How to setup and run ruby remote debug session

In case you have a script that you'd like to debug but cannot run locally, e.g. it requires access to a database, which is located only in production, RubyMine provides an ability to connect to the remote debug session from the local host and use all benefits of the graphical built-in debugger.

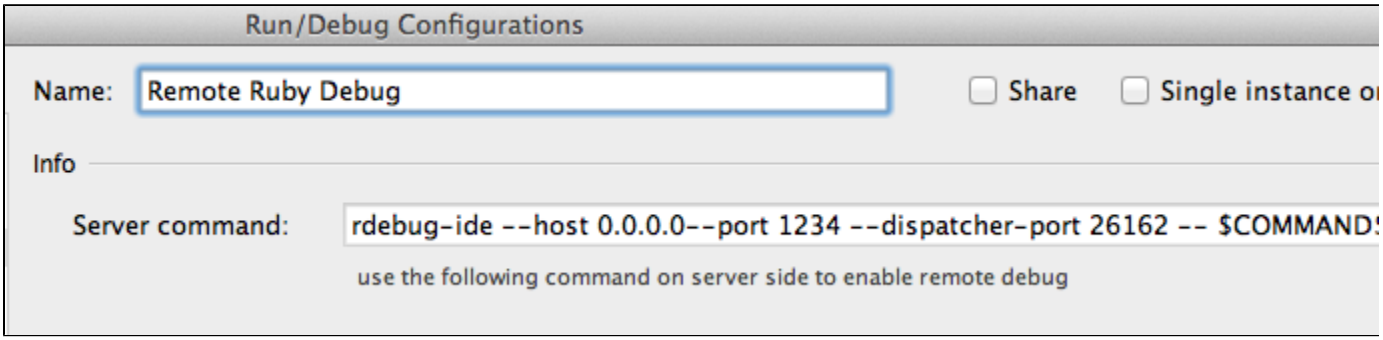
In this tutorial we will use RubyMine 7.1.2 and demo-script to illustrate basic steps. Please clone sample from Github.

0. Starting remote debug session on the remote host

The common view of the command is:

```
rdebug-ide --host 0.0.0.0 --port 1234 --dispatcher-port 26162 -- $COMMAND$
```

This command is also shown in Run/Debug Configuration as a tip:



So in order to do it we have to run SSH session to the remote host. Open Tools | Start SSH session..., add remote host using E dit Credentials option, specify port, user/password and click OK.

Remote host connection will open in the Terminal and we can run remote debug session:

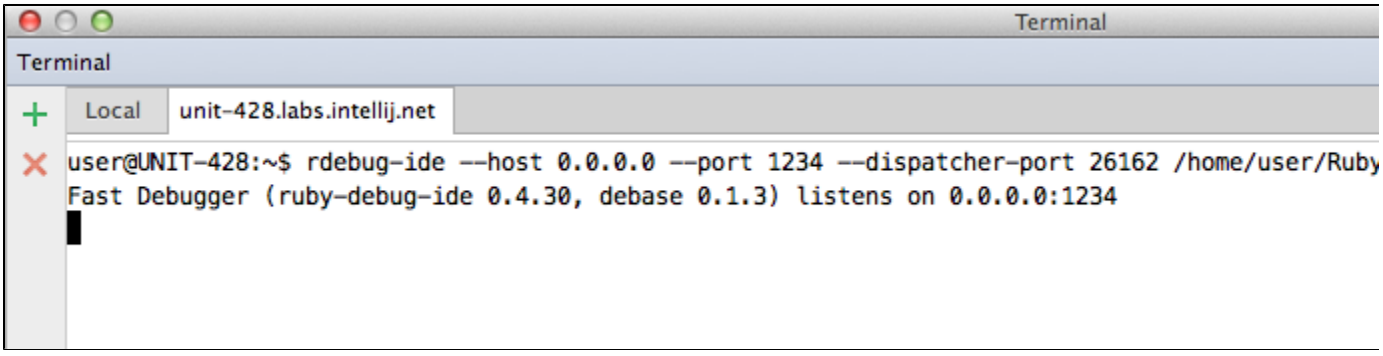
```
~$ rdebug-ide --host 0.0.0.0 --port 1234 --dispatcher-port 26162
/home/user/RubymineProjects/remote-debug-example/math_wiz.rb
```

If you have any problems with debugging or prefer a full log, it might be useful to start a session with an additional debugger log: add a parameter ---debug to the command:

```
rdebug-ide --host 0.0.0.0 --port 1236 --dispatcher-port 26167 --debug
/home/user/RubymineProjects/remote-debug-example/math_wiz.rb
```

In this command:
port (1234) - port on the remote host
dispatcher-port (26162) - port on your localhost

Now we have remote debug session started.

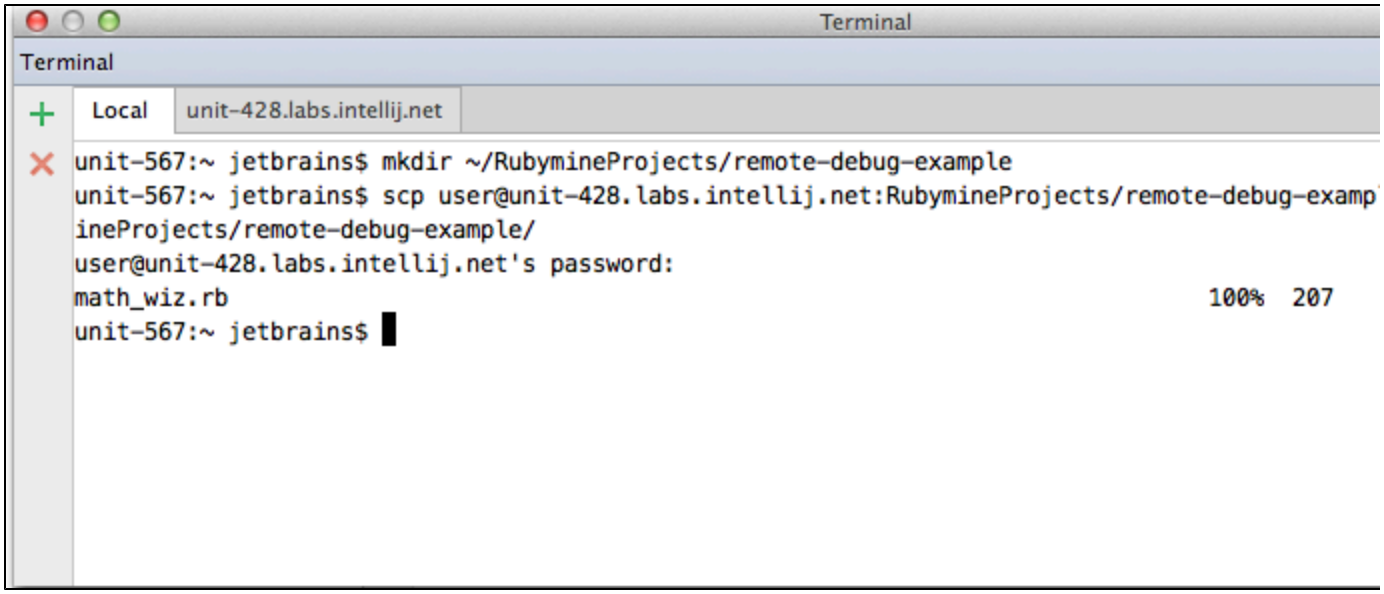


1. Creating a local copy of the script in question

In order to open our script in RubyMine to set breakpoints and finally start debugging, we need to copy it to the local host. The fastest way would be to switch to the local tab (or open a new one) in the Terminal and copy our script:

```
#create a local folder
~$ mkdir ~/RubymineProjects/remote-debug-example

#copy file to the local folder
~$ scp user@remote-host:RubymineProjects/remote-debug-example/math_wiz.rb
~/RubymineProjects/remote-debug-example/
```

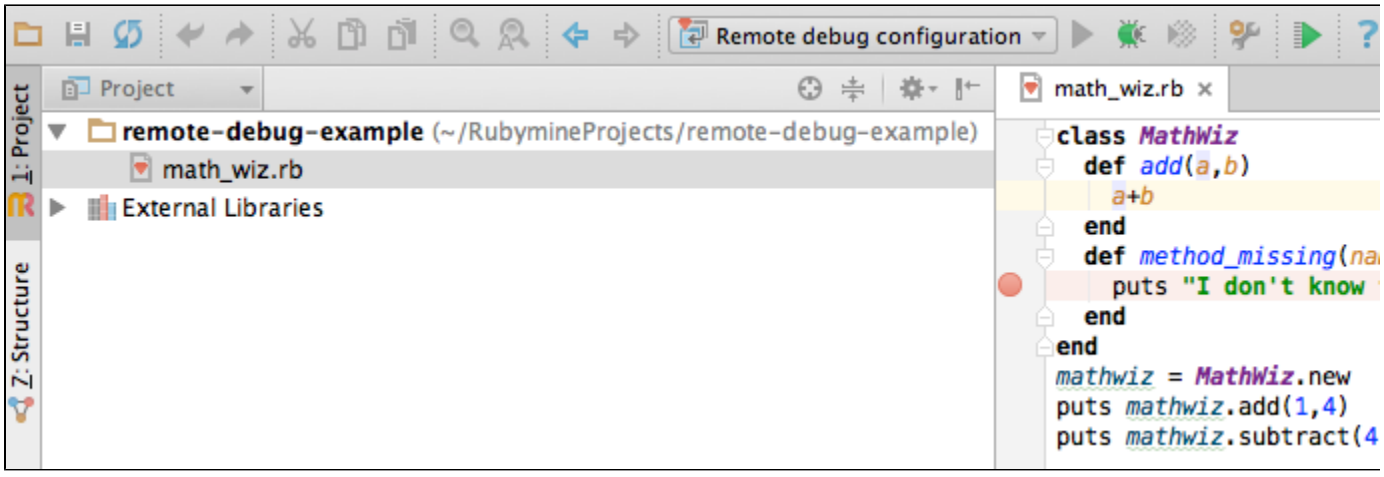


Then open the folder ~/RubymineProjects/remote-debug-example with script in RubyMine. It will create a new project and RubyMine will suggest you to configure a Ruby interpreter for it.

In case you need to debug some internal(native) Ruby code, it would be better to use the same environment as on the remote host (you could select the same Ruby SDK in the Settings | RubyMine SDK and Gems).

2. Configuring breakpoints

Just click on the left side of the code line in Editor to set a breakpoint (or use shortcut Cmd+F8).



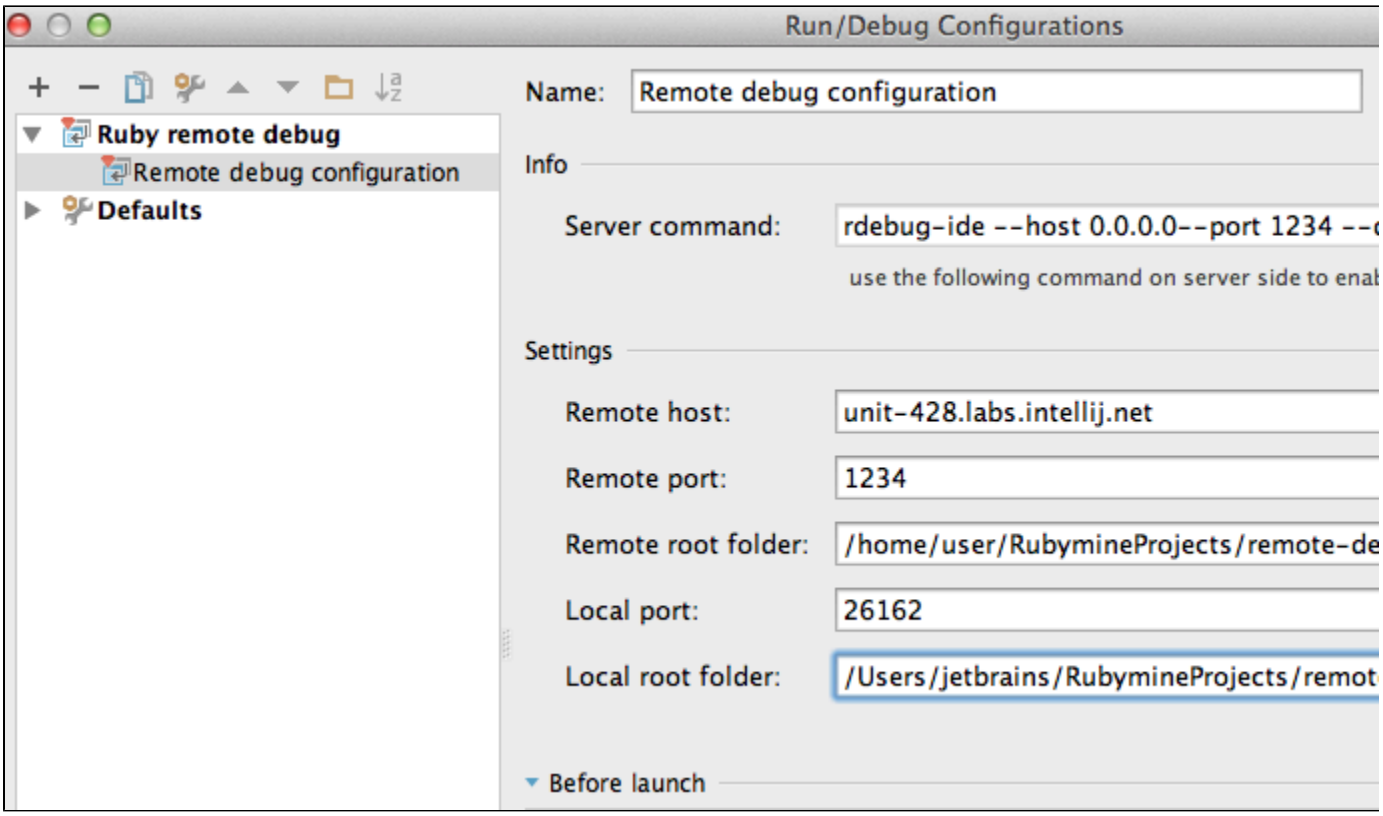
3. Configuring the Ruby Remote Debug Run/Debug configuration

- Open Run/Edit Configurations...
- Click Add New configuration... button ('+') and add Ruby remote debug configuration

- Specify its name, i.e. 'Remote debug configuration'

Then set other parameters in the configuration:

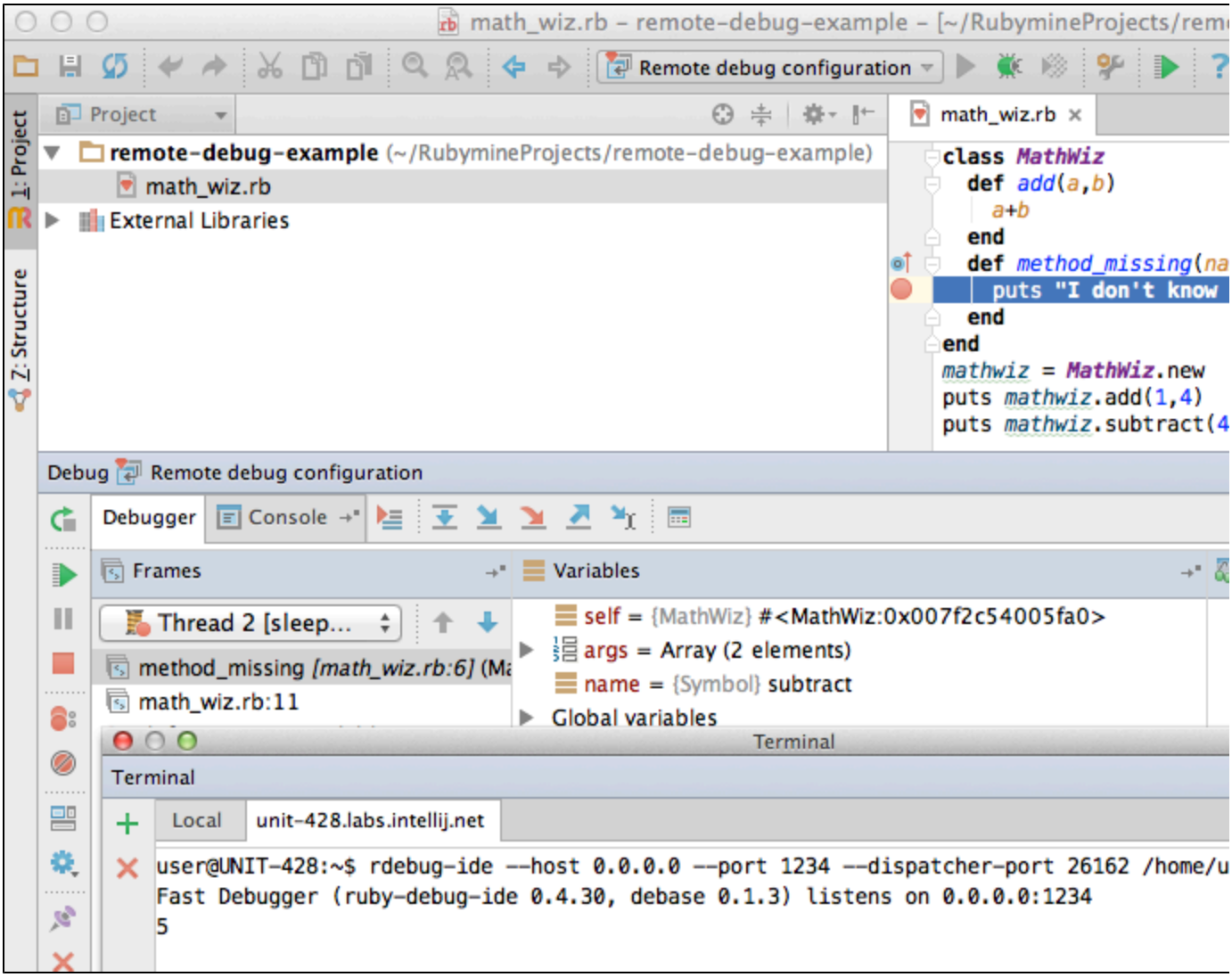
- Remote host - name of the remote host
- Remote port - should be the same as we used in the step 1 as 'port'
- Remote root folder - remote path to the script, e.g. '/home/user/RubymineProjects/remote-debug-example/'
- Local port - the same as port 'dispatcher'
- Local root folder - folder where the copied script is located on the local host. e.g. /Users/jetbrains/RubymineProjects/remote-debug-example. To check yourself: it should be the same folder as the one we opened during the step 2.



Save configuration with Apply and OK buttons. Now we are ready to connect to the remote session and debug the script.

4. Running remote debug configuration in debug mode

Run 'Remote debug configuration' in debug mode (please note that only this mode is available). Local debugger connects to the remote session and stops on the breakpoint that we've set locally in our script.



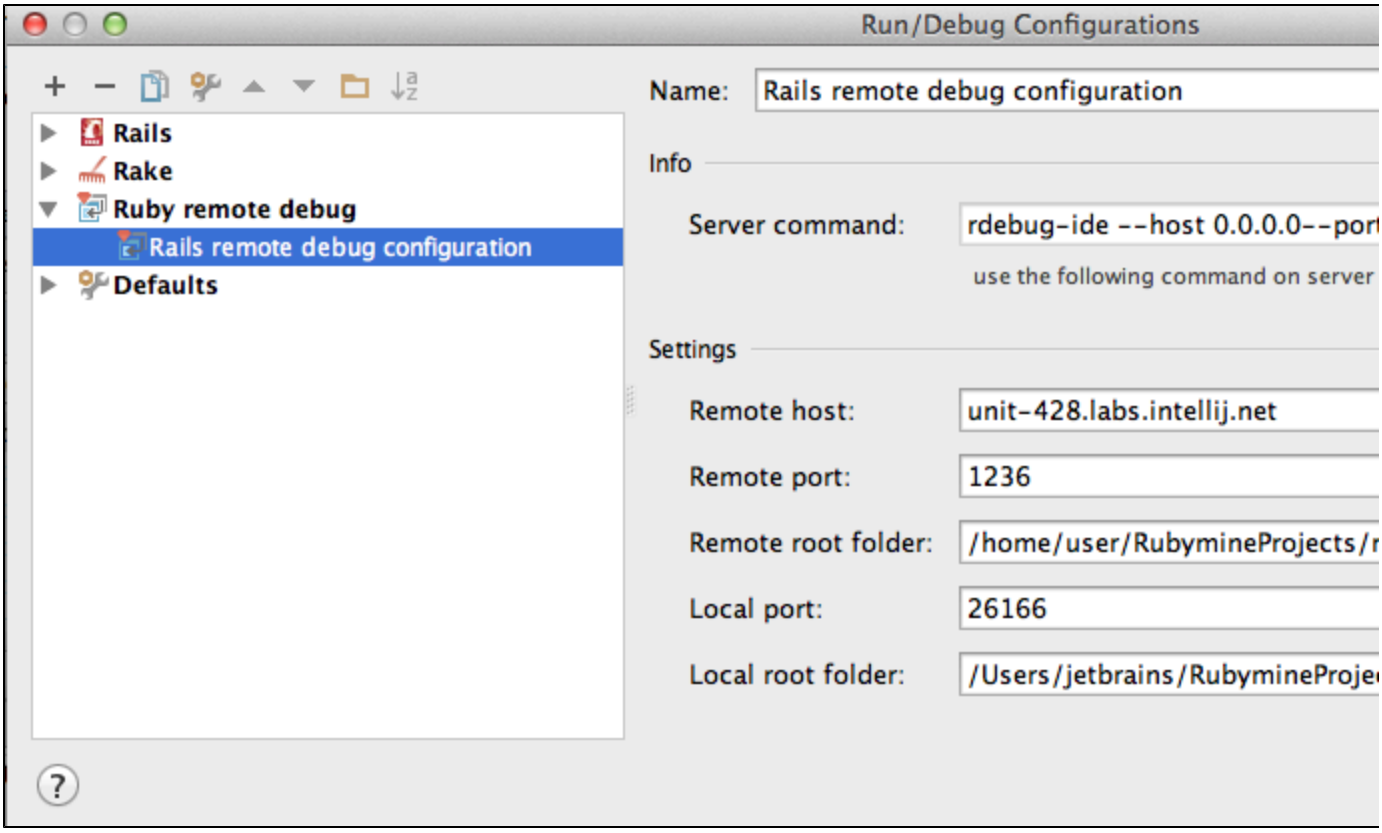
5. Running remote debug for Rails applications

It’s possible to use this way of debugging in case of Rails applications as well.

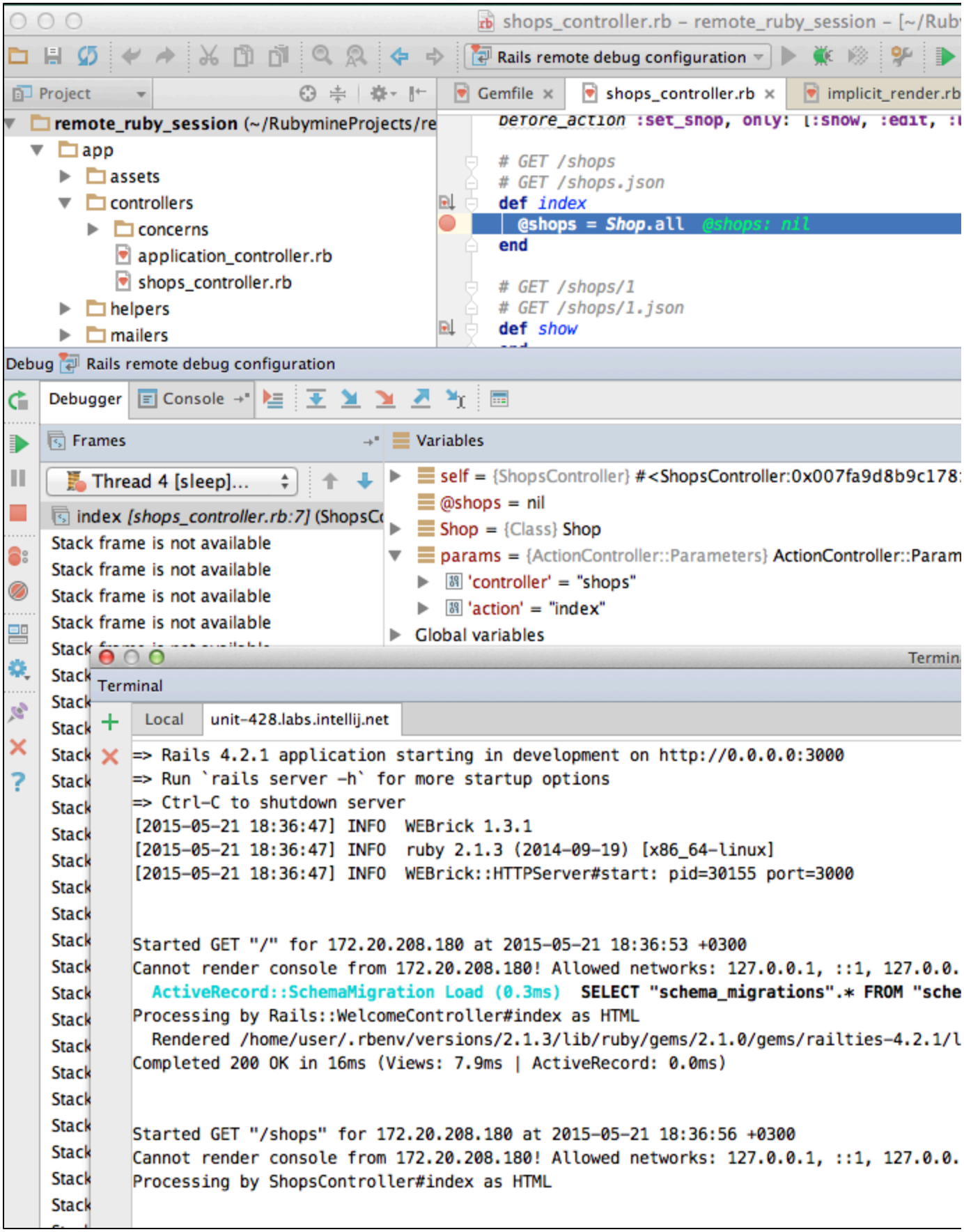
For example, the command which starts debug session could be:

```
~$ rdebug-ide --port 1236 --dispatcher-port 26166 --host 0.0.0.0 bin/rails – s -b 0.0.0.0
```

Run/Debug Configuration example:



Then we can run Rails remote debug configuration in debug mode and stop on the breakpoint using remote debug session:


shops_controller.rb – remote_ruby_session – [~/Rub
Rails remote debug configuration
Project
Gemfile
shops_controller.rb
implicit_render.rb
remote_ruby_session (~/RubymineProjects/re
app
assets
controllers
concerns
application_controller.rb
shops_controller.rb
helpers
mailers
before_action :set_shop, only: [:show, :edit,
GET /shops
GET /shops.json
def index
@shops = Shop.all @shops: nil
end
GET /shops/1
GET /shops/1.json
def show
Debug Rails remote debug configuration
Debugger
Console
Frames
Variables
Thread 4 [sleep]...
index [shops_controller.rb:7] (ShopsC
Stack frame is not available
Stack frame is not available
Stack frame is not available
Stack frame is not available
self = {ShopsController} #<ShopsController:0x007fa9d8b9c178
@shops = nil
Shop = {Class} Shop
params = {ActionController::Parameters} ActionController::Param
'controller' = "shops"
'action' = "index"
Global variables
Terminal
Local
unit-428.labs.intellij.net
=> Rails 4.2.1 application starting in development on http://0.0.0.0:3000
=> Run `rails server -h` for more startup options
=> Ctrl-C to shutdown server
[2015-05-21 18:36:47] INFO WEBrick 1.3.1
[2015-05-21 18:36:47] INFO ruby 2.1.3 (2014-09-19) [x86_64-linux]
[2015-05-21 18:36:47] INFO WEBrick::HTTPServer#start: pid=30155 port=3000
Started GET "/" for 172.20.208.180 at 2015-05-21 18:36:53 +0300
Cannot render console from 172.20.208.180! Allowed networks: 127.0.0.1, ::1, 127.0.0.
ActiveRecord::SchemaMigration Load (0.3ms) SELECT "schema_migrations".* FROM "sche
Processing by Rails::WelcomeController#index as HTML
Rendered /home/user/.rbenv/versions/2.1.3/lib/ruby/gems/2.1.0/gems/railties-4.2.1/l
Completed 200 OK in 16ms (Views: 7.9ms | ActiveRecord: 0.0ms)
Started GET "/shops" for 172.20.208.180 at 2015-05-21 18:36:56 +0300
Cannot render console from 172.20.208.180! Allowed networks: 127.0.0.1, ::1, 127.0.0.
Processing by ShopsController#index as HTML