[First-time repository setup](http://ruby.railstutorial.org/book/ruby-on-rails-tutorial#sec:1.3.1.2)

1. **INITIALIZING**  
     
   Now we come to some steps that are necessary each time you create a new *repository* (which only happens once in this book, but is likely to happen again some day). First navigate to the root directory of the first app and initialize a new repository:

**$** git init

Initialized empty Git repository in /Users/mhartl/rails\_projects/first\_app/.git/

1. **ADDING – STAGING PART**

Finally, we’ll add the files in your new Rails project to Git and then commit the results. You can add all the files (apart from those that match the ignore patterns in **.gitignore**) as follows:

**$** git add .

Here the dot ‘**.**’ represents the current directory, and Git is smart enough to add the files*recursively*, so it automatically includes all the subdirectories. This command adds the project files to a *staging area*, which contains pending changes to your project; you can see which files are in the staging area using the **status** command.

git status

**#** On branch master

**#**

**#** Initial commit

**#**

**#** Changes to be committed:

**#** (use "git rm --cached <file>..." to unstage)

**#**

**#** new file: README

**#** new file: Rakefile

.

.

.

(The results are long, so I’ve used vertical dots to indicate omitted output.)

1. **ADDING – COMMINTING (actually keeping the changes)**
2. **$** git commit -m "Initial commit"
3. [master (root-commit) df0a62f] Initial commit
4. 42 files changed, 8461 insertions(+), 0 deletions(-)
5. create mode 100644 README
6. create mode 100644 Rakefile
7. .

By the way, you can see a list of your commit messages using the log command:

$ git log

commit df0a62f3f091e53ffa799309b3e32c27b0b38eb4  
Author: Michael Hartl [michael@michaelhartl.com](mailto:michael@michaelhartl.com)  
Date: Thu Oct 15 11:36:21 2009 -0700  
 Initial commit  
  
To exit git log, you may have to type q to quit.