Open Opened 3 hours ago by **Rubén Montero**

Peticiones asíncronas

Resumen

- Crearemos un componente `TimestampChecker.js`
- Usaremos la librería `axios` para lanzar una petición a http://time.akamai.com

Descripción

Visita la siguiente página:

- http://time.akamai.com

Sólo sale un número, ¿verdad?

Es un *timestamp*. Concretamente, el número de segundos que han pasado desde el 1 de enero de 1970[1].

Vamos a crear un componente que lanza una *petición asíncrona* a http://time.akamai.com, obtiene la hora y la pinta.

La tarea

Abre un terminal (cmd.exe) y **posiciónate** en la carpeta `react-dashboards/react-frontend/`. Desde ahí, **ejecuta**:

```
npm install axios
```

¡Enhorabuena! Has instalado la librería `axios`, que usaremos para que nuestros componentes manden peticiones HTTP *asíncronas*.

Ahora, **añade** un nuevo componente `src/components/timestamp_checker/TimestampChecker.js`:

```
import { useState, useEffect } from "react"

const TimestampChecker = (props) => {
    const [time, setTime] = useState(undefined);

    useEffect(() => {

    }, [])

    return <div data-cy='timestampChecker'>
        <p data-cy='title'>El timestamp según Akamai:</p>
        <p data-cy='timestamp'>{time}</p>
      </div>
}

export default TimestampChecker;
```

Ahora, ¡vamos a mandar la petición HTTP!

Primero, **importa** `axios`:

```
import axios from "axios";
```

...y luego, en el `useEffect`, **envía** una petición a http://time.akamai.com **así**:

```
    useEffect(() => {
        // Esto se ejecuta cuando sucede el efecto. ¡Enviamos la petición!
        axios.get("http://time.akamai.com")
    }, [])
```

Y después de enviar la petición... Nos *llegará* una respuesta. Para *hacer cosas* una vez llegue la respuesta, empleamos `.then`[2].

Completa el código en `TimestampChecker.js` para que, tras obtener la respuesta de http://time.akamai.com, *guarde* esa respuesta en el estado `time`. **Así**:

```
    useEffect(() => {
        // Esto se ejecuta cuando sucede el efecto. ¡Enviamos la petición!
        axios.get("http://time.akamai.com").then(response => {
            // Esto se ejecuta cuando la respuesta HTTP (será response.data) nos llega
            setTime(response.data);
        })
    }, [])
```

Para finalizar, en `Examples.js`, **añade** un nuevo `<div>` *a continuación* (debajo) del correspondiente al ejercicio más reciente. En él, **incluye** el componente que hemos creado y además un encabezado, **así**:

```
<div data-cy='issue14div'>
  <h1>Ejercicio 14</h1>
  <TimestampChecker />
</div>
```

¿Ves el *timestamp* desde http://localhost:3000/examples?

🏆 Por último

Sube tus cambios al repositorio en un nuevo *commit*.

---

1. También denominado *timestamp* Unix. Es una convención para representar una fecha y hora en el tiempo independientemente del tipo de calendario que se use. ↩️
2. La sintaxis `async-await` es más habitual que `.then` ↩️

Rubén Montero @ruben.montero changed milestone to %Sprint 3 3 hours ago