Sinatra Cheat Sheet

A concise reference for Sinatra, a lightweight Ruby web framework. Covers essential concepts, routing, views, and helpers for quick development.

Core Concepts & Setup

Installation

Install Sinatra via RubyGems:

```
gem install sinatra
```

For logging:

```
gem install sinatra-contrib
```

Basic Structure

A simple Sinatra application structure:

```
require 'sinatra'

get '/' do
  'Hello, world!'
end
```

Running the app:

```
ruby your_app.rb
```

Configuration

`set :port, 8080`	Sets the port the application listens on.
`set :environment, :production`	Sets the environment (development, production, test).
`enable :sessions`	Enables session support.

Routing

HTTP Methods

`get 'path'`	Handles GET requests.
`post 'path'`	Handles POST requests.
`put 'path'`	Handles PUT requests.
`delete 'path'`	Handles DELETE requests.
`patch 'path'`	Handles PATCH requests.
`head 'path'`	Handles HEAD requests.
`options 'path'`	Handles OPTIONS requests.

Route Parameters

Accessing parameters from the route:

```
get '/hello/:name' do
  # Matches /hello/foo and /hello/bar
  # params[:name] will be 'foo' or 'bar'
  "Hello #{params[:name]}!"
end
```

Optional parameters:

```
get '/posts/:id?' do
  # Matches /posts/123 and /posts
  # params[:id] will be '123' or nil
end
```

Splats (capturing multiple segments):

```
get '/download/*/*' do
  # Matches /download/path/to/file.txt
  # params[:splat] will be ['path/to', 'file.txt']
end
```

Views & Templates

Rendering Views

Rendering a view:

```
get '/' do
  erb :index
end
```

This will render `views/index.erb`.

Specifying a different template path:

```
get '/' do
  erb :index, :views => settings.template_path
end
```

Template Engines

`erb`	Embedded Ruby templates.
`haml`	Haml templates (requires `haml` gem).
`slim`	Slim templates (requires `slim` gem).
`liquid`	Liquid templates (requires `liquid` gem).
`markdown`	Markdown templates (requires `rdiscount` or similar gem).

Layouts

Using a layout:

```
get '/' do
  erb :index, :layout => :default
end
```

This will render `views/index.erb` inside `views/default.erb`.

Disabling layout:

```
get '/' do
  erb :index, :layout => false
end
```

Helpers & Extensions

Defining Helpers

Defining a helper:

```
helpers do
  def bar(name)
    "Hello, #{name}!"
  end
end

get '/' do
  bar('World')
end
```

Built-in Helpers

`halt`	Immediately stops request processing.
`pass`	Passes control to the next matching route.
`redirect 'url'`	Redirects the client to the given URL.
`session`	Access the session hash (if sessions are enabled).
`params`	Access request parameters.

Sinatra Extensions

Using extensions:

```
require 'sinatra/reloader'

configure :development do
  register Sinatra::Reloader
end
```

Common extensions:

- `sinatra/reloader`: Automatic reloading on code changes.
- `sinatra/activerecord`: ActiveRecord integration.