Secure API Key Management

In the previous lesson, I stored the API key for rawg.io in the source code and committed it to Git. I want to emphasize that this approach is a temporary solution and not the recommended way to manage API keys.

API keys should not be included in the source code because they can be seen by anyone who has access to the repository. This includes not only the developers working on the project but also potential attackers or malicious users. They can use our API key to perform malicious attacks on behalf of our client app.

So, what's a better way to handle API keys? We should store them as environment variables. Environment variables are like secret configuration values that can be set outside the code. By using environment variables, we can keep our API keys separate from the source code.

Also, we should avoid storing API keys in client apps such as our GameHub app. Even if we store them as environment variables, they'll still be included in the JavaScript bundle that we send to the client. So, they'll be visible with every network request and can be easily accessed by inspecting the network tab in Chrome DevTools.

For a more robust and secure solution, we should build a custom backend server that acts as a proxy between the client app and the external API. We can securely store the API key on the backend server, and then our client app communicates with the backend, which takes care of making the API requests on our behalf. This way, the API key stays hidden from the client-side code and remains safe from those who shouldn't lay their eyes on it.

Building a full backend to handle API calls and manage API keys is outside the scope of this course. However, I encourage you to explore backend development and API security in future studies to gain a comprehensive understanding of securing your applications.