

API development

Node.js, express.js, mysql2

NIFR 2023 - <https://github.com/nicklasdean/mysql2-example.git>

API development

Agenda

- What is a web API?
- Concept of a basic web API
- Mysql2
- Prepared Statements



Frontend

Focuses on layout, animations, content organization, navigation, graphics.

Programming languages:
JavaScript, HTML, CSS

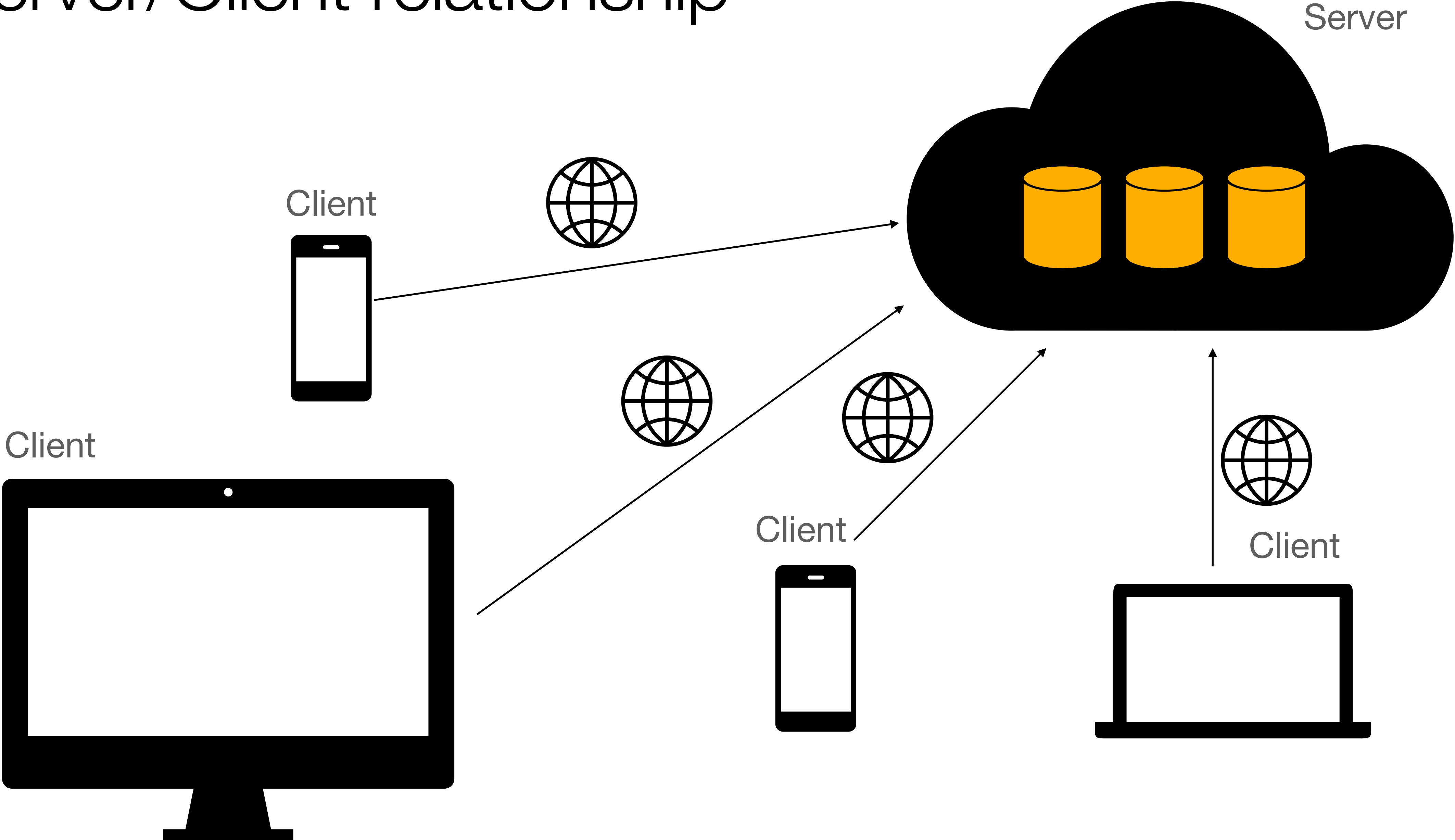


Backend

Focuses on building code, debugging, database management.

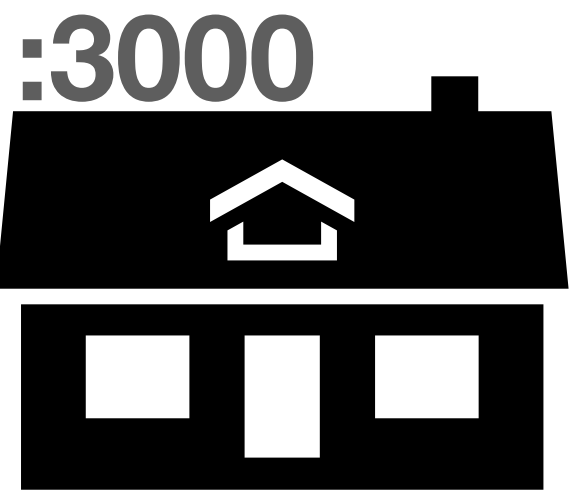
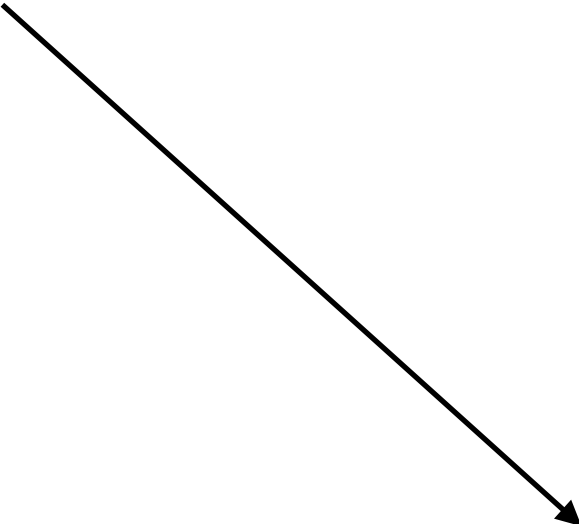
Programming languages:
Node.js, Python, Java

Server/Client relationship

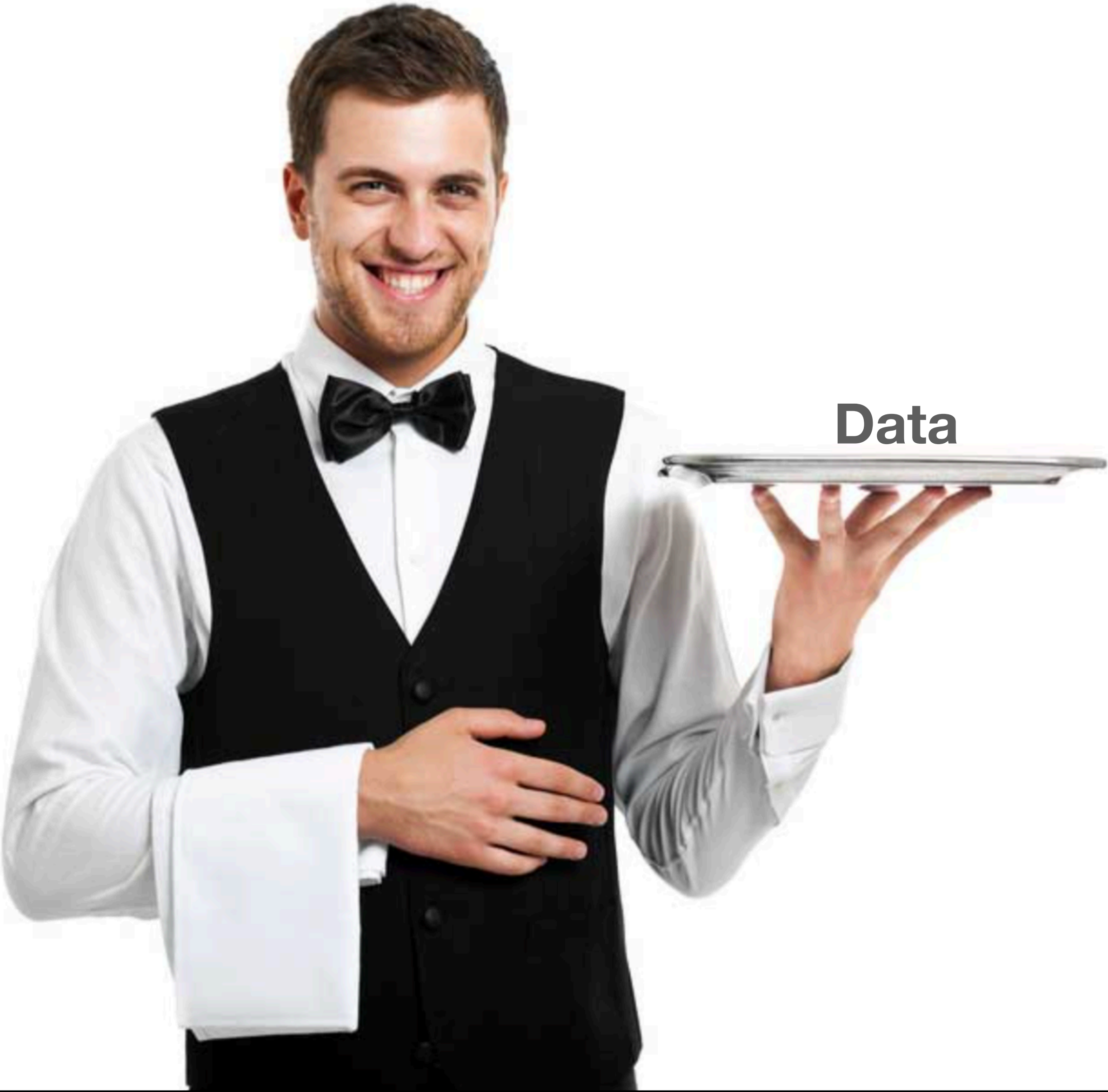


```
GET /docs/index.html HTTP/1.1
Host: www.nowhere123.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
(blank line)
```

Get *me* some *data*!

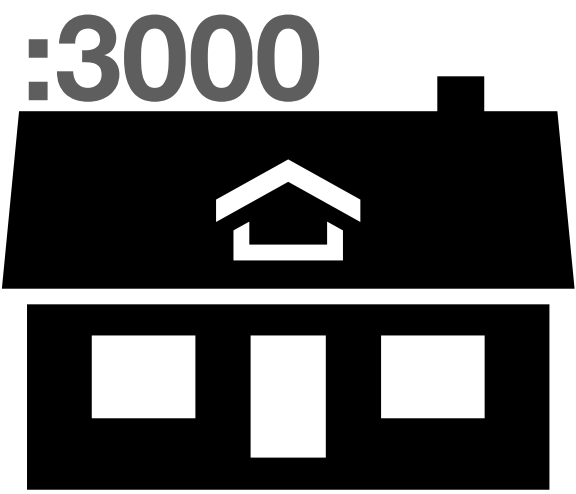


Server/backend

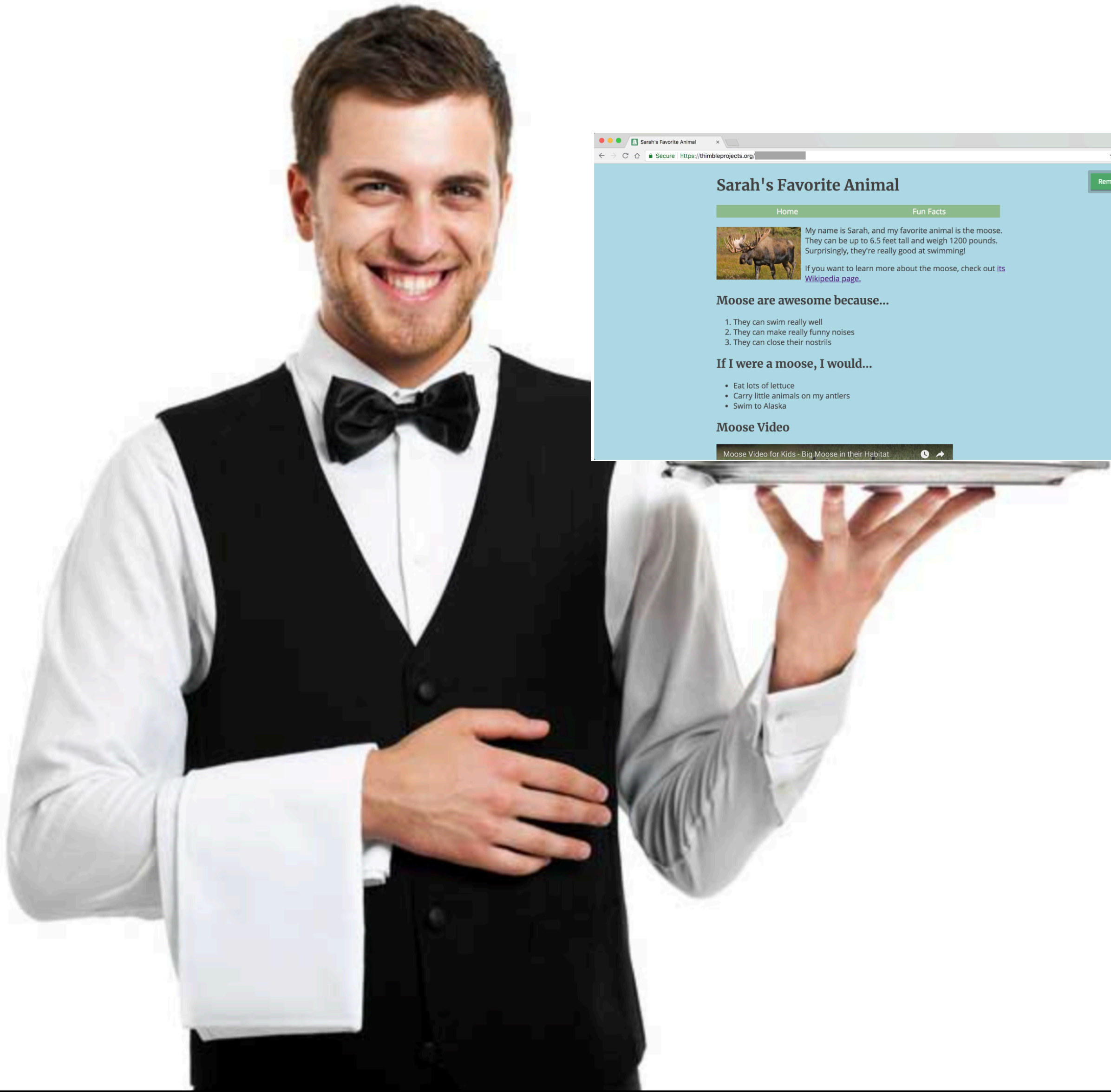



```
GET /docs/index.html HTTP/1.1
Host: www.nowhere123.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
(blank line)
```

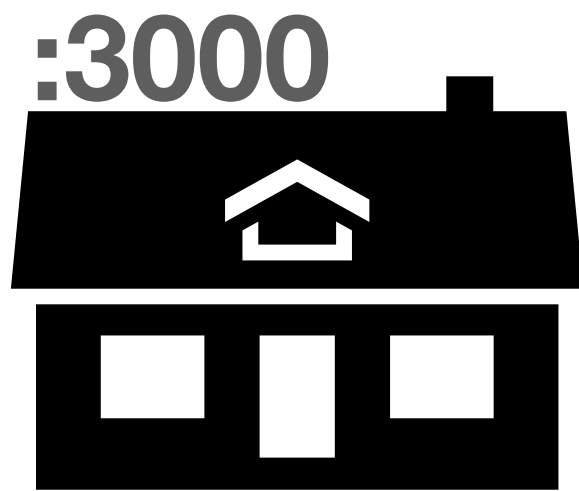
Get *me some data!*



Server side rendering

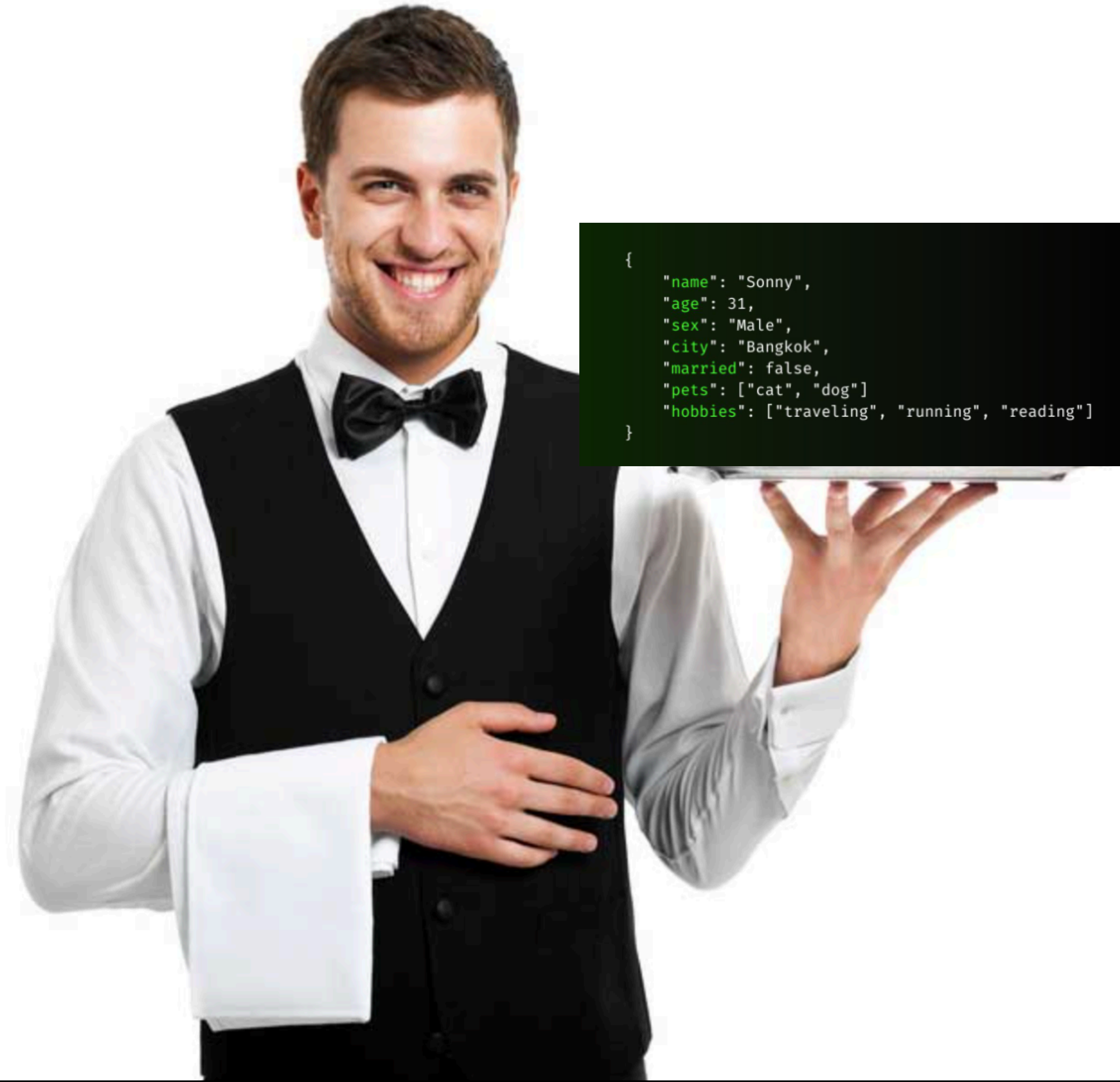


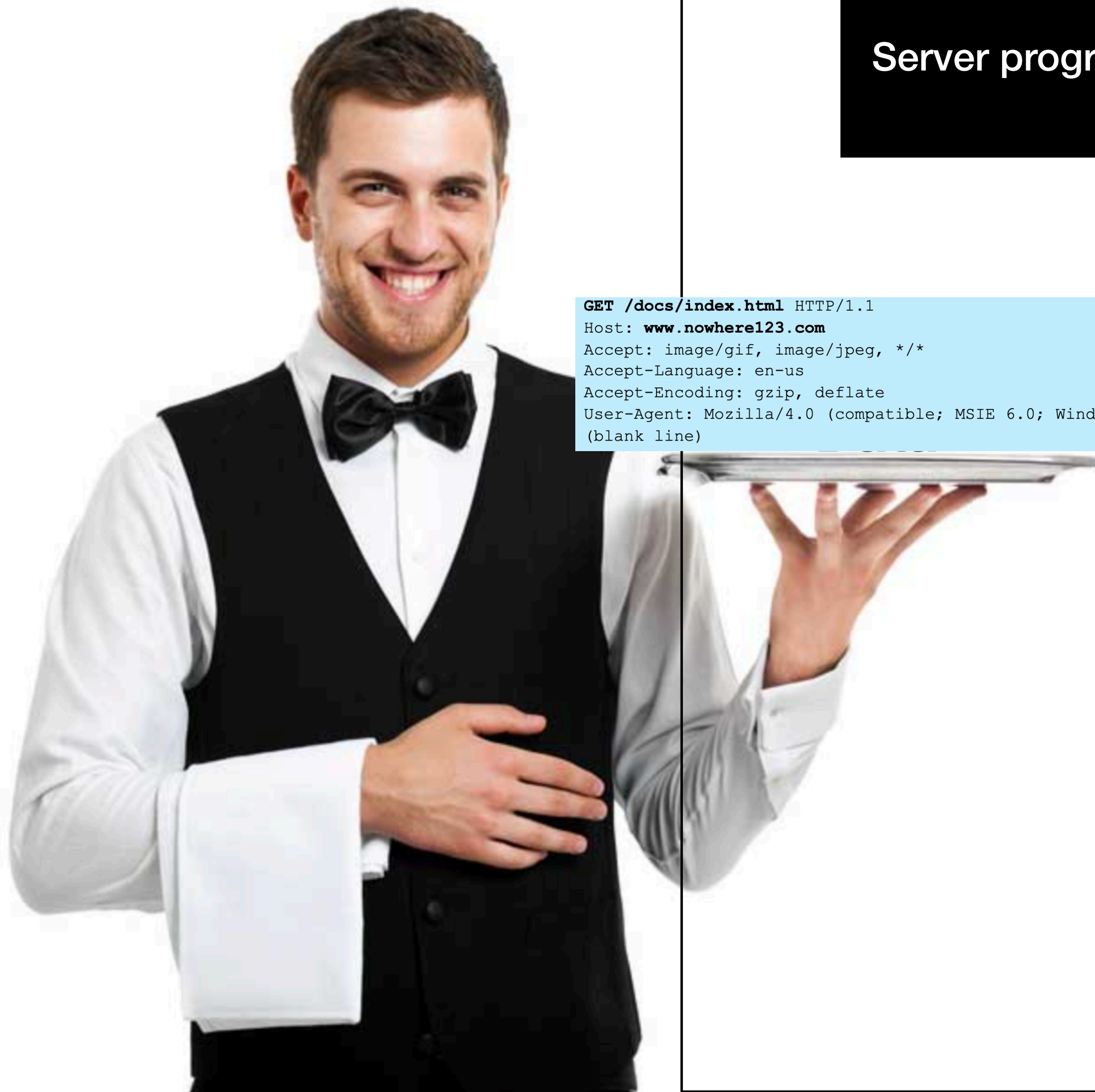
Fetch API: Get *me data!*



Client side rendering

Web api

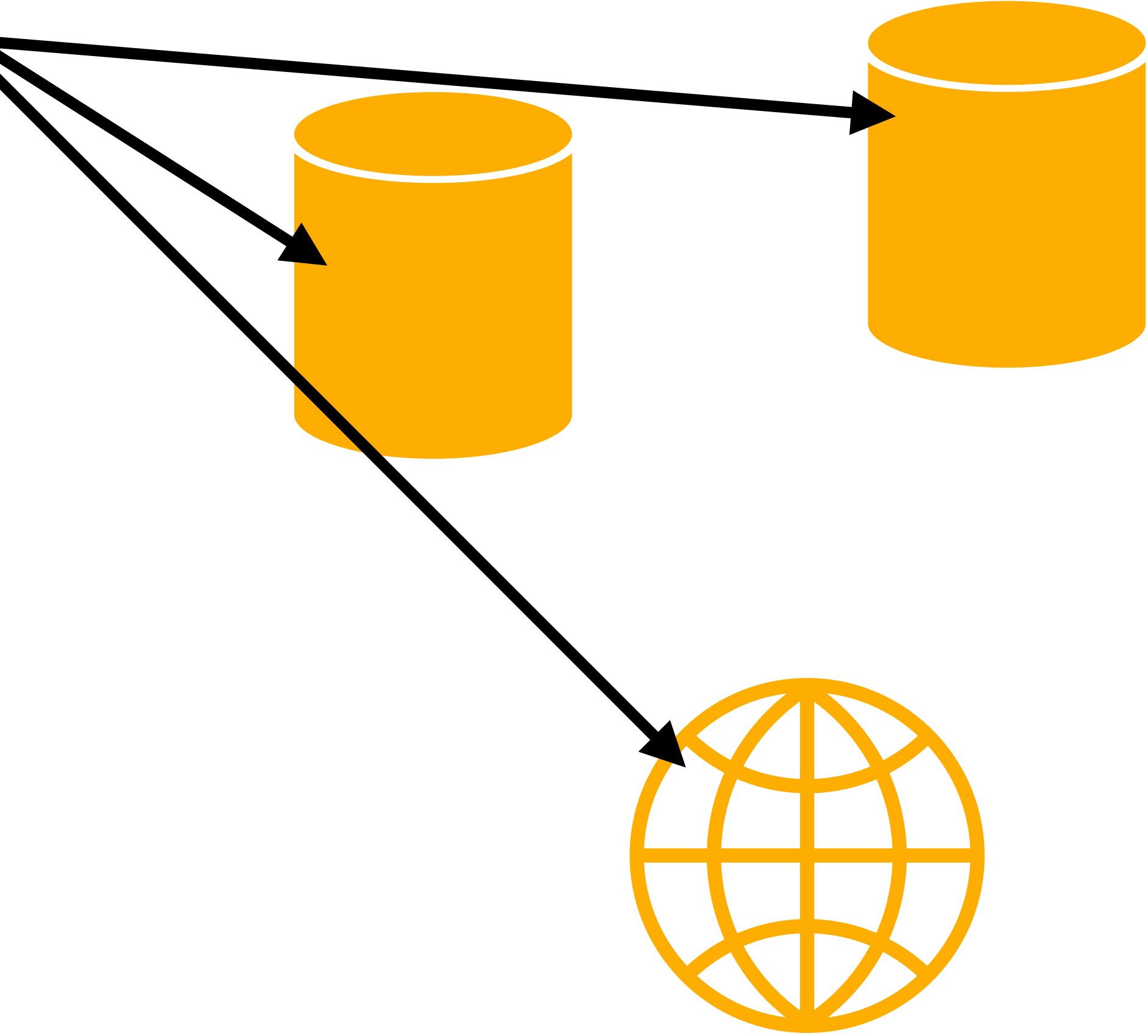


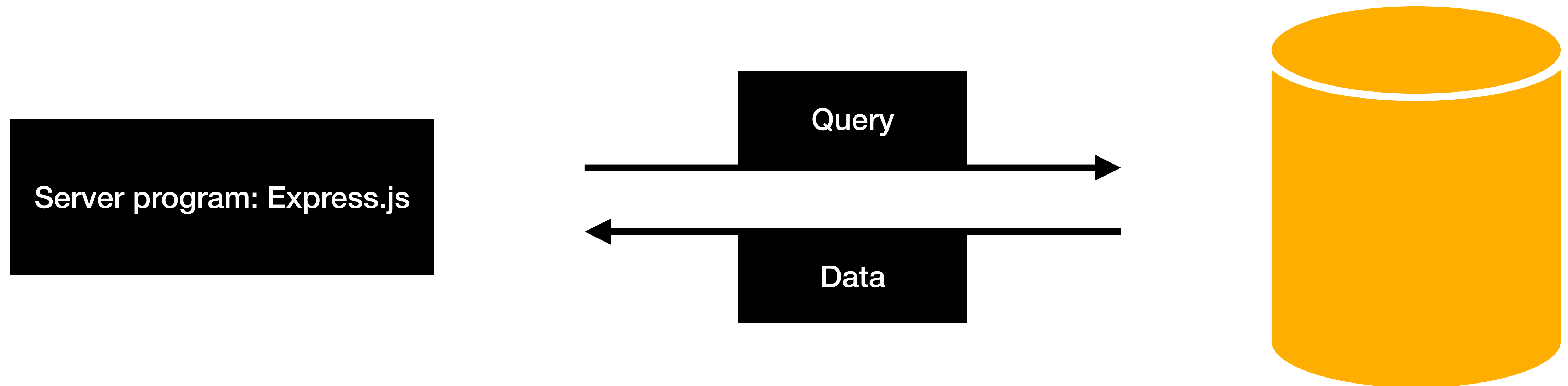


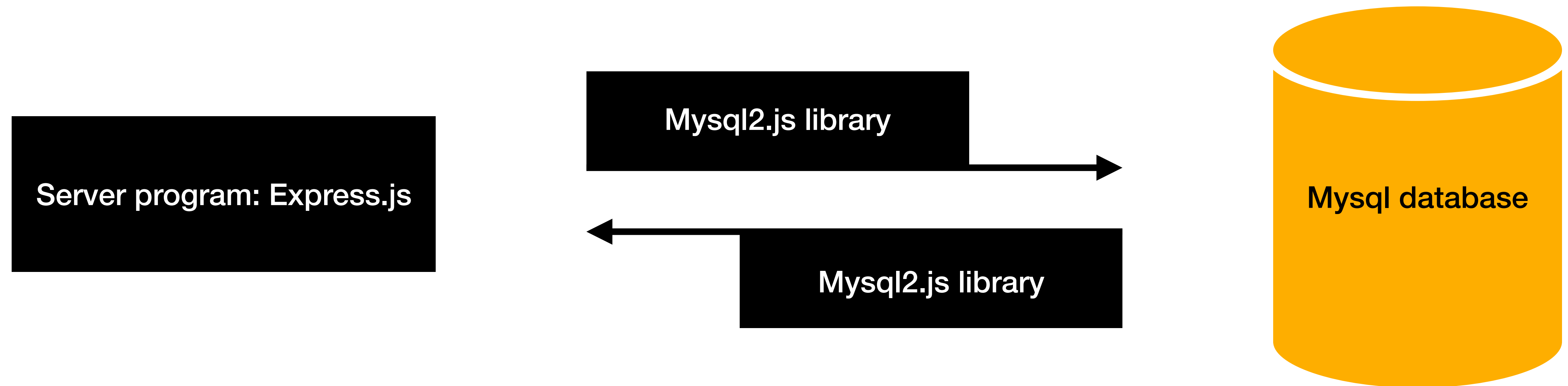
```
GET /docs/index.html HTTP/1.1
Host: www.nowhere123.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
(blank line)
```

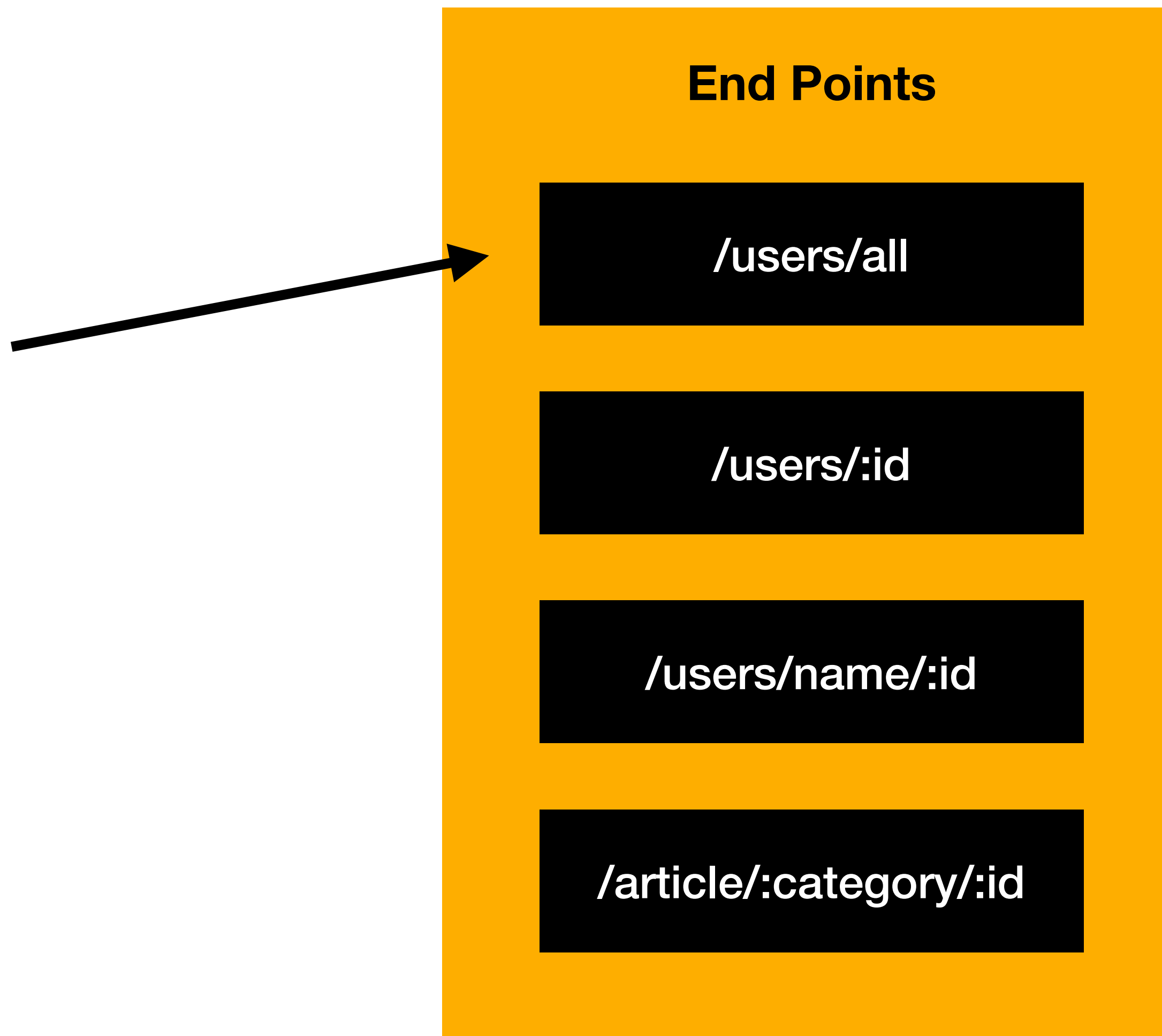
Server program: Express.js

Server/backend

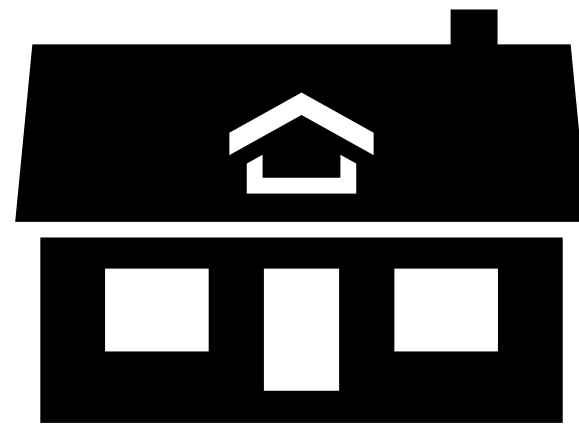








:3000



```
{  
  "name": "Sonny",  
  "age": 31,  
  "sex": "Male",  
  "city": "Bangkok",  
  "married": false,  
  "pets": ["cat", "dog"]  
  "hobbies": ["traveling", "running", "reading"]  
}
```

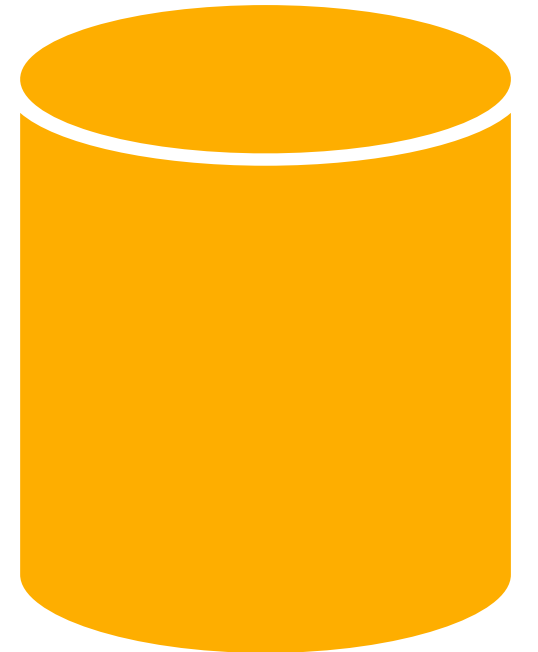
End Points

/users/all

/users/:id

/users/name/:id

/article/:category/:id



An end-point

For GET-requests

```
app.get('/', (req, res) => {  
    connection.query('SELECT `name` FROM pokemon', (error, results) => {  
        console.log(results);  
    });  
});
```

Insomnia

Connection.query has 2 parameters

Query itself + callback function

```
Connection.query(  
  [query],  
  [callback function(error, results)]  
)
```

Connection.query is overloaded

Can be called with 3 parameters

```
Connection.query(  
[query],  
[query parameters]  
[callback function(error, results)])
```

Connection.query is overloaded

Can be called with 2 parameters

```
app.get('/pokemon/:name/', (req, res)=>{
  const parameter = req.params.name;
  connection.query('SELECT * FROM pokemon WHERE `name` = "' + parameter + '"',
    (error, results) =>{
      res.send(results);
    });
});
```


Prepared statement

Multiple query parameters

```
app.get('/pokemon/:name/:primaryType/:speed', (req, res)=>{
  const name = req.params.name;
  const primaryType = req.params.primaryType;
  const speed = req.params.speed;

  connection.query('SELECT * FROM pokemon WHERE `name` = ? AND `primaryType` = ? AND `speed` > ?',
    [name, primaryType, speed],
    (error, results) =>{
      res.send(results);
    });
});
```

SQL injections

Basic web security

`www.pokedex.com/pokemon/;DROP
TABLE POKEMON/`

Prepared statements

Connection.query is overloaded

Can be called with 2 parameters

```
app.get('/pokemon/:name/', (req, res)=>{
  const parameter = req.params.name;
  connection.query('SELECT * FROM pokemon WHERE `name` = "' + parameter + '"',
    (error, results) =>{
      res.send(results);
    });
});
```

Connection.query is overloaded

Can be called with 3 parameters

```
app.get('/pokemon/:name/', (req, res)=>{
  const parameter = req.params.name;
  connection.query('SELECT * FROM pokemon WHERE `name` = ?',
    [parameter],
    (error, results) =>{
      res.send(results);
    });
});
```

Prepared statement

Connection.query using 3 parameters

```
app.get('/pokemon/:name/', (req, res)=>{
  const parameter = req.params.name;
  connection.query('SELECT * FROM pokemon WHERE `name` = ?',
    [parameter],
    (error, results) =>{
      res.send(results);
    });
});
```

Example