



<http://stackoverflow.com/questions/3057526/framework-vs-toolkit-vs-library>

the framework takes all the common works we do for every application, prepackage them all up to make it easy for user to start.



#Intro to Express

\* What is a framework? How is it different from a library?

framework is similar to library: both use someone else's code. But we have less control by using framework(framework already predefined lots of things for us.)

\* What is Express?

Express is a relatively small framework that sits on top of Node.js’s web server functionality

to simplify its APIs and add helpful new features. It makes it easier to organize

your application’s functionality with middleware and routing; it adds helpful

utilities to Node.js’s HTTP objects; it facilitates the rendering of dynamic HTML views;

it defines an easily implemented extensibility standard.. other web framework includes: python: flask, jango; ruby: rails;

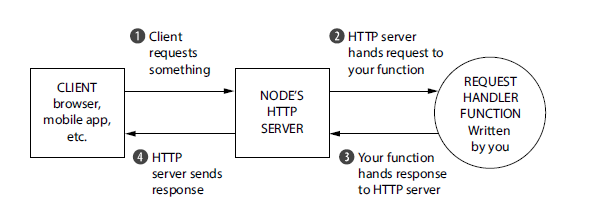
\* Why are we using Express?

- the mostly used framework in nodejs

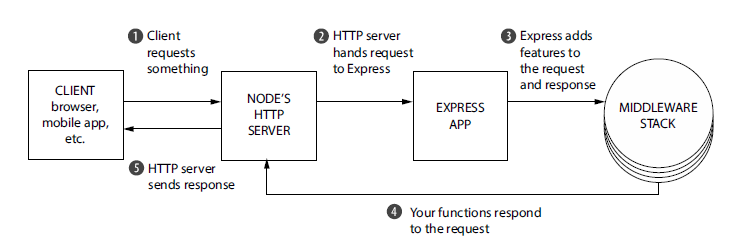
- heavy weight vs light weight framework: heavy framework: lots of predefined. you do little work; light weight framework, less predefined,you are in more control.

- express is a light weight, unopinionated(flexible, you can do anything you like) framework

The flow of a request through a Node.js web application:



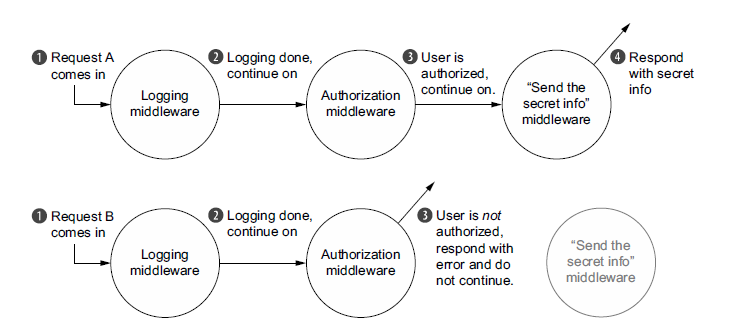
How a request would flow through an Express application:



The core parts of Express:

1. Middleware: small request handler functions each with different tasks: logging, authorization, etc

Two requests flowing through middleware functions:



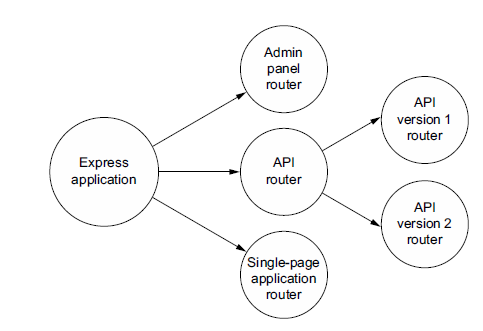
1. Routing for subapplications(named as routers)

Express applications can often be pretty small, even fitting in just one file. As your applications get larger, though, you’ll want to break them up into multiple folders and

files. Express is unopinionated about how you scale your app, but it provides one

important feature that’s super helpful: subapplications. In Express lingo, these miniapplications

are called *routers*.



1. Conveniences: just like jquery, write less do more!
2. Ecosystem: Express lives inside nodejs ecosystem and you could use lots of useful modules easily.