

Using Sessions

A session is used to keep state during requests. If activated, you have one session hash per user session:

```
enable :sessions

get '/' do
  "value = " << session[:value].inspect
end

get '/:value' do
  session['value'] = params['value']
end
```

Note that `enable :sessions` actually stores all data in a cookie. This might not always be what you want (storing lots of data will increase your traffic, for instance). You can use any Rack session middleware: in order to do so, do **not** call `enable :sessions`, but instead pull in your middleware of choice as you would any other middleware:

```
use Rack::Session::Pool, :expire_after => 2592000

get '/' do
  "value = " << session[:value].inspect
end

get '/:value' do
  session['value'] = params['value']
end
```

To improve security, the session data in the cookie is signed with a session secret. A random secret is generated for you by Sinatra. However, since this secret will change with every start of your application, you might want to set the secret yourself, so all your application instances share it:

```
set :session_secret, 'super secret'
```

If you want to configure it further, you may also store a hash with options in the `sessions` setting:

```
set :sessions, :domain => 'foo.com'
```

To share your session across other apps on subdomains of `foo.com`, prefix the domain with a `.` like this instead:

```
set :sessions, :domain => '.foo.com'
```