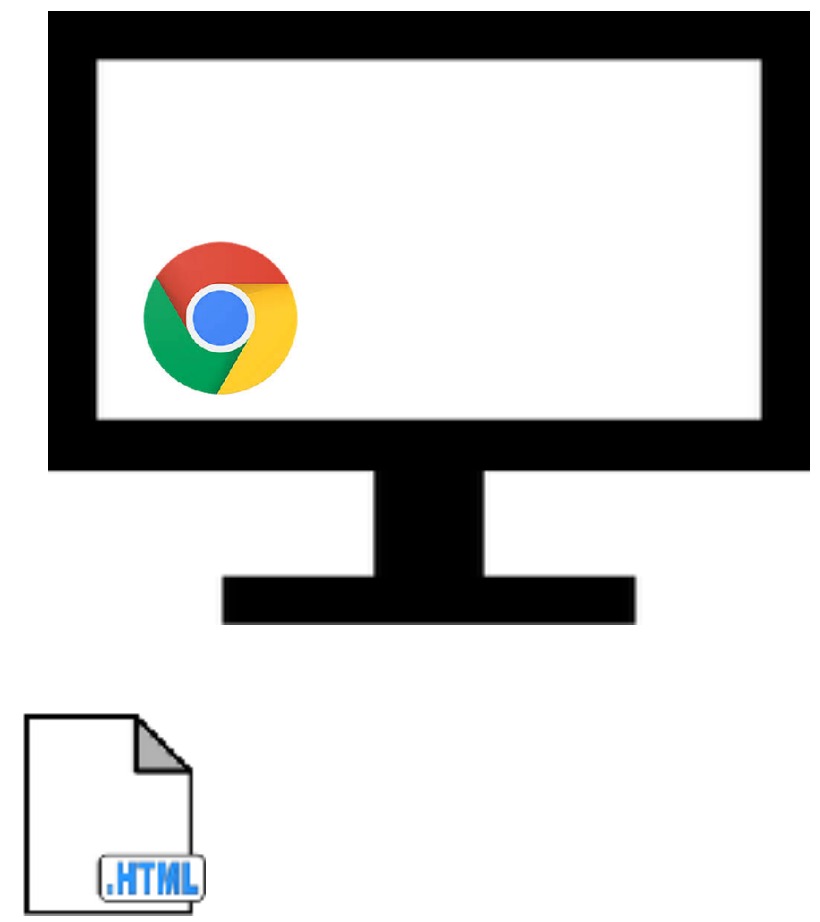


Server

HTTP GET /style.css



“Internet”



Client

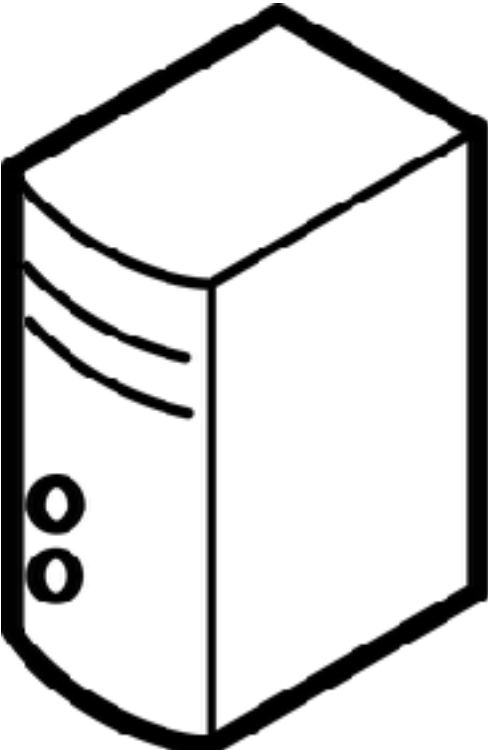
index.html



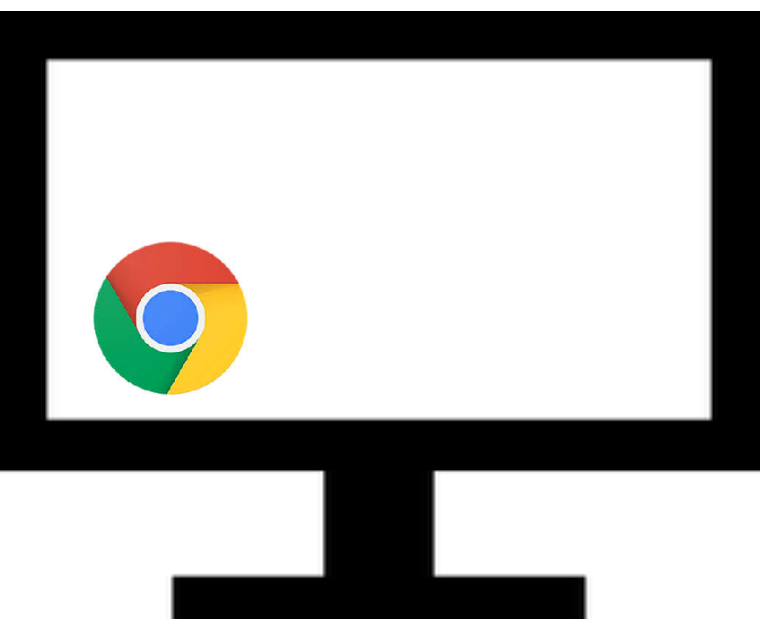
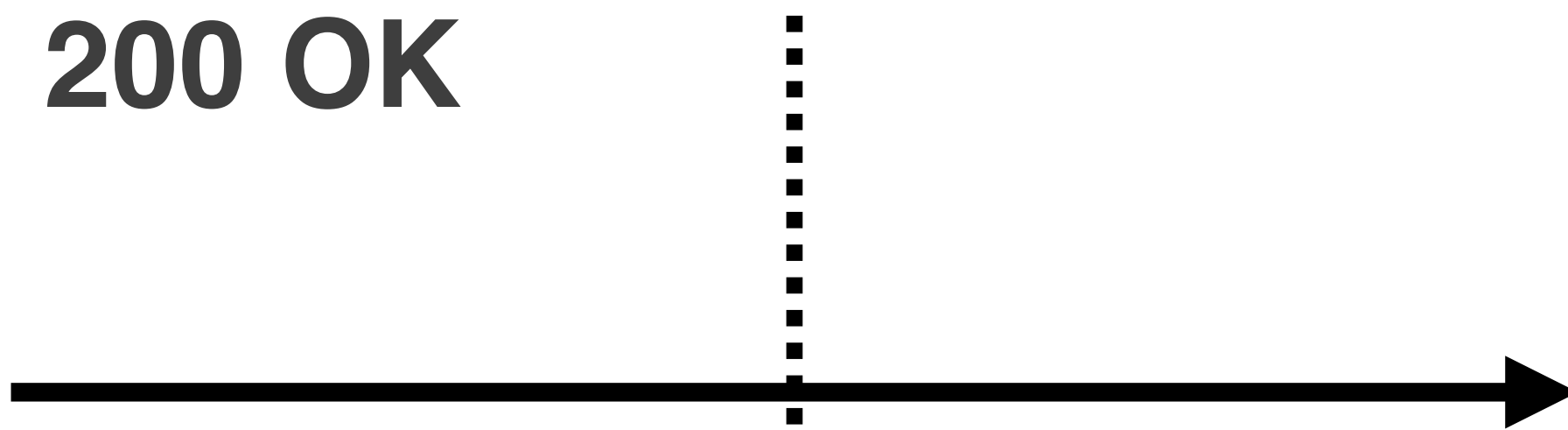
style.css



bundle.js



200 OK



Server

“Internet”

Client



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

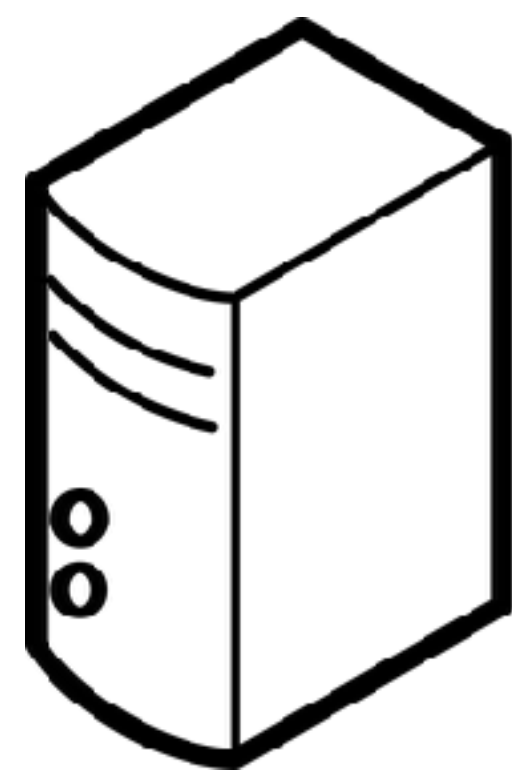


```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

index.html 

style.css 

bundle.js 



Server

“Internet”



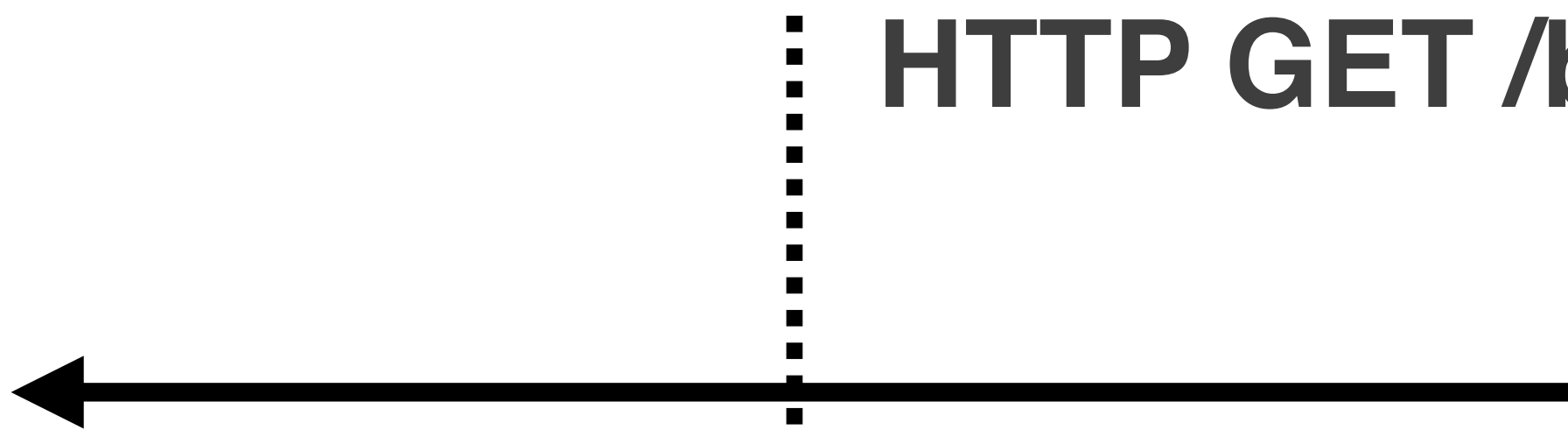
Client

index.html
style.css
bundle.js

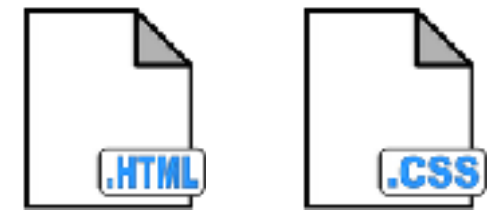


Server

HTTP GET /bundle.js



“Internet”



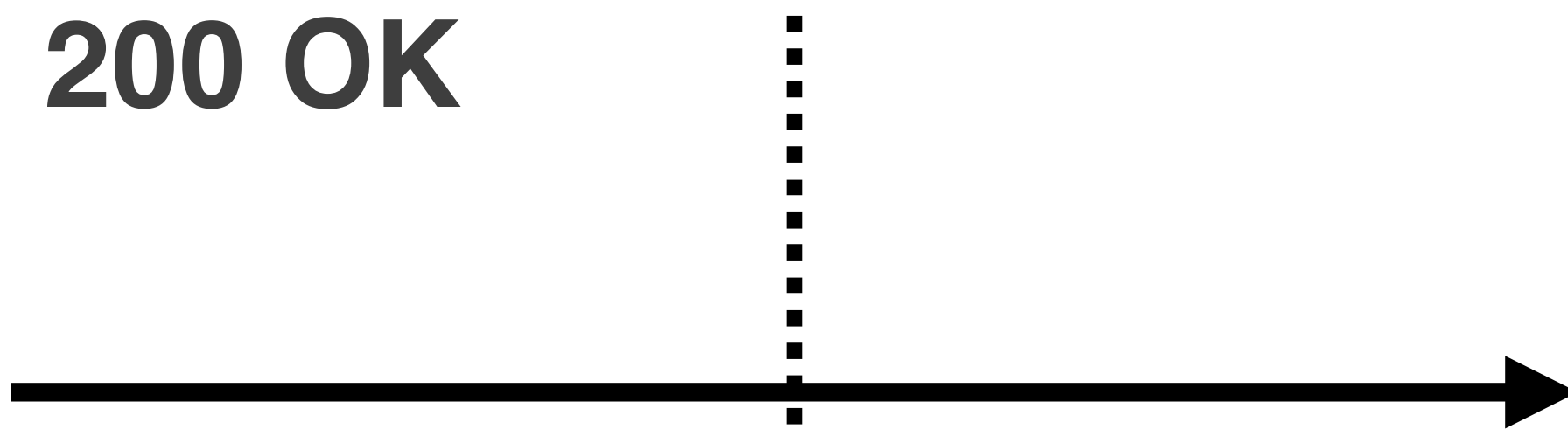
Client

index.html
style.css
bundle.js

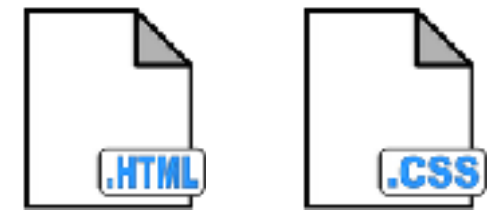


Server

200 OK



“Internet”



Client



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

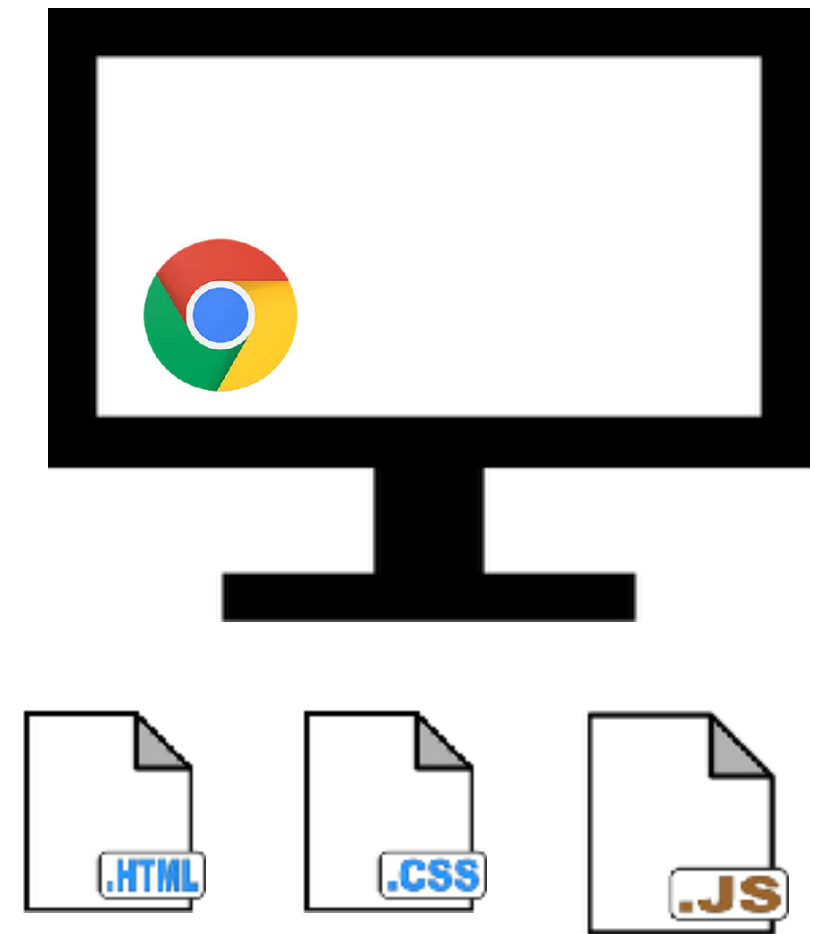
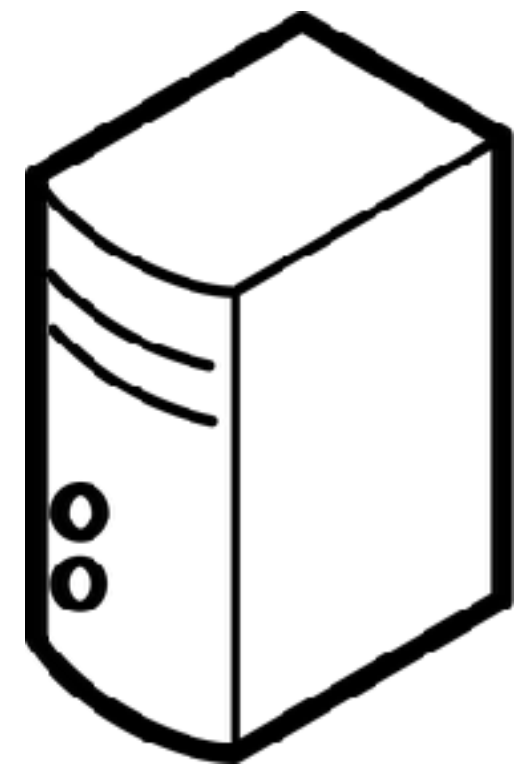


```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```




```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="/style.css" />
    <script defer src="/bundle.js"></script>
  </head>
  <body>
    <div id="app"></div>
  </body>
</html>
```

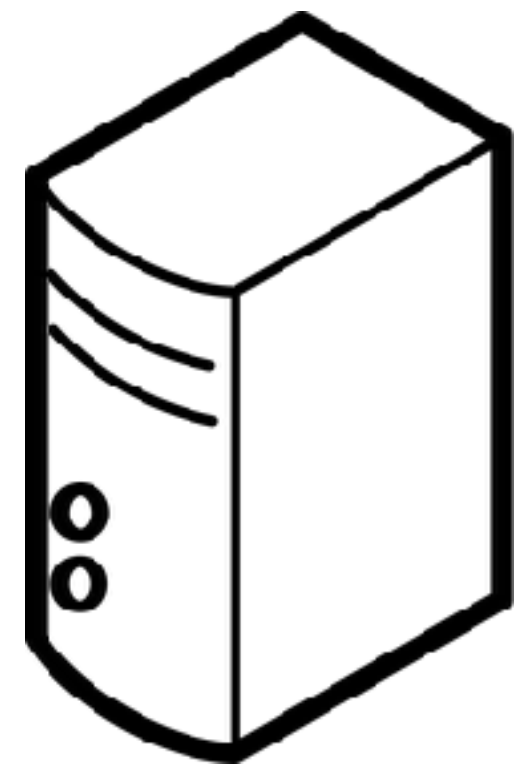
Time to execute this bundle.js file!



Server

“Internet”

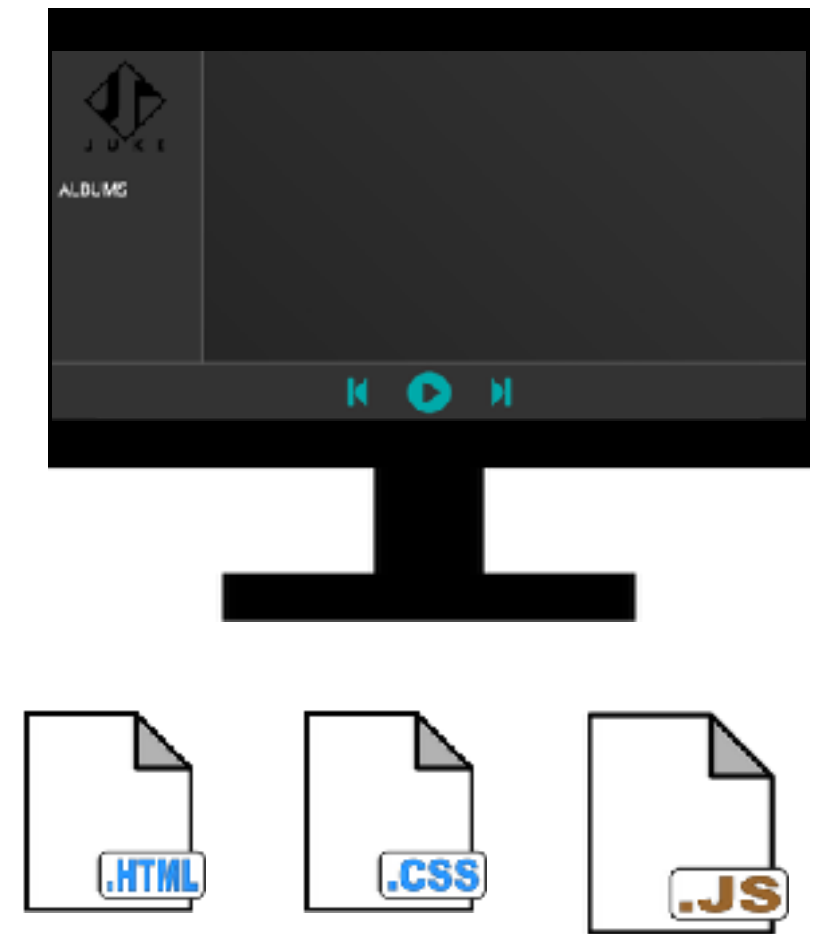
Client



Server

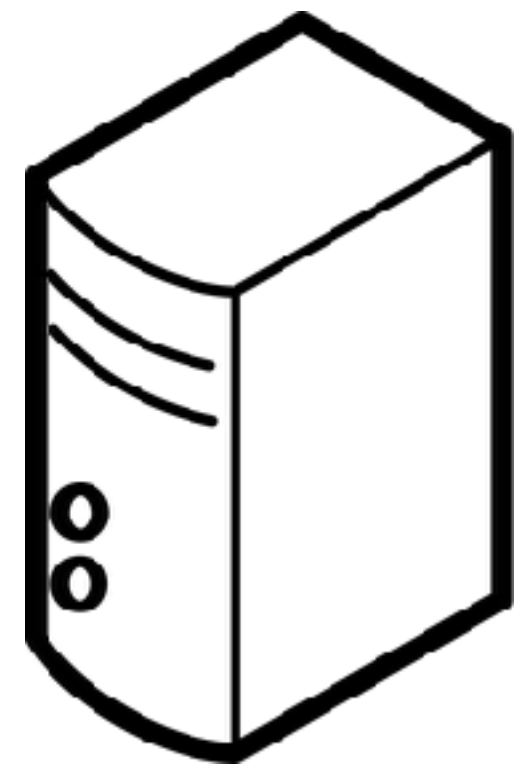


“Internet”



Client

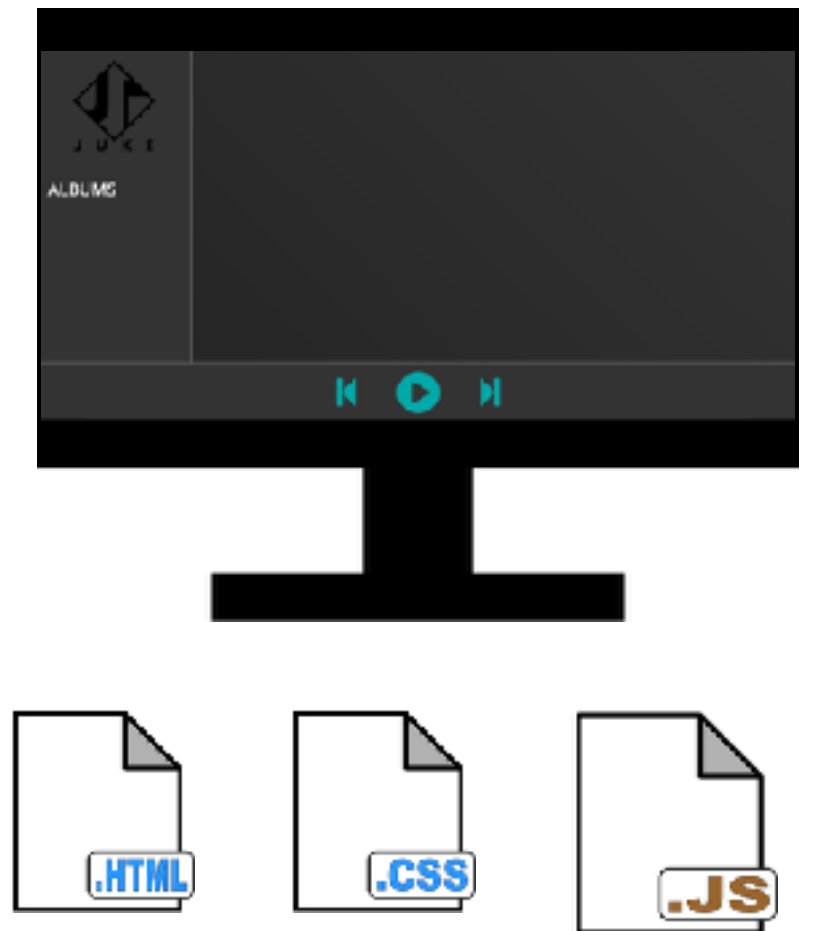
```
this.state = {  
  albums: []  
}
```



Server



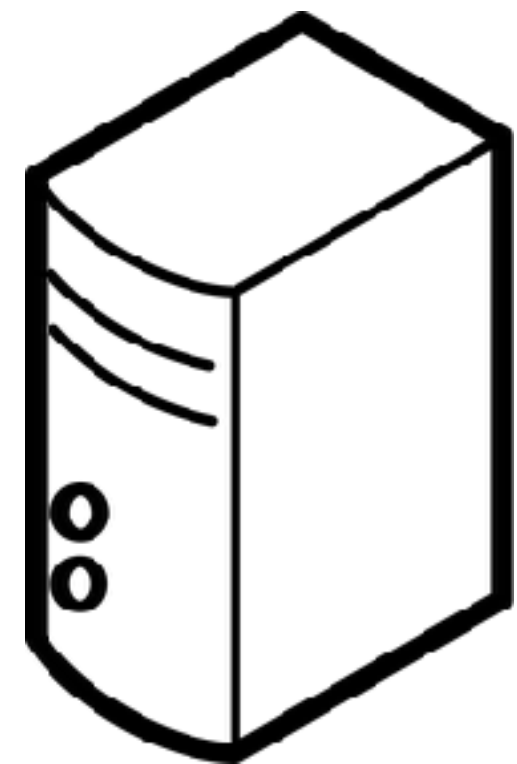
“Internet”



Client

```
this.state = {  
  albums: []  
}
```

HTTP GET /api/albums



Server

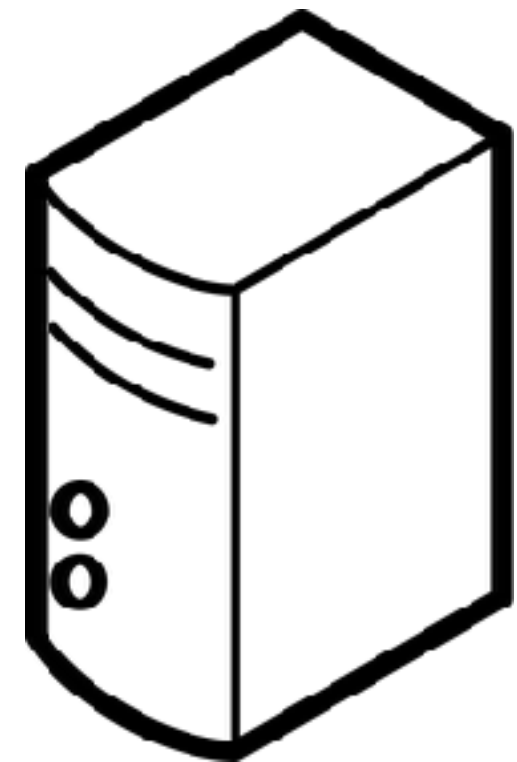
“Internet”

Client

```
this.state = {  
  albums: []  
}
```

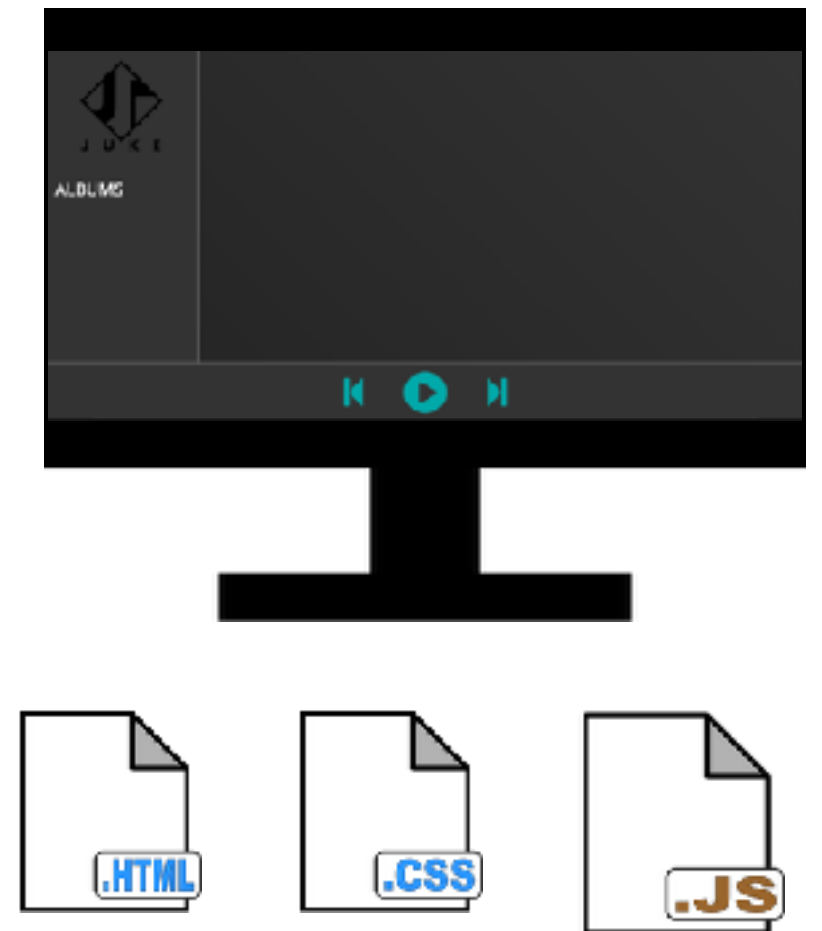
[{...}, {...}, {...}]

200 OK

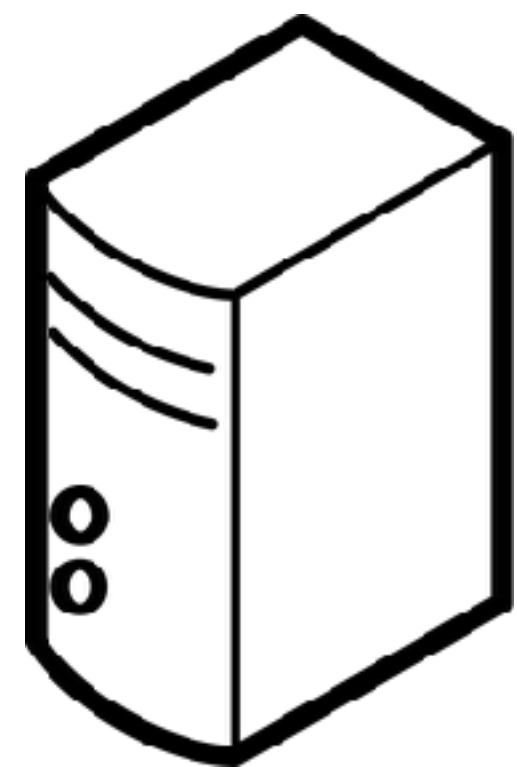


Server

“Internet”



Client



Server

“Internet”

```
this.state = {  
  albums: [  
    {...},  
    {...}  
  ]  
}
```



Client