



Workshop #6

Build a RubyGem

Abstract

RubyGems sind Container aus Code die in jede Ruby-Applikation integriert werden können. Beispiele sind

- Rails
- Sidekiq
- Whenever
- Forminator
- email_validator

rubygems.org

Das zentrale Repository für öffentliche RubyGems ist

<http://rubygems.org>

Dort kann jeder sein Rubygem veröffentlichen.

Bundler

Bundler ist ein tool, um RubyGems zu installieren. Dabei werden Dependencies automatisch aufgelöst.

<http://bundler.io/>

gem

Alternativ zu Bundler bietet RubyGems selbst einen Package-Manager. Das Programm heisst

gem

sollte aber nicht mehr genutzt werden.

Gemfile und Gemfile.lock

Um auf einfache Weise mit dem Bundler gems zu installieren erstellt man eine Datei namens `Gemfile`

Der minimale Inhalt ist:

```
source 'https://rubygems.org'  
gem 'super_gem_power_gem'
```

Um das bzw. die gems zu installieren wird bundler aufgerufen:

```
$ bundle install
```

Mehr Info unter <http://bundler.io/>

Lesen

Es ist ein wenig schwer gute Ressourcen zu finden. Zum Glück aber gibt es das hervorragende Buch *'Build a RubyGem'* von Brandon Hilkert:

<http://brandonhilkert.com/books/build-a-ruby-gem/>

Ein gem mit bundler bauen

```
$ bundle gem kenny_pics
```

```
create  kenny_pics/Gemfile
create  kenny_pics/Rakefile
create  kenny_pics/LICENSE.txt
create  kenny_pics/README.md
create  kenny_pics/.gitignore
create  kenny_pics/kenny_pics.gemspec
create  kenny_pics/lib/kenny_pics.rb
create  kenny_pics/lib/kenny_pics/version.rb
Initializing git repo in /Users/andwen/project/sumcumo/rails-workshop-s/rails-workshop-6/code/kenny_pics
```


kenny_pics erste Schritte

- .rvmrc Datei mit eigenem gemset erstellen
- .rvmrc in .gitignore packen
- cd .
- Lizenz wählen (wenn nicht MIT)
- README füllen
- initialer commit

git

Auf <http://github.com/sumcumo> ein neues
Repositroy namens kenny_pics erstellen.
Dann:

```
$ git remote add origin git@github.com:sumcumo/kenny_pics.git  
$ git push -u origin master
```

kenny_pics.gemspec

Die Konfiguration des gems. Fehlende Werte ergänzen oder ändern.

Dependencies hinzufügen:

```
spec.add_dependency "google-search"  
spec.add_development_dependency "RSpec"
```

gems für das gem installieren

```
$ bundle install
Fetching gem metadata from https://rubygems.org/.....
Resolving dependencies...
Using rake 10.4.2
Using bundler 1.7.11
Installing diff-lcs 1.2.5
Using json 1.8.2
Using google-search 1.0.3
Using kenny_pics 0.0.1 from source at .
Installing rspec-support 3.1.2
Installing rspec-core 3.1.7
Installing rspec-expectations 3.1.2
Installing rspec-mocks 3.1.3
Installing rspec 3.1.0
Your bundle is complete!
Use `bundle show [gemname]` to see where a bundled gem is installed
```

rspec

```
$ rspec --init  
create   .rspec  
create   spec/spec_helper.rb
```

In spec/spec_helper.rb

```
require "kenny_pics"
```

```
$ rspec spec  
No examples found.  
Finished in 0.00019 seconds (files took 0.08752 seconds to load)
```

Rakefile

Um die console nutzen zu können den Rakefile anpassen:

```
require "bundler/gem_tasks"  
require "rspec/core/rake_task"  
  
RSpec::Core::RakeTask.new("spec")  
  
task :default => :spec  
  
task :console do  
  exec "irb -r kenny_pics -I ./lib"  
end
```

Erster rspec test

in spec/lib/ Datei kenny_pics_spec.rb anlegen:

```
require 'spec_helper'
```

```
RSpec.describe KennyPics do  
  it 'is described module' do  
    expect(described_class).to eq(KennyPics)  
  end  
end
```

```
$ rspec spec
```

```
.  
Finished in 0.00077 seconds (files took 0.16143 seconds to load)  
1 example, 0 failures
```

KennyPics::Search

code!

gem lokal installieren

```
$ rake -T
rake build      # Build kenny_pics-0.0.1.gem into the pkg directory
rake install   # Build and install kenny_pics-0.0.1.gem into system gems
rake release   # Create tag v0.0.1 and build and push kenny_pics-0.0.1.gem to Rubygems
rake spec      # Run RSpec code examples
```

als cli nutzen

im Ordner bin/ die Datei kenny_pics erstellen

```
#!/usr/bin/env ruby
```

```
require_relative '../lib/kenny_pics/search'
```

```
puts KennyPics::Search.run
```

```
$ git ci -a -m 'bin and cli data'
```

```
$ chmod +x bin/*
```

```
$ rake install
```

```
$ kenny_pics
```

```
http://upload.wikimedia.org/wikipedia/en/6/6f/KennyMcCormick.png
```

```
http://images.buddytv.com/btv\_2\_2324\_1\_434\_593\_0\_/kenny-mccormick-phot.jpg
```

```
...
```

In Rails views

in lib die Datei railtie.rb erstellen. Diese wird beim Start von Rails geladen:

```
require 'rails'
require 'kenny_pics'

module KennyPics
  # load KennyPics into ActionView during the Rails start
  class Railtie < Rails::Railtie
    initializer 'kenny_pics.action_view' do
      ActiveSupport.on_load(:action_view) do
        include KennyPics::Helper
      end
    end
  end
end
```

Rails Gemfile

```
gem 'kenny_pics', path: '../kenny_pics'
```

```
$ bundle install
```

```
...
```

```
Using kenny_pics 0.0.1 from source at ../kenny_pics
```

```
...
```

In Rails view

Hallo Kenny

```
<% kenny_pics_search.each do |img| %>  
    
<% end %>
```

And

<http://localhost:3000>

FIN

“OH NO - THEY KILLED KENNY”