

Platform agnosticism

Nest is a platform-agnostic framework. This means you can develop **reusable logical parts** that can be used across different types of applications. For example, most components can be re-used without change across different underlying HTTP server frameworks (e.g., Express and Fastify), and even across different types of applications (e.g., HTTP server frameworks, Microservices with different transport layers, and Web Sockets).

Build once, use everywhere

The **Overview** section of the documentation primarily shows coding techniques using HTTP server frameworks (e.g., apps providing a REST API or providing an MVC-style server-side rendered app). However, all those building blocks can be used on top of different transport layers ([microservices](#) or [websockets](#)).

Furthermore, Nest comes with a dedicated [GraphQL](#) module. You can use GraphQL as your API layer interchangeably with providing a REST API.

In addition, the [application context](#) feature helps to create any kind of Node.js application - including things like CRON jobs and CLI apps - on top of Nest.

Nest aspires to be a full-fledged platform for Node.js apps that brings a higher-level of modularity and reusability to your applications. Build once, use everywhere!