Open Opened 5 days ago by 韼 Rubén Montero

Implementar detalle de un dashboard (otra vez)

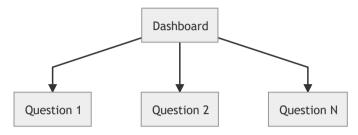


Resumen

• Implementaremos la página DashboardDetail.js

Descripción

Cada dashboard tiene una la lista de preguntas asociadas:



Vamos a trabajar en la página que muestra el detalle de un dashboard.

El endpoint REST que consumiremos (según qué dashboard queramos mostrar, e.g.: 1, 2,...) será:

- http://raspi:8082/api/v2/dashboards/1
- http://raspi:8082/api/v2/dashboards/2
- ..

¡Al lío!



La tarea

Una celda

Crea un fichero src/components/question_list_item/QuestionListItem.css:

```
.question-li {
    margin: 32px;
    padding: 5px;
    background-color: rgba(225, 230, 243, 0.5);
    text-align: center;
    border-radius: 15px;
    box-shadow: rgba(0, 0, 0, 0.15) 1.95px 1.95px 2.6px 0px;
}

.question-li:hover {
    cursor: pointer;
    background-color: rgb(220, 230, 230);
}
```

Y, luego, crea QuestionListItem.js en la misma carpeta:

```
import "./QuestionListItem.css"
import { useNavigate } from 'react-router-dom';

const QuestionListItem = (props) => {
   const navigate = useNavigate();

   const onClick = () => {
      navigate('questions/'+props.question.question_id)
   }

   return <div onClick={onClick} className='question-li' key={props.question.question_id}>
      <h3>{props.question.title}</h3>
      {props.question.description}
   </div>
}
```

https://raspi/francisco.gomez/dwec/issues/131

```
export default QuestionListItem;
```

Pedir los datos y mostrarlos

Primero, en DashboardDetail.js añade las dos líneas necesarias para poder usar los parámetros de la URL:

```
+import { useParams } from "react-router-dom";
const DashboardDetail = (props) => {
+ const params = useParams();
```

Y luego **añade** los siguientes estados:

```
const [dashboardTitle, setDashboardTitle] = useState('');
const [dashboardDescription, setDashboardDescription] = useState('');
const [questions, setQuestions] = useState([]);
```

Ahora, lo más importante.

Añade este useEffect para pedir al API REST los datos detallados del dashboard:

```
useEffect(() => {
  axios.get('http://raspi:8082/api/v2/dashboards/' + params.dashboardId).then(response => {
    setDashboardTitle(response.data.title);
    setDashboardDescription(response.data.description);
    setQuestions(response.data.questions);
 })
}, [])
```

Y, para terminar, **modifica** el return en DashboardDetail.js para añadir lo siguiente dentro del <div> principal (encima del <footer>):

```
<h1 data-cy="header">{dashboardTitle}</h1>
<h3 data-cy="description">{dashboardDescription}</h3>
<div data-cy="list_of_questions">
 { questions.map( q => { return <QuestionListItem question={q} /> }) }
</div>
```

Ya puedes verificar que si clicas en cada uno de los dashboards de http://localhost:3000 te lleva a un detalle donde ves correctamente los datos del dashboard y sus preguntas.



🥎 Por último

Sube tus cambios al repositorio en un nuevo commit.



Rubén Montero @ruben.montero changed milestone to %Sprint 5 5 days ago