

Webbserverprogrammering 1 – Sinatra Cheat Sheet v0.7 (rev 2022-04)

Köra ruby-program (i kommandotolken cmd.exe)

Ange kommandot **ruby** följt av namnet på filen som innehåller källkoden:

```
C:\>ruby app.rb
```

Ange adress i webbläsare:

```
http://localhost:4567/
```

Stäng ner server

```
CTRL + C
```

Navigera mellan mappar med kommandot **cd**:

```
C:\>cd MyFolder      byt mapp till MyFolder
C:\MyFolder\>cd ..   backa till föregående
```

Installationer

```
C:\>gem install sinatra
```

```
C:\>gem install slim
```

```
C:\>gem install sqlite3
```

```
C:\>gem install webrick
```

```
C:\>gem install sinatra-contrib
```

app.rb (controllern)

Funktionen **gets** (get string) låter användaren

```
require 'sinatra'
require 'slim'
require 'sqlite3'
require 'sinatra/reloader'
```

```
enable :sessions
```

```
get('/') do
  "Hello World!" #Visa sträng
end
get('/bananapage') do
  slim(:banan) #Visa innehåll i banan.slim
end
```

Formulär

Med formulär skickar du data ifrån webbläsaren till kontrollern **app.rb**. Oftast används metoden POST för dölja data som skickas.

```
form action="/bananroute" method="POST"
  input type="text" name="banannyckel"
  input type="submit" value="Skicka!"
```

Mapstruktur

I public-mappen ligger allt som du vill klienten ska kunna komma åt eller ladda ner

```
projekt
|- data (databas-filer)
|- public
|  |-img
|  |-css
|  |-js
|  index.html(ev statiska publika)filer
|- views
|  layout.slim
|  foo.slim
app.rb
```

Sessions

Session sparar data i en "session cookie" i webbläsaren.

```
enable :sessions #aktivera sessions (app.rb)
session[:banana_key] = "Apelsin" #Spara
h1 = session[:banana_key] #Läs (tex slimfil)
session.destroy #Töm session
```

Params, redirect

Params-hashan används för att komma åt data som har skickats ifrån ett formulär.

```
#app.rb
```

```
post('/bananroute') do
  my_data = params[:banannyckel]
  #hantera data
  redirect('/') #gå vidare till get-route
end
```

Params kan även användas för att plocka data ifrån en dynamisk route

```
get('/user/:banan_id') do
  my_data = params[:banan_id]
  #hantera data
  slim(:profile)
end
```

Locals

Använd locals för att skicka med data till SLIM-fil

```
#app.rb
```

```
get('/bananapage') do
  username = "Torsten"
  slim(:profile, locals:{key:username})
end
```

```
#profile.slim
```

```
h1 Welcome #{key}!
```

Instansvariabler

Alternativ till locals

```
#app.rb
```

```
get('/index') do
  @username = "Torsten"
  slim(:index)
end
```

```
#profile.slim
```

```
h1 Welcome #{@username}!
```

Slim (templatering)

Lägg alla dina slim-filer views-mappen.

Mallen måste kallas layout.slim:

```
<!DOCTYPE HTML>
html lang="en"
  head
    meta...
    link...
    title...

  body
    h1 Bananasite!
    == yield #HÄR hamnar dina andra slim-filer
```

Webbserverprogrammering 1 – Sinatra Cheat Sheet v0.7 (rev 2022-04)

SQLite3

Kommunicera med databasen med splite3

```
require 'sinatra'
require 'slim'
require 'sqlite3'
```

```
get('/') do
  #Skapa koppling till databasen
  db = SQLite3::Database.new("db/store.db")
  #Få svar i strukturen [{},{},{}]
  db.results_as_hash = true
  #Hämta data, skicka data
  result = db.execute("SELECT * FROM users")
  slim(:users, locals:{key:result})
end
```

SQL injection, white list, lägg till

Skydda din SQL-kod mot SQL injection

```
post('/user') do
  #Hämta data ifrån formulär
  user = params[:user]
  tel = params[:tel]

  #Lägg till data i databasen
  db = SQLite3::Database.new("db/store.db")
  db.execute("INSERT INTO users (a,b) VALUES
  (?,?)",user,tel)
  redirect('/')
end
```

SQL (ta bort, uppdatera)

```
post('/user/:id/update') do
  [...]
  db.execute("UPDATE users SET name=?,age=? WHERE
  id=?", params[:name],params[:age],params[:id])
  redirect('/')
end

post('/user/:id/delete') do
  [...]
  db.execute("DELETE FROM users WHERE
  id=?",params[:id])
  redirect('/')
end
```

BCRYPT, registrera ny användare

```
#app.rb
[...]
require 'bcrypt'

post('/user') do
  user = params["user"]
  pwd = params["pwd"]
  pwd_confirm = params["pwd_confirm"]

  db = SQLite3::Database.new("db/store.db")
  result=db.execute("SELECT id FROM users WHERE
  user=?",user)

  if result.empty?
    if pwd==pwd_confirm
      pwd_digest=BCrypt::Password.create(pwd)
      db.execute("INSERT INTO users(user,
      pwd_digest) VALUES(?,?)",user,pwd_digest)
      redirect('/welcome')
    else
      redirect('/error') #Lösenord matchar ej
    end
  else
    redirect('/login') #User existerar redan
  end
end
```

SQL Inner join

Hämta data ifrån flera tabeller

albums	id title
artist	id name
rel_tbl	id albums_id artist_id

```
get('/albums_for_artist_1') do
  [...]
  #Hämta artist 1's album
  @albums_of_artist1 = db.execute("SELECT * FROM
  rel_tbl INNER JOIN albums ON rel_tbl.album_id =
  album.id WHERE artist_id = 1")
  slim(:banan)
end
```

BCRYPT, login, Authentication

```
#app.rb
[...]
require 'bcrypt'

enable :session

post('/login') do
  user = params["user"]
  pwd = params["pwd"]

  db = SQLite3::Database.new("db/store.db")
  result=db.execute("SELECT id,pwd_digest FROM
  users WHERE user=?",user)

  if result.empty?
    redirect('/error') #Fel lösenord/användarnamn
  end

  user_id = result.first["id"]
  pwd_digest = result.first["pwd_digest"]

  if BCrypt::Password.new(pwd_digest) == pwd
    session[:user_id] = user_id
    redirect('/welcome')
  else
    redirect('/error') #Fel lösenord/username
  end
end
```

Webbserverprogrammering 1 – Sinatra Cheat Sheet v0.7 (rev 2022-04)

Before-block (lista vissa routes)

```
#app.rb

@protected_routes = ["/hej", "/hopp"]

before do
  if @protected_routes.include?(request.path_info)
    #Do nothing
  else
    redirect to (/error)
  end
end

get('/hej') do
  #Funkar!
end

get('/willneverbeeseen') do
  "Alla ser mig!"
end
```

Authorization (Äger användare resurs?)

```
post('/photos/:id/delete') do
  # 1.Äger inloggad resursen?
  owner=db.execute("SELECT user_id FROM photos
WHERE id=?",params[:id]).first["id"]
  # 2. Jämför med inloggads id
  if owner == session[:user_id]
    [fortsätt]
  else
    redirect to ('/error')
  end
end
```

Before-block (tex, är användare inloggad?)

```
before (/protected/*) do
  #Se tex om användare är inloggad
  if session[:user] == nil
    redirect to (/error)
  end
end

get('/protected/foo') do
  "Du ser mig bara om du är inloggad!"
end

get('/foo') do
  "Alla ser mig!"
end
```

```
before do
  if request.path_info != '/' # Alla sidor...
    redirect to('/error') # ...utom '/'
  end
end
```

