

Hello – Simply display a string from an object/HTTP request/Database and war file deployment on Glassfish.

Use Rails Controller to retrieve a String from either one of the following, and send to Rails View to generate an appropriate HTML:

- String object
- HTTP request parameter
- Database

WAR file (which stands for "web application archive") is a JAR file used to distribute a collection of JavaServer Pages, servlets, Java classes, XML files, tag libraries and static Web pages (HTML and related files) that together constitute a Web application.

Walkthrough these steps:

Summary:

1. **Open Netbeans IDE (double click the Netbeans desktop icon)**
2. **Create a new Ruby on Rails Application:** Call it "Hello" and select "Add rake targets to support app server deployment (.war)".
3. **Generate a controller** – Name: home Views: index
4. **Configure Controller and Viewer:**
 - a. Configure Controller - change "index" method such that it looks like:

```
def hello
  @hello_string = "Hello, I`m just a String"
end
```
 - b. Expand "Views", "home", open "index.html.erb" and add the following fragment at the bottom of the page:

```
<%= @hello_string %>
```
5. **GET/POST Parameters** - Add to the project a GET parameter:
 - a. Add to the controller "home_controller.rb" another line to use a GET parameter:

```
def index
  @hello_string = "Hello, I`m just a String"
  @from_get= params[:who]
end
```
 - b. Add to the viewer "index.html.erb" this lines at the bottom

```
<BR><BR>
What`s this?: <%= @from_get %>
```
6. **Add MySql Database to the project:**
 - a. Create Database – Right click on the project and select "Run/Debug Rake Task..."

b. In the Filter field, write "create" the machine task will be db:create
Click RUN

7. **Create model** -right click on the project and select "Generate", choose "model" as the generator and "greeting" in the "Arguments".

8. **Configure Model**

a. Configure Model - expand "Database Migrations -> migrate" and open "XXX_create_greetings.rb".
Change the "self.up" helper method such that it looks like:

```
def self.up
  create_table :greetings do |t|
    t.column :data, :string
  end
end
```

9. **Create Database** – Right click on the project and select "Run/Debug Rake Task... In the Filter field, write "create" the machine task will be db:create

10. **Migrate Database** - Right-select the project, select 'Migrate Database'-'>'To Current Version'.

11. **Add data to the table** – Execute SQL query – Right click on the "Tables" and choose "Execute a command..." Write the SQL query "**insert into greetings values (1, 'Hi, I'm from database')**"

Click on "Run SQL", mark in the red circle.

12. **Configure Controller** - Expand "Controllers", open "home_controller.rb" and change "hello" helper method such that it looks like:

```
def hello
  @hello_string = "Hello, I'm just a String"
  @from_get= params[:who]
  @hello_MySql = Greeting.find(1).data;
end
```

13. Expand "Views", "home", open "index.html.erb" and add the following fragment at the bottom of the page:

```
<BR><BR>
<%= @hello_MySql %>
```

14. **Run the application in glassfish** – Right click on the project and select "Run".

Direct your browser to the viewer to see the Strings:

<http://localhost:8080/Hello/home/index>

15. Add a GET parameter

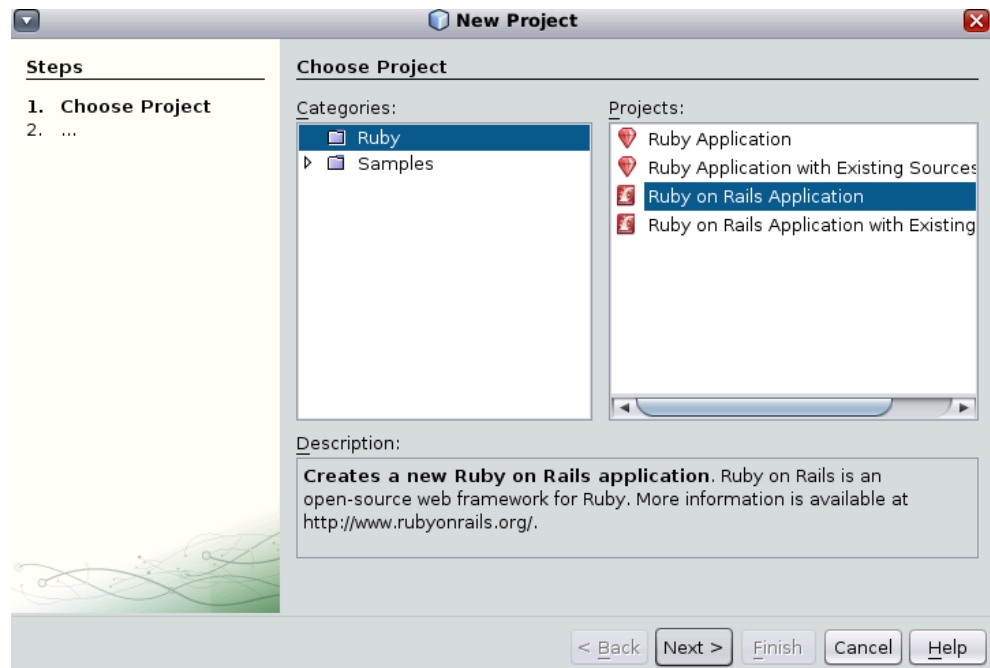
<http://localhost:8080/Hello/home/index?who=TheGET>

16. From the "Service" tab you can expand "Servers", expand "Glassfish V3" and expand "Applications" to see your application deployed in Glassfish.

(Note: To undeploy the application, just right click on it and select "Undeploy")

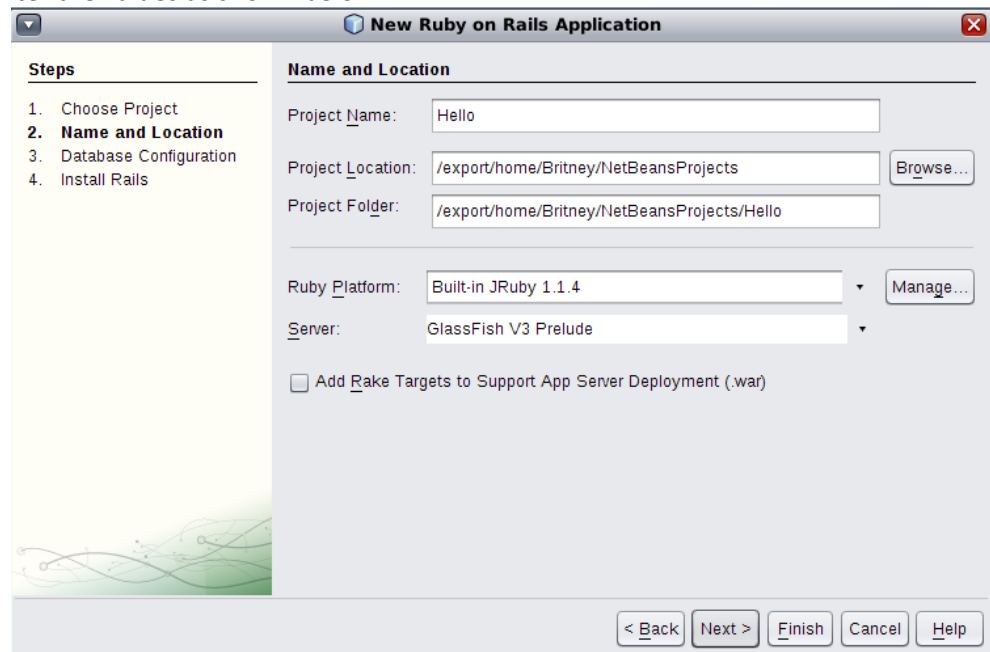
Full description:

1. *Open Netbeans IDE (double click the Netbeans desktop icon)*
2. *Create a new Ruby on Rails Application:*
 - a. Click on file -> "New project", select "Ruby on rails application"



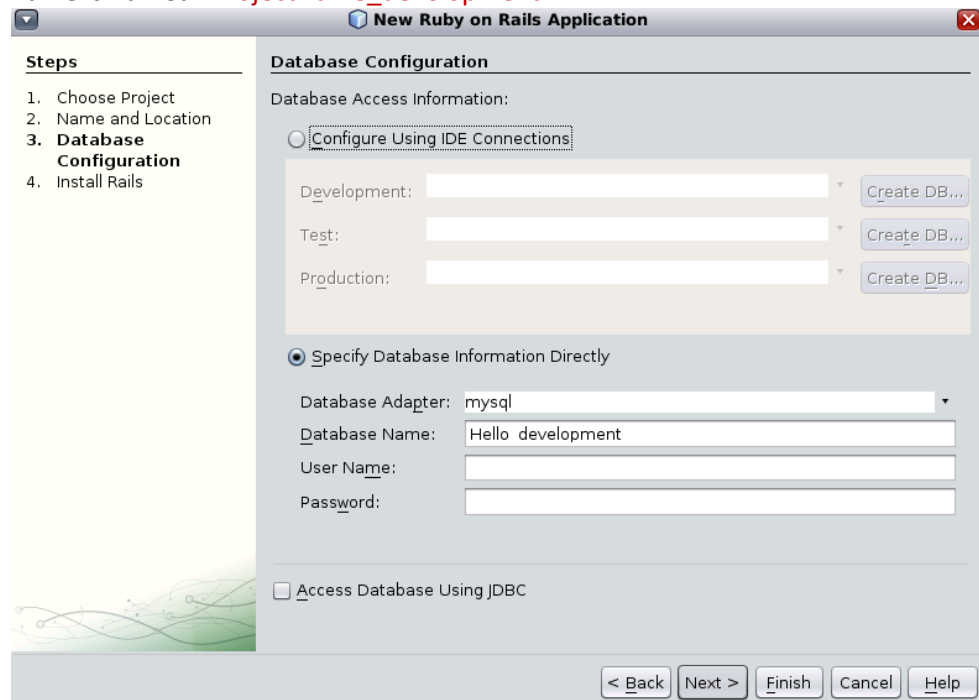
Click "Next"

- b. Enter the values as shown below:



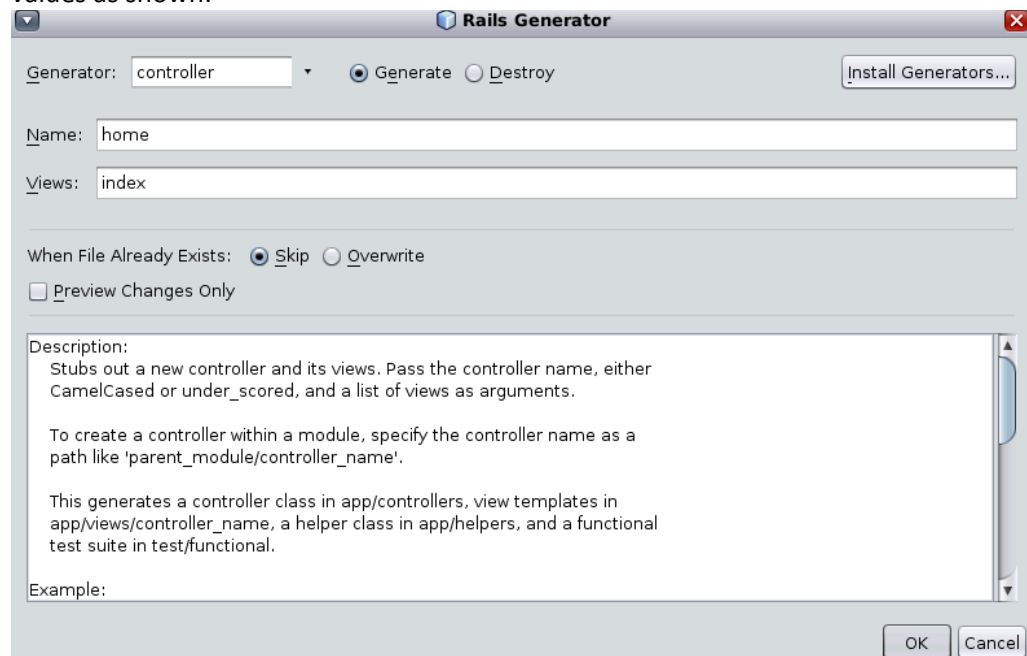
Notice, we've selected "Add rake targets to support app server deployment (.war)". This downloads the goldfish (nee rails integration) plugin and installs it in your application.

- c. Following Rails conventions, and since we are in development mode, the database name is named "ProjectName_development".



Click "Finish". NetBeans is generating a Rails project and install a warbler in it.

3. **Generate a controller** - Right-select the project, select "Generate..." and enter the values as shown:



Click "OK".

The controller generator created 4 files as shown in the output window:

```
create app/controllers/home_controller.rb
create test/functional/home_controller_test.rb
create app/helpers/home_helper.rb
create app/views/home/index.html.erb
```

4. **Configure Controller and Viewer:** (Note: The files are already open)

- a. Configure Controller - Expand "Controllers", open "home_controller.rb" and change "index" method such that it looks like:

```
def hello
  @hello_string = "Hello, I'm just a String"
end
```

- b. Expand "Views", "home", open "index.html.erb" and add the following fragment at the bottom of the page:

```
<%= @hello_string %>
```

5. **GET/POST Parameters** - You might want to access data sent in by the user or other request parameters in your controller actions. Rails does not make any distinction between GET and POST parameters, and both are available in the `params` hash structure in your controller.

Let's add to the project a GET parameter:

- a. Add to the controller "home_controller.rb" another line to use a GET parameter:

```
def index
  @hello_string = "Hello, I'm just a String"
  @from_get = params[:who]
end
```

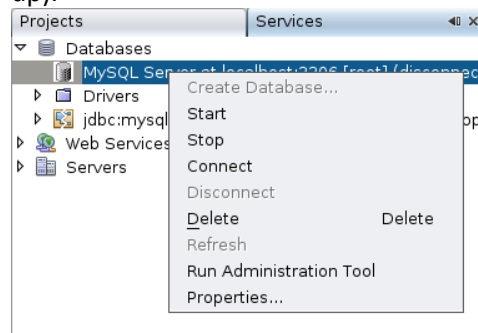
- b. Add to the viewer "index.html.erb" this lines at the bottom

```
<BR><BR>
```

```
What's this?: <%= @from_get %>
```

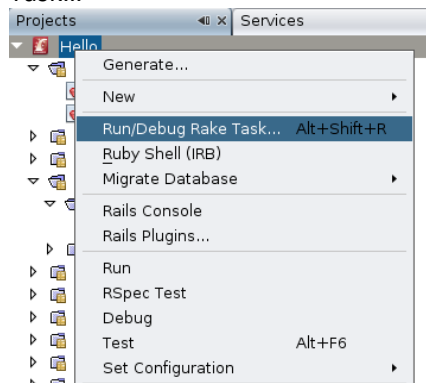
6. **Now, Let's add MySql Database to the project:**

- a. Back to NetBeans – go to "Services" tab expand Databases, right click on MySQL server" and select Connect (MySQL has been automatically started in start up).

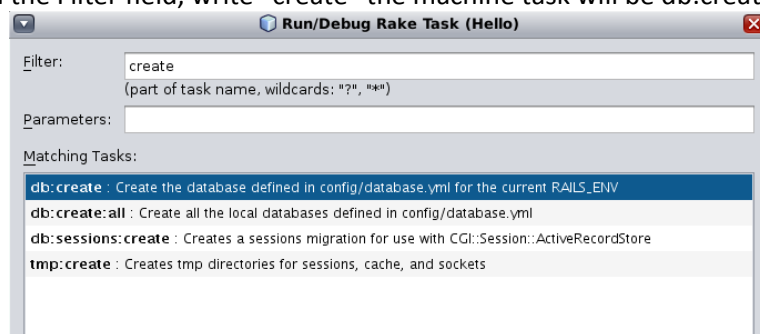


MySQL server is running and NetBeans is now connected, all operations on the server can be done from the NetBeans IDE.

b. Create Database – Right click on the project and select "Run/Debug Rake Task..."

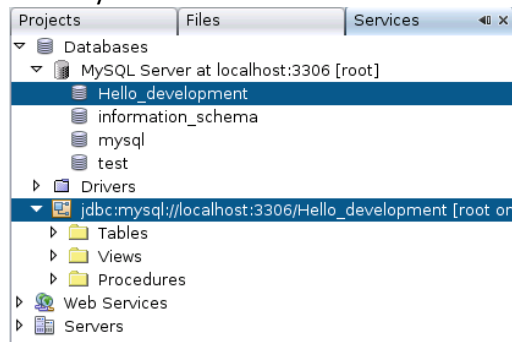


c. In the Filter field, write "create" the machine task will be db:create



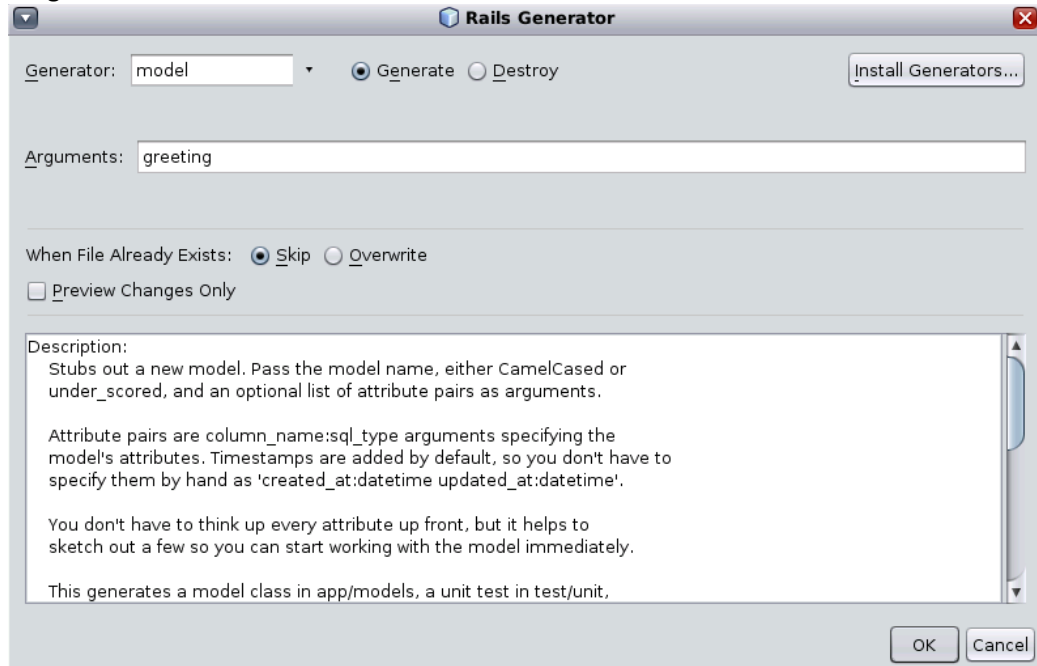
Click RUN

Notice you've created a new Database and a new connector.



7. **Create model** (Rails representation of a database table) – right click on the project and select "Generate", choose "model" as the generator and "greeting" in the

"Arguments".



The screenshot shows the 'Rails Generator' dialog box. At the top, the title bar says 'Rails Generator'. Below it, there's a 'Generator:' dropdown menu set to 'model'. To its right are two radio buttons: 'Generate' (selected) and 'Destroy'. Further right is a button labeled 'Install Generators...'. Below this is an 'Arguments:' text input field containing the word 'greeting'. Underneath that is a section 'When File Already Exists:' with two radio buttons: 'Skip' (selected) and 'Overwrite'. Below that is a checkbox labeled 'Preview Changes Only' which is unchecked. A large text area contains a 'Description:' of the generator's function, explaining that it stubs out a new model with attributes like timestamps and a migration file. At the bottom right are 'OK' and 'Cancel' buttons.

Click "OK"

4 new file been generated as you can see in the output window:

```
create app/models/greeting.rb
create test/unit/greeting_test.rb
create test/fixtures/greetings.yml
create db/migrate/20081106131652_create_greetings.rb
```

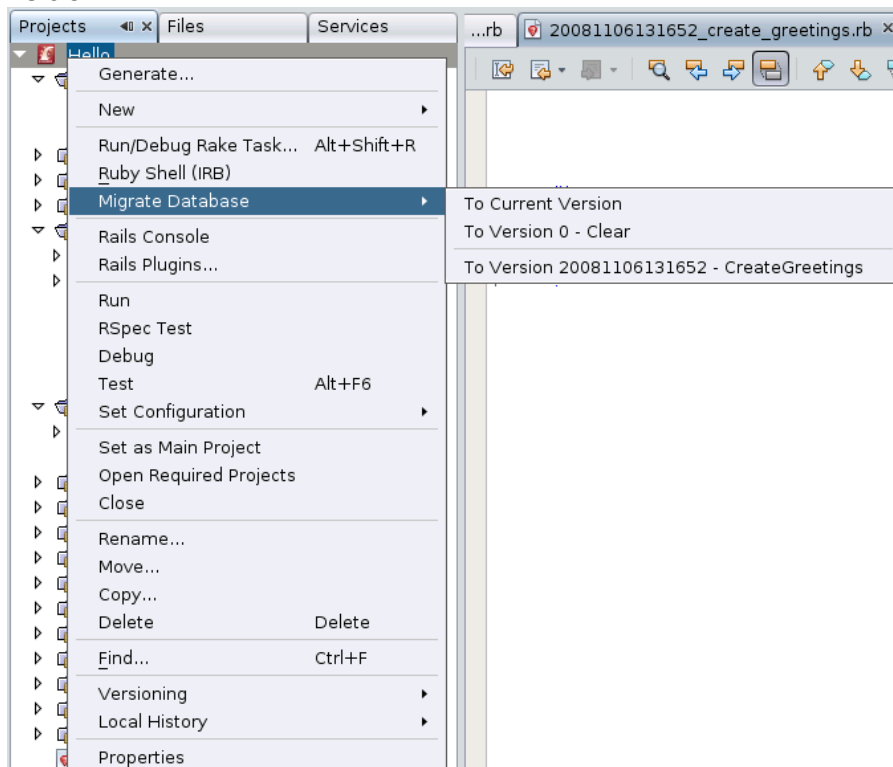
8. Configure Model

a. Configure Model - In the NetBeans IDE, expand "Database Migrations -> migrate" and open "XXX_create_greetings.rb".

Change the "self.up" helper method such that it looks like:

```
def self.up
  create_table :greetings do |t|
    t.column :data, :string
  end
end
```

9. **Migrate Database** - Right-select the project, select 'Migrate Database'-'>'To Current Version'.

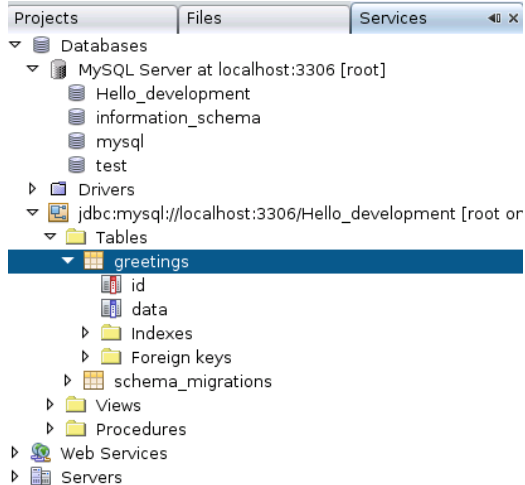


This generates the appropriate database tables and the following is shown in the output window:

```
(in /export/home/Britney/NetBeansProjects/Hello)
== 20081106131652 CreateGreetings: migrating
=====
-- create_table(:greetings)
--> 0.0104s
== 20081106131652 CreateGreetings: migrated (0.0146s)
=====
```


10. You can see the tables from the Services tab.

(Note: if you can't see the drivers, right click on the database and select connect)



11. **Add data to the table** – Execute SQL query – Right click on the "Tables" and choose "Execute a command..." Write the SQL query "**insert into greetings values (1, "Hi, I'm from database")**"



Click on "Run SQL", mark in the red circle.

The output window shows:

Executed successfully in 0.007 s, 1 rows affected.

Line 1, column 1

Execution finished after 0.007 s, 0 error(s) occurred.

(Note: to view the data, right click on the table and select "View data...")

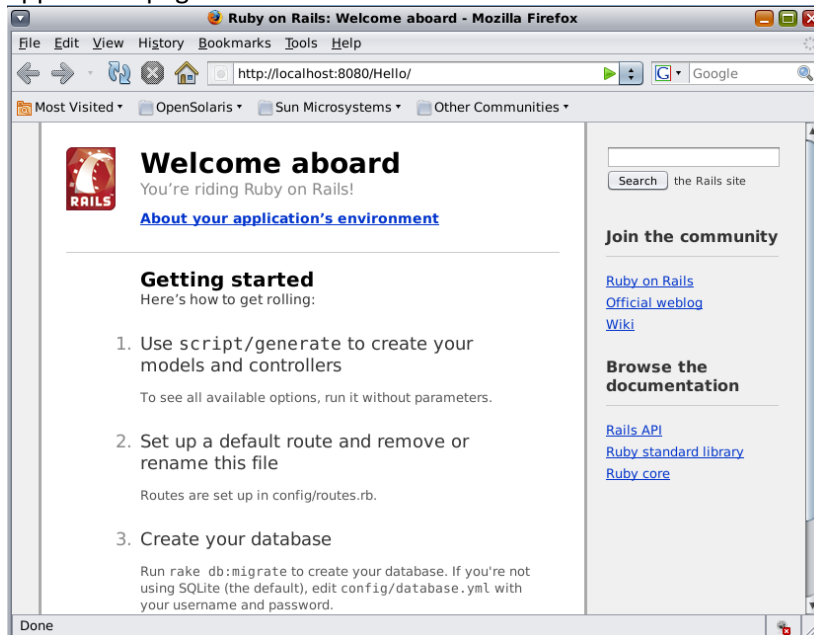
12. Configure Controller - Expand "Controllers", open "home_controller.rb" and change "hello" helper method such that it looks like:

```
def hello
  @hello_string = "Hello, I'm just a String"
  @from_get= params[:who]
  @hello_MySql = Greeting.find(1).data;
end
```

13. Expand "Views", "home", open "index.html.erb" and add the following fragment at the bottom of the page:

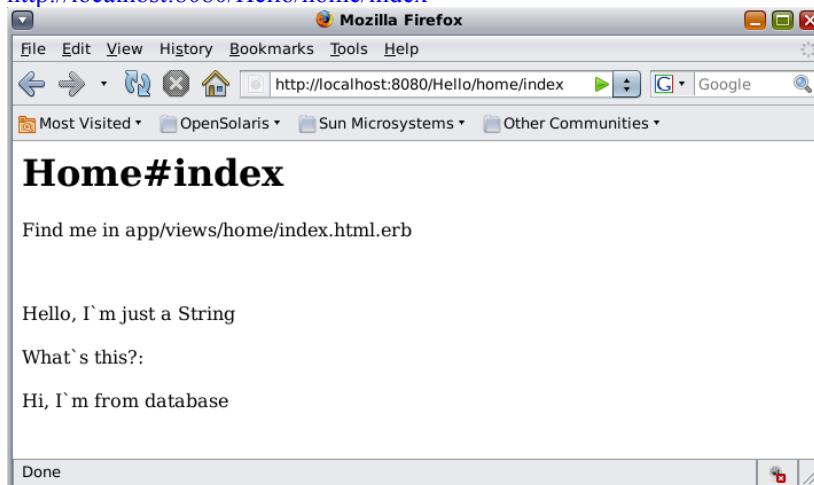
```
<BR><BR>
<%= @hello_MySql %>
```

14. **Run the application in glassfish** – Right click on the project and select "Run".
Glassfish is starting and NetBeans deploys the application to the server.
You'll see a new Firefox window pointing to <http://localhost:8080/Hello/>.
This is the root URL of our application, you're looking at the default Ruby on Rails application page.



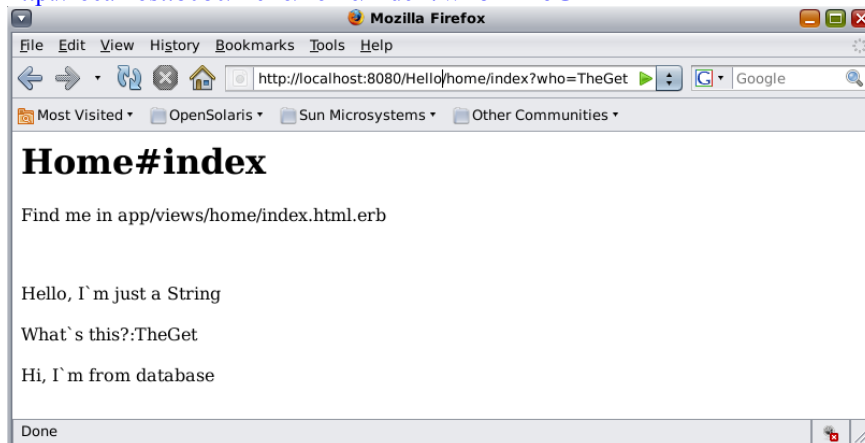
Direct your browser to the viewer to see the Strings:

<http://localhost:8080/Hello/home/index>



15. Add a GET parameter

<http://localhost:8080>Hello/home/index?who=TheGET>



16. From the "Service" tab you can expand "Servers", expand "Glassfish V3" and expand "Applications" to see your application deployed in Glassfish.
(Note: To undeploy the application, just right click on it and select "Undeploy")

