Open Opened 3 weeks ago by Rubén Montero

## Subir (POST) una respuesta



- Crearemos un <form> en QuestionDetailPage.js que permitirá subir una respuesta
- Lo enlazaremos a un nuevo estado formDescription
- Cuando se haga submit del formulario, mandaremos un HTTP POST
- Añadiremos un nuevo estado para que tras subir una respuesta, aparezca automáticamente

### Descripción

Vamos a implementar, de forma análoga a las tareas anteriores, la funcionalidad para subir una respuesta.

Recuerda que la especificación para el *endpoint* POST que usaremos está en el fichero react-dashboards/django-backend/doc/swagger/swagger.yaml en tu repositorio.



#### La tarea

Como primer paso, añade un nuevo estado a QuestionDetailPage.js:

```
const [formDescription, setFormDescription] = useState('');
```

...y dos funciones, así:

```
const handleChangeDescription = (event) => {
    setFormDescription(event.target.value);
}

const handleSubmit = (e) => {
    e.preventDefault();
    if (formDescription.length == 0) {
        return;
    }

    // Vaciar el formulario
    setFormDescription('');
}
```

#### <form>

Crea src/screens/question/QuestionDetailPage.css:

```
.formContainer {
  margin: 25px auto; /* Con auto, el div se centrará */
  padding: 13px;
  width: 50%;
  background-color: rgb(185, 200, 200);
  text-align: center;
  box-shadow: rgba(0, 0, 0, 0.15) 1.95px 1.95px 2.6px 0px;
}
```

Y, luego, **importa** ese fichero desde QuestionDetailPage.js:

```
import "./QuestionDetailPage.css"
```

A continuación, **añade** un <div> con un <form> donde el usuario podrá escribir la respuesta. Será similar al de la tarea anterior. **Añádelo** debajo del :

```
<div data-cy='formContainer' className='formContainer'>
  <h4>¿Conoces la respuesta?</h4>
  <form data-cy='newAnswerForm' onSubmit={handleSubmit}>
  <input data-cy='newAnswerQuestionDescription' placeholder='Texto de la respuesta' value={formDescription} onChange={onChange}</pre>
```

```
<button data-cy='postDataButton'>Escribir respuesta/button>
  </form>
</div>
```

#### Petición POST

**Añade** las siguientes líneas a handleSubmit para que envíe la petición HTTP POST responsable de *publicar* la respuesta:

```
const handleSubmit = (e) => {
 e.preventDefault();
 if (formDescription.length == 0) {
 }
 const httpRequestBody = { description: formDescription }
  axios.post('http://raspi:8081/api/v1/dashboards/' + params.idDashboard + '/questions/' + params.idQuestion + '/answers', httpRed
  // Vaciar el formulario
  setFormDescription('');
}
```

#### Refrescar automáticamente

Para terminar, igual que en la tarea anterior:

- Añade un nuevo estado (por ejemplo, answersPublishedRightNow) con valor por defecto 0
- Asegúrate de que el useEffect en QuestionDetailPage.js reacciona ante ese estado
- Modifica ese estado (incrementándolo en 1) si la petición POST para publicar una respuesta tiene éxito (.then)

¡Enhorabuena!

Ya funciona la publicación de respuestas.



# Por último

Sube tus cambios al repositorio en un nuevo commit.

Rubén Montero @ruben.montero changed milestone to <a href="MSprint 4">MSprint 4</a> 3 weeks ago