

## Available Settings

<http://www.sinatrarb.com/intro.html#Available%20Settings>

### absolute\_redirects

If disabled, Sinatra will allow relative redirects, however, Sinatra will no longer conform with RFC 2616 (HTTP 1.1), which only allows absolute redirects.

Enable if your app is running behind a reverse proxy that has not been set up properly. Note that the `url` helper will still produce absolute URLs, unless you pass in `false` as the second parameter.

Disabled by default.

### add\_charset

Mime types the `content_type` helper will automatically add the charset info to. You should add to it rather than overriding this option: `settings.add_charset << "application/foobar"`

### app\_file

Path to the main application file, used to detect project root, views and public folder and inline templates.

### bind

IP address to bind to (default: `0.0.0.0` or `localhost` if your ``environment`` is set to `development`). Only used for built-in server.

### default\_encoding

Encoding to assume if unknown (defaults to `"utf-8"`).

### dump\_errors

Display errors in the log.

### environment

Current environment. Defaults to `ENV['RACK_ENV']`, or `"development"` if not available.

### logging

Use the logger.

### lock

Places a lock around every request, only running processing on request per Ruby process concurrently.

Enabled if your app is not thread-safe. Disabled per default.

### method\_override

Use `_method` magic to allow put/delete forms in browsers that don't support it.

### port

Port to listen on. Only used for built-in server.

### prefixed\_redirects

Whether or not to insert `request.script_name` into redirects if no absolute path is given. That way `redirect '/foo'` would behave like `redirect to('/foo')`. Disabled per default.

### protection

Whether or not to enable web attack protections. See protection section above.

### public\_dir

Alias for `public_folder`. See below.

### public\_folder

Path to the folder public files are served from. Only used if static file serving is enabled (see `static` setting below). Inferred from `app_file` setting if not set.

#### reload\_templates

Whether or not to reload templates between requests. Enabled in development mode.

#### root

Path to project root folder. Inferred from `app_file` setting if not set.

#### raise\_errors

Raise exceptions (will stop application). Enabled by default when `environment` is set to "test", disabled otherwise.

#### run

If enabled, Sinatra will handle starting the web server. Do not enable if using rackup or other means.

#### running

Is the built-in server running now? Do not change this setting!

#### server

Server or list of servers to use for built-in server. Order indicates priority, default depends on Ruby implementation.

#### sessions

Enable cookie-based sessions support using `Rack::Session::Cookie`. See 'Using Sessions' section for more information.

#### show\_exceptions

Show a stack trace in the browser when an exception happens. Enabled by default when `environment` is set to "development", disabled otherwise.

Can also be set to `:after_handler` to trigger app-specified error handling before showing a stack trace in the browser.

#### static

Whether Sinatra should handle serving static files.

Disable when using a server able to do this on its own.

Disabling will boost performance.

Enabled per default in classic style, disabled for modular apps.

#### static\_cache\_control

When Sinatra is serving static files, set this to add `Cache-Control` headers to the responses. Uses the `cache_control` helper. Disabled by default.

Use an explicit array when setting multiple values: `set :static_cache_control, [:public, :max_age => 300]`

#### threaded

If set to `true`, will tell Thin to use `EventMachine.defer` for processing the request.

#### traps

Whether Sinatra should handle system signals.

#### views

Path to the views folder. Inferred from `app_file` setting if not set.

#### x\_cascade

Whether or not to set the X-Cascade header if no route matches. Defaults to `true`.