

NeoGame

App en Angular 17 con Ionic, Capacitor y TailwindCSS

Esta es una aplicación desarrollada con Angular 17, Ionic, Capacitor y TailwindCSS. La aplicación utiliza Swiper para mostrar los juegos de manera interactiva y atractiva. Además, hace uso de una API externa para obtener la lista de juegos.

Tecnologías utilizadas

- Angular 17
- Node 23 >=
- Ionic
- Capacitor
- TailwindCSS
- Swiper
- API externa: Free-to-Play Games Database

Instalación

1. Clona el repositorio desde GitHub.
2. Navega al directorio del proyecto.
3. Instala las dependencias con el gestor de paquetes.
4. Si estás trabajando en un entorno de desarrollo web, ejecuta los comandos necesarios para construir el proyecto y sincronizarlo con Capacitor.
 - a. `npm i`
 - b. `npm start` (desplegamos versión local)
5. Para trabajar con un entorno móvil(Android)
 - a. `ionic build`
 - b. `npx cap sync android`
 - c. `npx cap open android`
 - d. Run app

API

La aplicación utiliza la API de Free-to-Play Games Database para obtener información sobre juegos gratuitos. Para evitar problemas de CORS al trabajar en un entorno local, la aplicación está configurada para pasar las solicitudes a través de un servidor proxy.

Sin embargo, al compilar la aplicación para Android o iOS, la API no proporciona un encabezado `Access-Control-Allow-Origin`, lo que provoca que la aplicación no pueda acceder a los datos.

Para solucionar este problema, se utiliza RapidAPI como intermediario. RapidAPI proporciona acceso a la API de juegos y maneja las solicitudes sin problemas de CORS.

Para configurar la API correctamente en tu entorno, se utiliza el siguiente endpoint proporcionado por RapidAPI:

<https://rapidapi.com/digiwalls/api/free-to-play-games-database>

Es importante tener en cuenta que, al usar RapidAPI, necesitarás una clave de API para autenticar tus solicitudes. Esta clave se puede incluir en el archivo de configuración del entorno.

Configuración de entorno

Asegúrate de agregar tu clave de API en el archivo de configuración del entorno, en la sección correspondiente a la API.

Uso de Swiper

La aplicación utiliza Swiper para la visualización interactiva de los juegos. Puedes personalizar los estilos y el comportamiento de Swiper directamente desde los archivos de configuración de TailwindCSS y Angular.

TailwindCSS

Para los estilos, se utiliza TailwindCSS para crear una interfaz de usuario responsiva y moderna. Si deseas modificar los estilos de la aplicación, puedes hacerlo directamente en los archivos de configuración de TailwindCSS o en los archivos HTML y CSS correspondientes.

Problema de CORS

Durante el desarrollo local, es posible que experimentes un error de CORS al hacer solicitudes a la API. Esto se soluciona utilizando RapidAPI como intermediario. Además, se pueden realizar configuraciones específicas en los archivos de configuración del proyecto para garantizar que las solicitudes a la API funcionen correctamente en los diferentes entornos.

Notas

- Si encuentras problemas de CORS en el entorno de producción móvil, verifica que estés utilizando la API de RapidAPI y que tu clave API esté configurada correctamente.
- Asegúrate de que la API de RapidAPI esté habilitada y que las solicitudes no excedan los límites del servicio.

