

NGINX Modules

WHEN SHOULD YOU BUILD ONE?

NGINX, at its core, is a collection of modules. Whether you are using core modules, like the http and stream modules, or 3rd party module, like geoip or RTMP, they are using the same module framework. With the addition of dynamic module support, modules are an even better way to add functionality to NGINX.

But just because you can write a module doesn't necessarily mean you should. There is a lot of customization that can be done in the basic NGINX configuration. Once you get beyond what can be done in the configuration, it is time to look into the next option, scripting. With the addition of the NGINX JavaScript Module (nginjscrip), it offered a higher performance scripting option to existing lua implementation. We are also working to add functionality so that you can write a full module in JavaScript, but this is still in a preview state. If the code for your solution running in nginjscrip is still causing too much of a performance impact, it is time to start looking into writing a fully integrated C module.



Scripting

If your desired behavior is not possible to handle with the configuration file alone, the next step would be to look into implementing that functionality with scripting. Currently in NGINX there are 3 support methods of adding functionality with scripting. The first and longest running is by using the perl module. This solution is a bit of a bridge between the and scripting, and usually used for less complex usecases. For more complex functionality, there is the option to either use Lua code, by leveraging the OpenResty Lua module, or by using JavaScript with the NGINX nginjscrip module. Scripting allows you to use existing, widely known languages to extend the functionality of NGINX. Both of these solutions do come with a performance impact, with nginjscrip being the more efficient of the two.



Config

The most straight forward way to modify how NGINX behaves is by creating customized configuration files. This may seem obvious, but by leveraging things like map blocks, you can drastically change how NGINX behaves and processes requests. This should be your first option, and if it can fit your use case, the best solution.



Modules

The most efficient, and most difficult, solution to add new features and functionality is by creating an NGINX module. This requires not only a good understanding of C programming, but also a good understanding of the NGINX internal architecture. This does allow you to implement an event based processing and non-blocking code that with the NGINX processes.

For more information, visit nginx.com or send us an email at nginx-inquiries@nginx.com