

Forum:[Hack Forums](#)

Thread: [How to build sessions/cookies in Sinatra](#)

severos - February 24, 2015, 7:24 am

I'm using Ruby Sinatra to build a blog for my self, but I'd like to have multiple users posting on it, I've looked around trying to find how to build this, but not really finding what I want so thought of asking here.

When building sessions/cookies, what is exactly going in the background? I want all details that you know since I'm building things from scratch.

What I currently can do:

- process HTTP requests (GET/POST/DELETE/...)

- set session variables: something like `session[:foo] = Time.now` and can destroy them.

- set cookies: something like: `set_cookie "foo", "bar"`

How do I go around authentication? and keeping track of who's logged in and not??

I'm totally lost at this stage since I have no previous knowledge, any ideas? or resources??

Thanks for your time.

Crypto™ - February 24, 2015, 11:59 am

You'll need to enable sessions by doing a very simple bit of code:

Code enable :sessions

```
get '/foo' do
  session[:message] = 'Hello World!'
  redirect to('/bar')
end
```

```
get '/bar' do
  session[:message] # => 'Hello World!'
end
```

If you need some more control over sessions, use `Rack::Session::Cookie`

Code use `Rack::Session::Cookie`, `:key => 'rack.session'`,
`:domain => 'foo.com'`,
`:path => '/'`,
`:expire_after => 2592000`, # In seconds
`:secret => 'change_me'`

However, it might be easier to use an [authentication gem](#) for actual authentication management.

severos - February 24, 2015, 8:30 pm

I've actually said that I know how to enable sessions and set values for them, also how to create cookies, I was more like asking how does the authentication actually works.

Somewhy I found it hard to understand the authentication gem, and generally I'm not good with databases (which is why I'm trying to practice), that's why I want to create authentication system from ground.

If you have actual websites code examples written in Sinatra with authentication that'll be awesome.
Thanks for your help.

