Open Opened 5 days ago by 📸 Rubén Montero

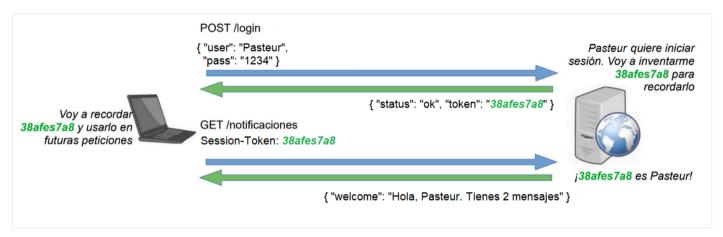
# ¡...publicando preguntas!



- Añadiremos estados enlazados a los campos del formulario de NewQuestion.js
- Cuando el usuario escriba una pregunta y haga submit, enviaremos la petición POST necesaria para publicar una pregunta nueva

# Descripción

¿Recuerdas el flujo de autenticación de sesiones?



En el tercer paso donde el cliente, para confirmar su identidad, manda el *token* al servidor (arriba Session-Token: 38afes7a8), dicho valor *podría* viajar en cualquier parte de la petición HTTP:

- 1. Línea de petición
- 2. Cabeceras
- 3. Cuerpo

...pero lo habitual es que viaje en las cabeceras HTTP.

## ¿Cómo mandar datos en las cabeceras HTTP con axios?

Enviando (pasando) un argumento 1 cuando invocamos axios.get, axios.post,... así:

```
{
  headers: {
    Cabecera1: "Valor1",
    Cabecera2: "Valor2"
  }
}
```

# Diferencia entre axios.get y axios.post

• axios.get <sup>2</sup>, admite 2 parámetros: (1) La "url" y (2) el *objeto* { headers: { Cabecera1: "Valor1" } }:

```
axios.get("url", { headers: { Cabecera1: "Valor1" } })

// Si sólo pasas 1 parámetro, se entiende que es la url
axios.get("url")
```

axios.post <sup>3</sup>, admite 3 parámetros: (1) La "url", (2) el cuerpo de la petición ( { } ) y (3) el objeto { headers: { Cabecera1: "Valor1" } }:

```
axios.post("url", requestBody, { headers: { Cabeceral: "Valor1" } })

// Si sólo pasas 2 parámetros, se entiende que son la url y el cuerpo de la petición
axios.post("url", requestBody)

// Si sólo pasas 1 parámetro, se entiende que es la url
axios.post("url")

// Para los dos primeros casos, recuerda que el requestBody
```

https://raspi/francisco.gomez/dwec/issues/146

```
// suele ser también un objeto JSON, e.g.:
axios.post("url", { nuevoComentario: "¡Genial!" })
```

En sí, esto se debe a que ciertos métodos HTTP (GET, DELETE) no soportan enviar un cuerpo de la petición.



#### La tarea

**Crea** estos estados en NewQuestion.js:

```
const [formTitle, setFormTitle] = useState("");
const [formDescription, setFormDescription] = useState("");
```

Añade estas funciones:

```
const onChangeTitle = (e) => {
    setFormTitle(e.target.value);
}

const onChangeDescription = (e) => {
    setFormDescription(e.target.value);
}

const onSubmitNewQuestion = (e) => {
    e.preventDefault();
    if ((formTitle.length === 0) || (formDescription.length === 0)) {
        return;
    }
    // To-Do: Enviar POST para subir nueva pregunta
    // (Lo compLetaremos Luego)
}
```

...y modifica el <form> para enlazar los estados a los <input> , así como el onSubmit :

## ¡Mandar el POST!

De acuerdo al fichero de especificación react-stackoverrun/django-backend/doc/swagger.yaml de tu repositorio, el POST debe ser así:

```
POST /dashboards/<id>
POST /dashboards/<id>
/questions
Session-Token: abc

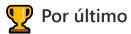
{
    "title": "Título ejemplo abc",
    "description": "Descripción ejemplo abc"
}
```

Completa la función onSubmitNewQuestion así:

```
const onSubmitNewQuestion = (e) => {
    e.preventDefault();
    if ((formTitle.length === 0) || (formDescription.length === 0)) {
        return;
    }
    // To-Do: Enviar POST para subir nueva pregunta
    // (Lo completaremos luego)
+ const sessionToken = sessionStorage.getItem("SESSION_TOKEN");
+ const requestBody = { title: formTitle, description: formDescription }
+ axios.post("http://raspi:8082/dashboards/" + params.dashboardId + "/questions", requestBody, { headers: { "Session-Token": session-Token": sessio
```

```
// Respuesta de éxito
  console.log(response);
})
```

¡Listo! Verifica en http://localhost:3000/dashboards/1/newQuestion que, en la consola del navegador, aparece algo demostrando que mandas con éxito un POST para publicar una nueva pregunta.



Sube tus cambios al repositorio en un nuevo commit.

1. Ese argumento es la *configuración* ( config ), y soporta, además de headers , <u>otros valores</u> 2. U otros como axios.delete  $\triangleright$ 3. U otros como axios.put

Rubén Montero @ruben.montero changed milestone to <a href="MSprint 5">MSprint 5</a> 5 days ago