

# Setting up a development environment for Buildr plugin

## Setting up a development environment for Buildr plugin



Content copied from <http://www.lunar-ocean.com/how-to-create-a-plugin-for-buildr/>

### Prerequisites:

- Install git.
- Install ruby, rubygems, and the braid gem.

### Basic setup:

- Create a new folder and cd to it. To create the minimal skeleton for your GEM
- run `git init && touch README && git add README && git commit -m "initial commit"`
- run `braid add git://git.apache.org/buildr.git --branch trunk buildr`
- run `ln -s buildr/rakelib rakelib`
- Create your `.gemspec` file.
- Then copy `buildr/Rakefile` to `Rakefile`, and replace the reference to `buildr.gemspec` to your `.gemspec`.
- Create your structure: a `lib`, a `spec` folder.
- run `echo "_reports" >> .gitignore`

### Specs setup:

- Create `spec/spec_helpers.rb`
- Paste this code in it:

```
unless defined?(SpecHelpers)
  module SandboxHook

    def SandboxHook.included(spec_helpers)
      # For testing we use the gem requirements specified on the buildr4osgi.gemspec
      spec = Gem::Specification.load(File.expand_path('../my.gemspec', File.dirname(__FILE__)))
      spec.dependencies.each { |dep| gem dep.name, dep.requirement.to_s }
      # Make sure to load from these paths first, we don't want to load any
      # code from Gem library.
      $LOAD_PATH.unshift File.expand_path('../lib', File.dirname(__FILE__))
      require 'mymodule'
    end
  end
end
require File.join(File.dirname(__FILE__), "../buildr/spec/spec_helpers.rb")

end
```

You're done.

Try `rake -T` to see the buildr tasks available, use `rake coverage` to run tests and `rake failed` when you need to insist.