

# README 文档版

GitHub 地址: <https://github.com/AndyDu0116/AdvancedSE>

## 一、Part0 Demystifying SaaS app creation

本部分的目标是熟悉理解用 Ruby 语言的 Sinatra 框架创建一个 SaaS 应用，并加入版本控制，最后部署。本实验作业需要的环境有：Ruby, Git。

## 1.1 实验过程

1)创建 Gemfile 文件来指示应用需要依赖的 gem 包和相关的资源地址,该文件名不能是其他,否则无法实现自动化。

```

'''
#Gemfile 文件
source 'https://rubygems.org'
ruby '2.6.6'

gem 'sinatra', '>= 2.0.1'
'''

```

2)使用 Bundler 来管理 Gem 包之间的依赖关系,在创建的目录下执行`Bundle`命令即可。

3)创建 Web 页面响应文件。

```

`
#app.rb 文件
require 'sinatra'
class MyApp < Sinatra::Base
  get '/' do
    "<!DOCTYPE html><html><head></head><body><h1>Hello,World</h1>
</body> </html>"
  end
end
`

```

#### 4)创建项目的配置文件

```
'''
#config.ru 配置文件
require './app'
run MyApp
'''
```

5)使用版本控制工具 `git` 对项目进行版本控制

```
...  
git init  
git add .  
git commit -m "Set up the Gemfile"  
...
```

6)部署运行，配置好依赖环境和编写完上述代码后就能够部署运行，在本地可以使用``bundle exec rackup --port 3000``命令运行

7)使用 `rerun Gem` 包来实现页面根据修改动态更新。因为只有在开发的过程中才会使用该特性，所以只需要在开发的模式下使用该 `Gem` 即可：

```
...  
#在 Gemfile 中添加如下依赖说明  
group :development do  
  gem 'rerun'  
end  
...
```

## 1.2 遇到的问题

1)Ruby 版本问题，我的电脑最初安装的是 Ruby3.0.2 版本，在修改 Gemfile 文件中 ruby 版本后对然 Bundle 检查通过，但是在执行过程中出错，是因为依赖包的问题。最终的解决方法是将 Ruby 降级为 2.6.6，最终成果执行。

```
F:\研一上\高级软件工程\code\SinatraNewApp>bundle exec rackup --port 3000  
D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/handler.rb:45:in `pick': Couldn't find handler for: puma, thin, f  
alcon, webrick. (LoadError)  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/handler.rb:60:in `default'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:334:in `server'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:255:in `block in logging_middleware'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:413:in `block in build_app'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:412:in `reverse_each'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:412:in `build_app'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:422:in `wrapped_app'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:312:in `block in start'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:379:in `handle_profiling'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:311:in `start'  
from D:/Ruby30/lib/ruby/gems/3.0.0/gems/rack-2.2.3/lib/rack/server.rb:168:in `start'  
from D:/Ruby30/bin/rackup:5:in `<top (required)>'  
from D:/Ruby30/bin/rackup:23:in `load'  
from D:/Ruby30/bin/rackup:23:in `<main>'  
F:\研一上\高级软件工程\code\SinatraNewApp>ruby list  
ruby: No such file or directory -- list (LoadError)
```

图 1 Gem 包依赖问题：

程序的运行结果：



图 2 运行结果

## 二、Part1 Wordguesser

此部分是学习使用测试驱动开发的流程(test-driven development ,TDD), 已经提供了猜词相关业务的场景描述, 需要根据 wordguesser\_game\_spec.rb 中的测试描述来编写 wordguesser\_game.rb 和 app.rb 中的缺失代码, 目标是通过给定的自动化测试。

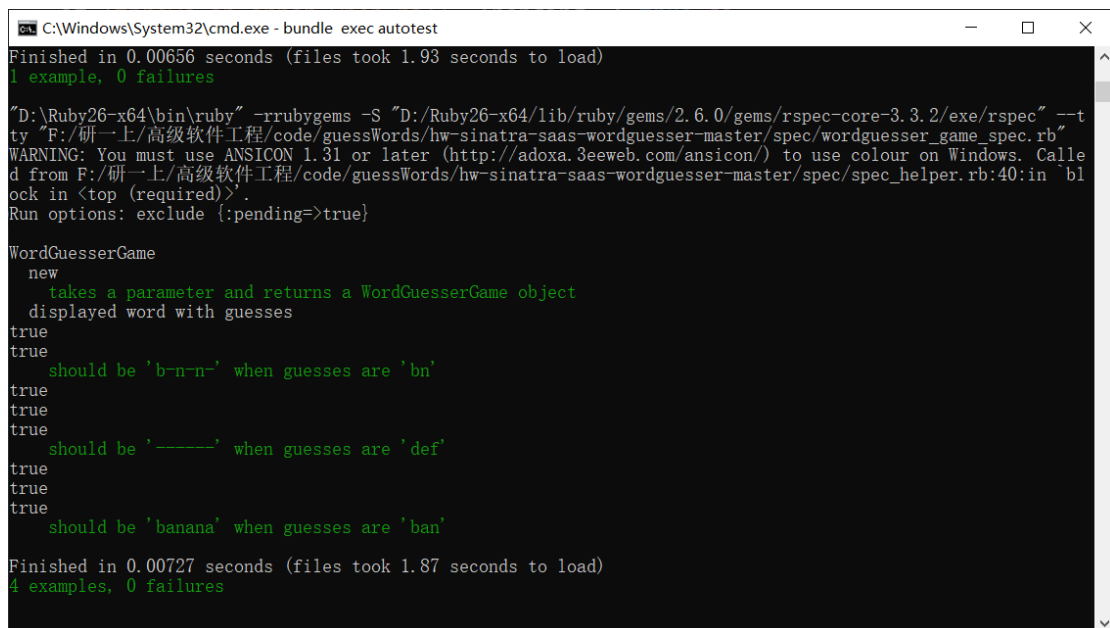
### 2.1 实验过程

1)克隆 clone 代码到本地以便开发, 然后使用 Bundle 命令来下载相应的依赖包。此步骤使用命令:

```
...
git clone https://github.com/saasbook/hw-sinatra-saas-wordguesser
cd hw-sinatra-saas-wordguesser
bundle
...
```

2) 依据 wordguesser\_game\_spec.rb 中队场景的描述, 编写补全 wordguesser\_game.rb 和 app.rb 中缺失的代码, 编写代码过程不再赘述。

最终 4 个 examples 均通过了自动化测试, 如下图所示:



```
C:\Windows\System32\cmd.exe - bundle exec autotest
Finished in 0.00656 seconds (files took 1.93 seconds to load)
1 example, 0 failures

"D:\Ruby26-x64\bin\ruby" -rrubygems -S "D:/Ruby26-x64/lib/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --t
ty "F:/研一上/高级软件工程/code/guessWords/hw-sinatra-saas-wordguesser-master/spec/wordguesser_game_spec.rb"
WARNING: You must use ANSICON 1.31 or later (http://adoxa.3eeweb.com/ansicon/) to use colour on Windows. Calle
d from F:/研一上/高级软件工程/code/guessWords/hw-sinatra-saas-wordguesser-master/spec/spec_helper.rb:40:in `bl
ock in <top (required)>'.
Run options: exclude {:pending=>true}

WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object
  displayed word with guesses
    true
    true
    should be 'b-n-n-' when guesses are 'bn'
    true
    true
    true
    should be '-----' when guesses are 'def'
    true
    true
    true
    should be 'banana' when guesses are 'ban'

Finished in 0.00727 seconds (files took 1.87 seconds to load)
4 examples, 0 failures
```

图3 系统化测试结果

## 2.2 遇到的问题

1)gem 包依赖问题,我使用的操作系统是 Windows10,将该项目下载到本地使用 Bundle 下载依赖包时报错,尽管我的 Ruby 版本和项目指定的一致都是 2.6.6 但是还是出现了问题。在 <https://rubygems.org/>网站查找后发现:nokogiri-1.8.4-x64-mingw32 这个包的 1.8.4 版本在 Linux 系统中是依赖 Ruby>=2.1.0 而在 Windows 系统中的版本是依赖 Ruby>=2.2.0,<2.6。这就导致了依赖错误,解决方法是使用 Bundle update 命令,此命令的作用是:更新系统中指定的 gem 包信息,并同时更新项目 Gemfile.lock 中指定的包信息,这样就能自动地更新相应依赖关系,解决依赖错误问题。

```
F:\研一上\高级软件工程\code\guessWords\hw-sinatra-saas-wordguesser-master>bundle
Warning: the running version of Bundler (1.17.2) is older than the version that created the lockfile (1.17.3).
We suggest you upgrade to the latest version of Bundler by running `gem install bundler`.
Fetching source index from https://rubygems.org/
Resolving dependencies....
nokogiri-1.8.4-x64-mingw32 requires ruby version >= 2.2, < 2.6, which is 依赖出错
incompatible with the current version, ruby 2.6.6p146

F:\研一上\高级软件工程\code\guessWords\hw-sinatra-saas-wordguesser-master>bundle update
Warning: the running version of Bundler (1.17.2) is older than the version that created the lockfile (1.17.3).
We suggest you upgrade to the latest version of Bundler by running `gem install bundler`.
Fetching source index from https://rubygems.org/
Resolving dependencies.....
Fetching ZenTest 4.11.2
Installing ZenTest 4.11.2
Fetching public_suffix 4.0.6 (was 3.0.3)
```

图 4 依赖报错

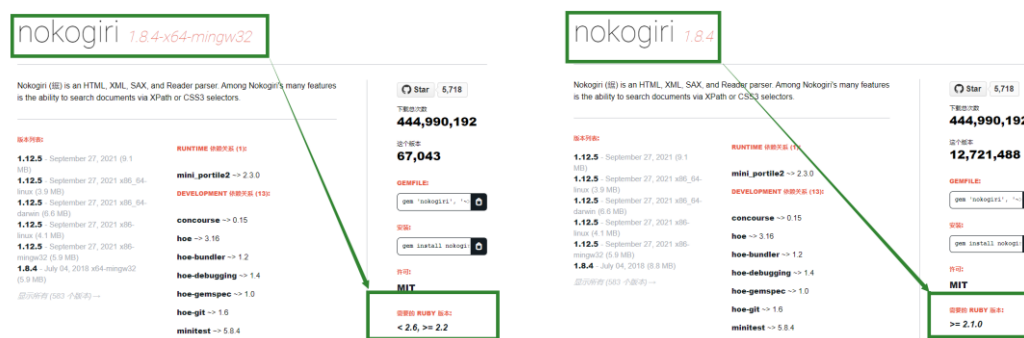


图 5 不同 OS 依赖差异

2)程序代码问题,因为 Ruby 是新学的一门语言,在编写代码时就出现了较多问题:语法错误、逻辑错误、效率慢等。刚开始使用自动化测试框架时,因为有 pending => true 检查跳过,所以没有 example (图 6,当删除后就出现了 1 个 example 但是没有通过测试,如图 6)。

```
F:\研一上\高级软件工程\code\guessWords\hw-sinatra-saas-wordguesser-master>bundle exec autotest
(Not running features. To run features in autotest, set AUTOFEATURE=true.)
loading autotest/rspec
'D:/Ruby26-x64/bin/ruby' -rrubygems -S "D:/Ruby26-x64/lib/ruby/gems/2.6.0/gems/rspec-core-3.3.2/exe/rspec" --t
ty "F:/研一上/高级软件工程/code/guessWords/hw-sinatra-saas-wordguesser-master/spec/wordguesser_game_spec.rb"
WARNING: You must use ANSICON 1.31 or later (http://adoxa.3eeweb.com/ansicon/) to use colour on Windows. Calle
d from F:/研一上/高级软件工程/code/guessWords/hw-sinatra-saas-wordguesser-master/spec/spec_helper.rb:40:in `bl
ock in <top (required)>'
Run options: exclude {:pending=>true}

All examples were filtered out

Finished in 0.00048 seconds (files took 2.21 seconds to load)
0 examples, 0 failures
```

图 6 Failed AutoTest

```
WordGuesserGame
  new
    takes a parameter and returns a WordGuesserGame object (FAILED - 1)
Failures:
  1) WordGuesserGame new takes a parameter and returns a WordGuesserGame object
     Failure/Error: expect(@game.word).to eq('glorp')
     NoMethodError:
       undefined method `word' for #<WordGuesserGame:0x00000000070e5ab8 @word="glorp">
       # ./spec/wordguesser_game_spec.rb:16:in `block (3 levels) in <top (required)>'
Finished in 0.00386 seconds (files took 1.87 seconds to load)
1 example, 1 failure
```

图 7 Passed AutoTest

### 三、Part2 RESTful thinking for Wordguesser

此部分讲解了如何将 Wordguesser 游戏设计为 REST Service 的思想，主要是解决 HTTP 这一无状态的协议如何保存用户或游戏的状态。该问题的解决方法有：使用 cookie，在用户端记录相关信息并在需要时发送给服务器。

另一方面，在本部分讨论了服务器的资源服务和 HTTP requests 的映射问题，以及可以使用的方法。本项目仅讨论了 GET 和 POST 方法，大致可以分为：无状态需求时使用 GET 方法，需要状态相关数据时使用 POST 方法。详细的设计分类可见下表：

表 1 服务 GET/POST 方法对照表

展示游戏状态界面	GET/show
跳转到 create 处理	Get/new
开始一个新游戏，跳转到展示游戏状态界面	POST/create
处理猜词结构，当用游戏状态改变后跳转到展示游戏状态页面	POST/guess
返回用户胜利界面并带有开始新游戏的按钮	GET/win
返回用户失败界面并带有开始新游戏的按钮	GET/lose

## 四、Part3 Connecting Wordguesser to Sinatra

此部分是将上一阶段的设计结果形成代码实现，最终部署到服务器以获取真实的运行结果，即一个和既定描述相符的猜词游戏。

### 4.1 实验过程

1)在 Part2 已经对各个页面/状态进行了定义，本阶段需要做的就是将设计的逻辑实现并得到可运行的程序。如图 8 所示为游戏的初始界面，点击“New Game”按钮可以在服务器端开始一个新游戏，这样便会跳转到游戏状态界面，如图 9 所示。图 10 是玩家失败的界面。

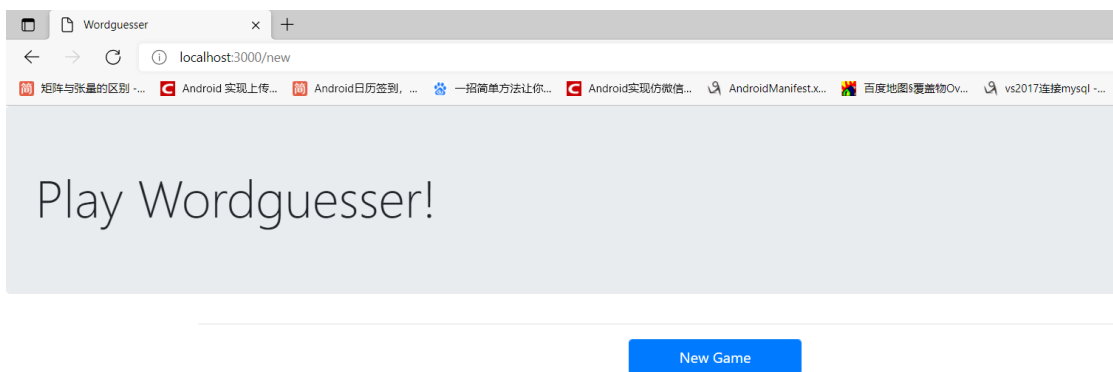


图 8 Wordguesser 初始界面

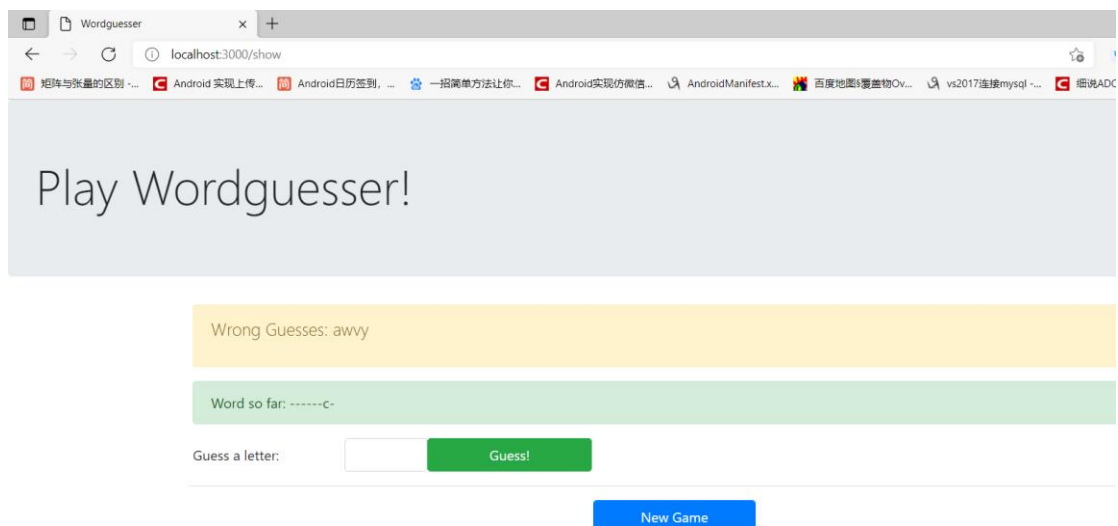


图 9 Wordguesser 游戏状态展示

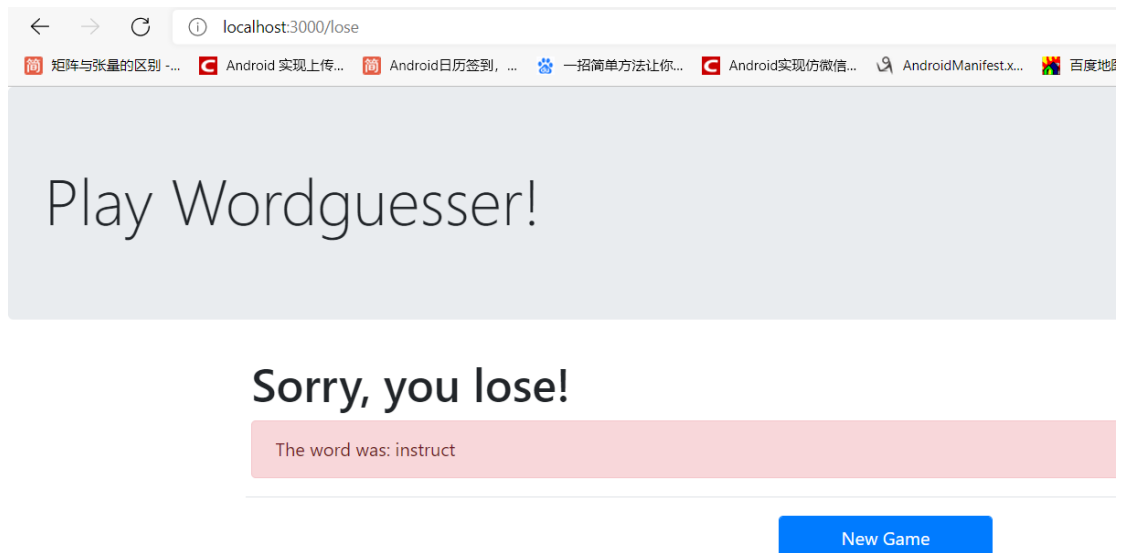


图 10 游戏失败界面

## 4.2 遇到的问题

在运行时由于电脑没有联网导致程序在运行到需要网络的语句时崩溃，这里需要网络的操作是获取一个随机词汇。可以做的优化是在运行出现网络问题时使用特定的单词。



图 11 系统因网络错误崩溃

## 五、Part4 Introducing Cucumber

本部分要求使用 cucumber 做自动化测试，测试的结果如图 12 所示。

```
D:/Ruby20-RC4-116/Ruby/2.0.0-RC4-mingw32: Reverting to Ruby lexer.
Feature: start new game

  As a player
  So I can play Wordguesser
  I want to start a new game

Scenario: I start a new game # features/start_new_game.feature:7
  Given I am on the home page # F:/研一上/高级软件工程/code/ReferDemo/hw-sinatra-saas-wordguesser-master/features/step_definitions/game_steps.rb:66
  And I press "New Game" # F:/研一上/高级软件工程/code/ReferDemo/hw-sinatra-saas-wordguesser-master/features/step_definitions/game_steps.rb:89
  Then I should see "Guess a letter" # F:/研一上/高级软件工程/code/ReferDemo/hw-sinatra-saas-wordguesser-master/features/step_definitions/game_steps.rb:79
  And I press "New Game" # F:/研一上/高级软件工程/code/ReferDemo/hw-sinatra-saas-wordguesser-master/features/step_definitions/game_steps.rb:89
  Then I should see "Guess a letter" # F:/研一上/高级软件工程/code/ReferDemo/hw-sinatra-saas-wordguesser-master/features/step_definitions/game_steps.rb:79

1 scenario (1 passed)
5 steps (5 passed)
0m0.199s
```

图 12 cucumber 自动化测试通过