


Open

Opened 5 days ago by  **Rubén Montero**

Implementar lista de dashboards (otra vez)


Resumen

- Implementaremos la página `Dashboards.js`

Descripción

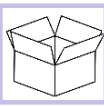
¿Recuerdas la metáfora del *museo de juguetes*?



localhost:3000




Orden de pedido


raspi:8081/api/v1/dashboards










Mercancia








Lista

localhost:3000



```
const Dashboards = () => {  
  
  const [dashboards, setDashboards] = useState([]);  
  useEffect(() => {  
    axios  
      .get('http://raspi:8081/api/v1/dashboards')  
      .then(response => {  
        setDashboards(response.data)  
      })  
  }, [])  
  
  return <div>  
    <div data-cy='dashboardsList'>  
      {  
        dashboards.map(  
          (d) => {  
            return <DashboardListItem dashboard={d}/>  
          }  
        )  
      }  
    </div>  
  </div>  
}
```

```
const DashboardListItem = (props) => {  
  return <div key={props.dashboard.id}>  
    <h3>{props.dashboard.title}</h3>  
    <p>{props.dashboard.description}</p>  
  </div>  
}
```

¡Volvamos a implementar una página que muestra *dashboards*!

En esta ocasión, nuestro [API REST](#) tiene el prefijo `/api/v2`. Puedes explorar su especificación en el fichero `react-stackoverrun/django-backend/doc/swagger/swagger.yaml` de tu repositorio.

La tarea

Instala `axios` (`npm install axios`) en `react-stackoverrun/react-frontend/`.

Una celda

Crea un fichero `src/components/dashboard_list_item/DashboardListItem.css`:

```
.dashboard-li {  
  margin: 21px;  
  padding: 5px;  
  background-color: rgba(239, 240, 240, 0.5);  
  text-align: center;  
  border-radius: 10px;  
  box-shadow: rgba(0, 0, 0, 0.15) 1.95px 1.95px 2.6px 0px;  
}  
  
.dashboard-li:hover {  
  cursor: pointer;  
  background-color: rgb(220, 230, 230);  
}
```

Y, luego, crea `DashboardListItem.js` en la misma carpeta:

```
import './DashboardListItem.css'  
import { useNavigate } from 'react-router-dom';  
  
const DashboardListItem = (props) => {  
  const navigate = useNavigate();
```

```
const onClick = () => {
  navigate('/dashboards/'+props.dashboard.id)
}

return <div onClick={onClick} className='dashboard-li' key={props.dashboard.id}>
  <h3>{props.dashboard.title}</h3>
  <p>{props.dashboard.description}</p>
</div>
}

export default DashboardListItem;
```

Pedir los datos y mostrarlos en celdas

Añade a `Dashboards.js` un estado `dashboards` :

```
const [dashboards, setDashboards] = useState([])
```

Más abajo, en `Dashboards.js` , **añade** un `useEffect` que se encarga de pedir los datos (GET) a <http://raspi:8082/api/v2/dashboards> y usa `.then` para gestionar la respuesta:

```
useEffect(() => {
  axios.get('http://raspi:8082/api/v1/dashboards').then(response => {
    setDashboards(response.data);
  })
}, [])
```

Y, finalmente, **incluye** en el `return` (dentro del `<div>` principal, por *encima* del `<footer>`) el siguiente `<div>` :

```
<div data-cy='listOfDashboards'>
  { dashboards.map((d) => { return <DashboardListItem dashboard={d}/> }) }
</div>
```

¡Ya puedes verificar que <http://localhost:3000> muestra la página inicial con una lista de *dashboards*!

🏆 Por último

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



Rubén Montero @ruben.montero changed milestone to [%Sprint 5](#) 5 days ago