

# Rack Middleware

Sinatra rides on [Rack](#), a minimal standard interface for Ruby web frameworks. One of Rack's most interesting capabilities for application developers is support for "middleware" -- components that sit between the server and your application monitoring and/or manipulating the HTTP request/response to provide various types of common functionality.

Sinatra makes building Rack middleware pipelines a cinch via a top-level `use` method:

```
require 'sinatra'
require 'my_custom_middleware'

use Rack::Lint
use MyCustomMiddleware

get '/hello' do
  'Hello World'
end
```

The semantics of `use` are identical to those defined for the [Rack::Builder](#) DSL (most frequently used from rackup files). For example, the `use` method accepts multiple/variable args as well as blocks:

```
use Rack::Auth::Basic do |username, password|
  username == 'admin' && password == 'secret'
end
```

Rack is distributed with a variety of standard middleware for logging, debugging, URL routing, authentication, and session handling. Sinatra uses many of these components automatically based on configuration so you typically don't have to `use` them explicitly.

You can find useful middleware in [rack](#), [rack-contrib](#), or in the [Rack wiki](#).