

WELCOME TO

REACT 11

LESSON 6

BASIC HTTP

ON TODAY'S LESSON

**BEFORE
ANYTHING WE
NEED TO REST**

REST



HTTP



MACHINE 1

MACHINE 2

**CREATE
READ
UPDATE
DELETE**

**CREATE
READ
UPDATE
DELETE**

**POST
GET
PUT/PATCH
DELETE**

**CREATE
READ
UPDATE
DELETE**

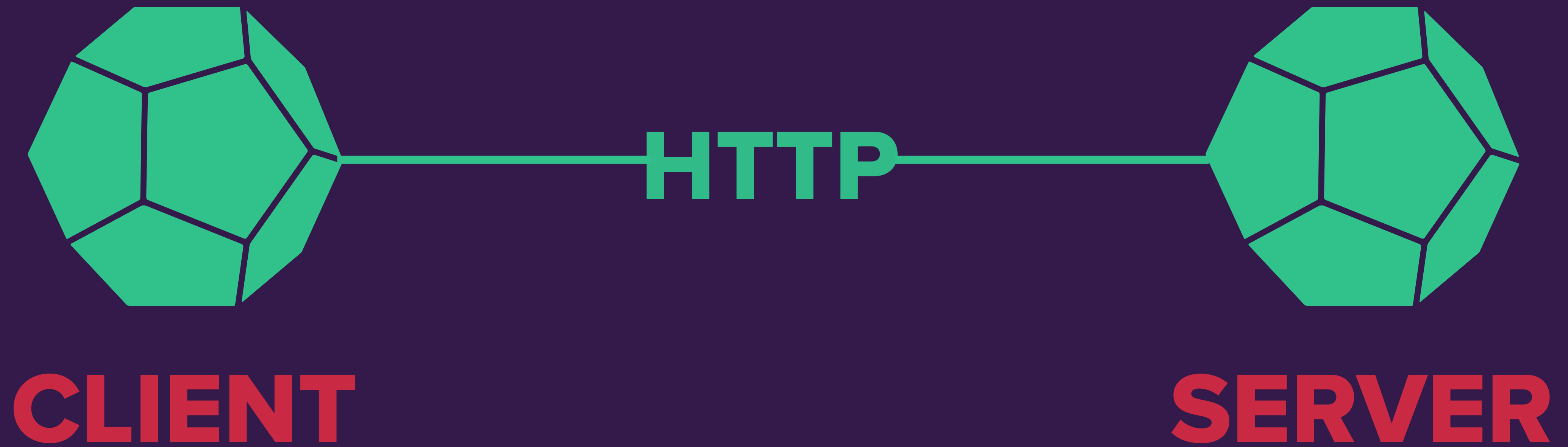
**POST
GET
PUT/PATCH
DELETE**

**WE RELY ON THESE
TO COMMUNICATE
WITH OUR
SERVER(S)**

**CREATE
READ
UPDATE
DELETE**

**POST
GET
PUT/PATCH
DELETE**

**WE RELY ON THESE
TO COMMUNICATE
WITH OUR
SERVER(S)**



**HOW DO WE DO THIS
IN REACT?**

HOW DO WE DO THIS IN REACT?

A LOT OF WAYS BUT THERE'S
THE SIMPLEST AND MOST
KNOWN WAY

**HOW DO WE DO THIS
IN REACT?**

**A LOT OF WAYS BUT THERE'S
THE SIMPLEST AND MOST
KNOWN WAY**

FETCH API

FETCH API

FETCH API **USES** **PROMISES**

FETCH API

USES

PROMISES

ARE

**OBJECTS THAT MAY
PRODUCE A SINGLE
VALUE IN THE FUTURE**

FETCH API

USES

PROMISES

FULFILLED

REJECTED

PENDING

ARE

OBJECTS THAT MAY
PRODUCE A SINGLE
VALUE IN THE FUTURE

PROMISE HANDLING

IN THE TRADITIONAL SENSE

GET EXAMPLE

componentDidMount() {

}

GET EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
  
  
}
```

GET EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
    .then ( (res)      => { return res.json() } )  
  
}
```

GET EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
    .then ( (res)      => { return res.json() } )  
    .then ( (resData) => { this.setState( { data: resData } ) } )  
  
}
```

GET EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
    .then ( (res)      => { return res.json() } )  
    .then ( (resData) => { this.setState( { data: resData } ) } )  
    .catch ( (error)   => { console.log(error) } )  
}
```

GET EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
    .then ( res      => res.json() )  
    .then ( resData => this.setState( { data: resData } ) )  
    .catch ( error   => console.log(error) )  
}
```

POST EXAMPLE

POST EXAMPLE

```
fetch('https://pokeapi.co/api/v2/pokemon/', {
```

POST EXAMPLE

```
fetch('https://pokeapi.co/api/v2/pokemon/', requestOptions)
```

POST EXAMPLE

```
const requestOptions = {  
  method: 'POST',  
  headers: { 'Content-Type': 'application/json' },  
  body:    JSON.stringify({ pokemon: { name: 'pikachu' } })  
}  
  
fetch('https://pokeapi.co/api/v2/pokemon/', requestOptions)
```

ASYNC AWAIT

AS INTRODUCED IN ES6

ASYNC AWAIT EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
    .then ( res      => res.json() )  
    .then ( resData => this.setState( { data: resData } ) )  
    .catch ( error   => console.log(error) )  
}
```

ASYNC AWAIT EXAMPLE

```
componentDidMount() {  
  // ...  
}
```

ASYNC AWAIT EXAMPLE

```
async componentDidMount() {  
  // ...  
}
```

ASYNC AWAIT EXAMPLE

```
async componentDidMount() {
```

```
  const res = await fetch('https://pokeapi.co/api/v2/pokemon/pikachu')
```

```
}
```


ASYNC AWAIT EXAMPLE

```
async componentDidMount() {  
  
  const res      = await fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
  const resData = await res.json()  
  
}
```

ASYNC AWAIT EXAMPLE

```
async componentDidMount( ) {  
  
  const res      = await fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
  const resData = await res.json()  
  this.setState( { data: resData } )  
  
}
```

ASYNC AWAIT EXAMPLE

```
componentDidMount() {  
  
  fetch('https://pokeapi.co/api/v2/pokemon/pikachu')  
    .then ( res      => res.json() )  
    .then ( resData => this.setState( { data: resData } ) )  
    .catch ( error   => console.log(error) )  
}
```

IMPORTANT NOTES

November 26, 2020 (Thursday)
December 3, 2020 (Thursday)

HOT and Cold Example Release
Capstone Deadline + Data Collection (Feedback etc.)

