

Doc Generator for ruby scripts

Friday, March 31, 2017 6:50 PM

Subject	Doc Generator for ruby scripts
From	Santhoshi Chepuru
To	AppQA Hyderabad Automation Team
Sent	Wednesday, August 09, 2017 11:07 PM

Hi All,

There is one gem with name '**yard**' which is a HTML Document Generator tool for ruby scripts which is more similar to JsDoc. It is very easy to use as we just need to include @tagname before the names/description.

Yard will identify the comments that are before def/class/module and generate the Documentation which will be more similar to Documentation of Watir.

Command to generate the doc: **yardoc 'filename.rb'**

We can run the above command in the folder which contains the ruby files. It will generate a folder with name docs that contains filename.html files. We also have index.html generated in the docs folder which will contain all the hyperlinks to .html files present in that folder

There is also one more gem with name 'RDoc' but the way of writing tag names is different. Yard has a nicer syntax than RDoc that will be familiar to scripters coming from other languages example protractor-javascript(JSDoc- Doc Generator)

Yard Link: <http://www.rubydoc.info/gems/yard>

Cheat Sheet Link: [Tags-CheatSheet](#)

One Example is:

```
# getGlyphIcon
# @param [Object] driver driver object
# @param [String] iconName Name of the icon required
# @note This function will return a icon reference Back/Forward
# @return [Object] Icon Reference
def VRSMainPage.getGlyphIcon(driver,iconName)
  begin
    Log.Utility("VRSMainPage::getGlyphIcon")

    # Return if driver and xpath parameters are not passed
    if driver.nil? || iconName.nil?
      Log.Warning "Passed invalid paramters in VRSMainPage::getGlyphIcon Utility"
      return false
    end

    iconRef = "//*[@title='#{iconName}']"
    return UtilsCommon.GetElementWithXPath(driver, iconRef,20, true, iconName, true)

  rescue Exception => e
    Log.Error e
    return false;
  ensure
    Log.CloseUtility
  end
end
```

Regards,
Santhoshi Chepuru
Automation Analyst II, QA Automation
FactSet Systems India Pvt. Ltd.
csanthoshi@factset.com
www.factset.com



FACTSET) SEE THE ADVANTAGE

FINANCIAL DATA | ANALYTICS | TECHNOLOGY | SERVICES

Connecting to specific node on Selenium-Grid

Friday, August 11, 2017 2:42 PM

Hi [@CharuvakaThallapelli](#)

If you want to connect to those nodes directly from your local machine. You need to do this way in your script (Ruby):

```
caps = Selenium::WebDriver::Remote::Capabilities.chrome(version: '54', platform: 'WINDOWS', :chrome_binary => 'C:\selenium\chromes\Chrome54\chrome.exe')
```

#this is the chrome binary path on tellusgridb09, not on your local machine.

```
wd = Watir::Browser.new(:remote, url: "http://tellusgridb09.pc.factset.com:5580/wd/hub", :desired_capabilities => caps)
```

Hope this will be helpful to you!

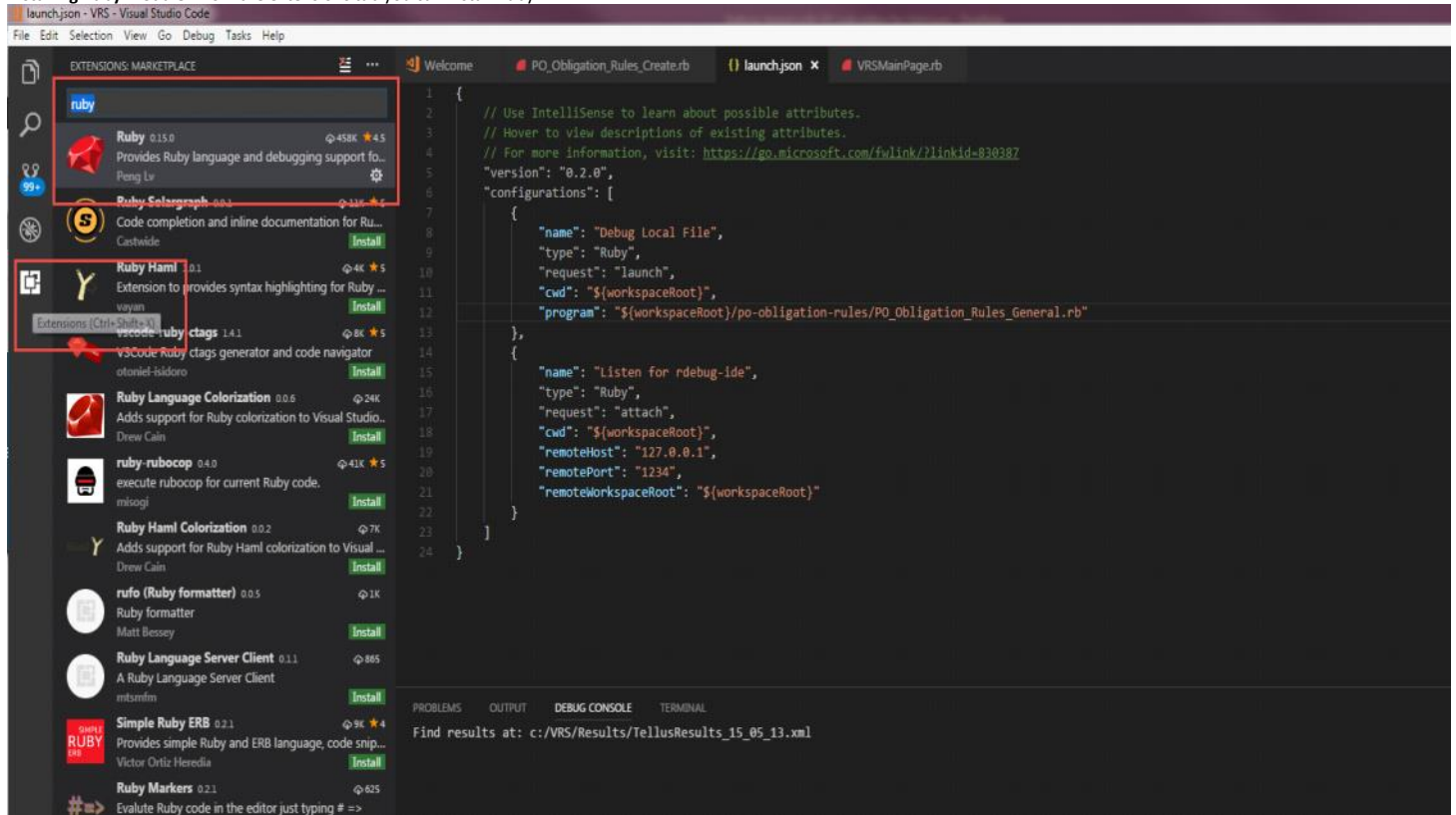
Thanks,
Lilun

From <<http://is.factset.com/rpd/summary.aspx?messageId=33312120&commentId=33341247>>

Getting started with VS code editor for Selenium

Thursday, October 12, 2017 4:52 PM

- Install Visual studio code.
- Open your project in VSC editor.
- **Installing Ruby Module:** From the extensions tab you can install Ruby.



- Configuration Setup for running the file:

In Debug section click on the cogwheel button and (select Ruby) edit the **launch.json**

```
// Use IntelliSense to learn about possible attributes.
// Hover to view descriptions of existing attributes.
// For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
"version": "0.2.0",
"configurations": [
  {
    "name": "Debug Local File",
    "type": "Ruby",
    "request": "launch",
    "cwd": "${workspaceRoot}",
    "showDebuggerOutput": true,
    "program": "${workspaceRoot}/po-obligation-rules/PO_Obligation_Rules_General.rb" || "${file}"
  },
  {
    "name": "Listen for rdebug-ide",
    "type": "Ruby",
    "request": "attach",
    "cwd": "${workspaceRoot}",
    "remoteHost": "127.0.0.1",
    "remotePort": "1234",
    "remoteWorkspaceRoot": "${workspaceRoot}"
  }
]
```

- User Settings Explanation:

- **"cwd"**: Current Working directory
- **"program"**: The location of the script to run. We can give the location of the script or we can also specify as "\${file}" to run the currently opened file.
- **"showDebuggerOutput"**: This value should be declared as true if you want to debug the script.
- To run your project by clicking on the **Run** button in **DEBUG** section. I observed that there is no separate buttons for run and debug. Shortcut for triggering the run is **F5**.
- To install the keyboard shortcuts same as WebStorm you can install **IntelliJ IDEA Key Bindings for Visual Studio Code** from the Extensions tab.
- To get the automatic code completion. Ex. Type begin and then do Ctrl+Space, select begin then it automatically populate the code like

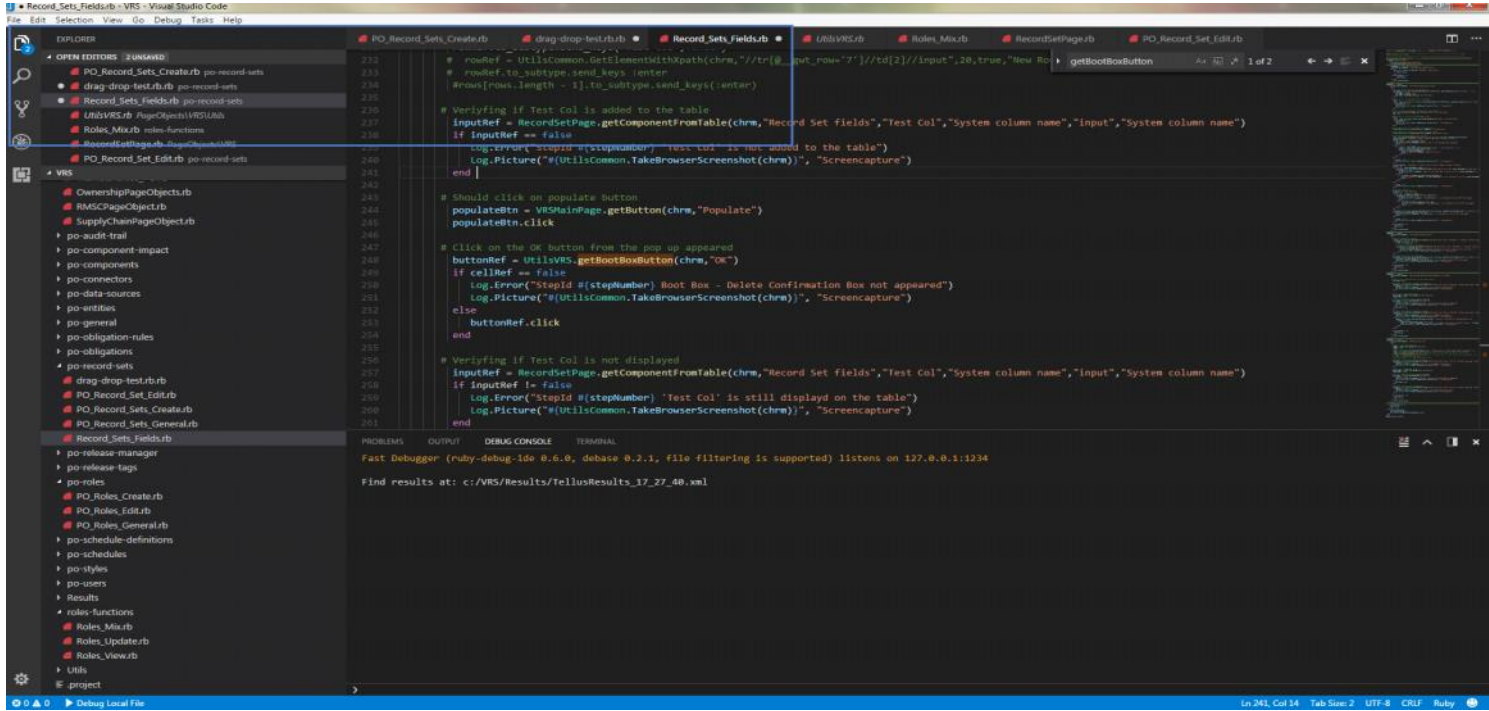
```
begin

rescue => exception

end
```

- Useful Extensions: Note: Once you install any extension you should restart sometimes inorder to initialize the installed extension sometimes.
 - 1) Ruby Solargraph: To get the ruby methods extensions. Ctrl+Space to get the suggestions.

- It will show the Un Saved Files:



If you observe the above screenshot on the top left for the tab File Explorer it is showing as 2 it means we have 2 Unsaved Files and in the Open Editors tab to the right of File Explorer tab we have 2 files called drag-drop-tested.rb, Record_Sets_Fields.rb with dot left to its file name so those are the files that are unsaved

- When we migrate to GIT it will be easy for us to track the Un Staged, Modified filed It will directly show as U - Un staged, M- Modified to the right of the file names.
- It will directly identify the merge conflicts and show as diff of 2 version in the Editor when we want to stage the file.
- To Add Custom Snippets in VSCode. Please navigate to
 - 1) File > Preferences > User Snippets
 - 2) Select your language from the new window/palette that appeared
 - 3) It will create language.json file. Ex: ruby.json or javascript.json
 - 4) Add you Snippet Code. Attached a snippet code



ruby

Which IDE?









05 December 2017 18:02

Meeting Subject: Selenium IDE and documentation

Meeting Date: 12/5/2017 5pm UK Time

Invitation Message

Participants

-  [Ramya Parthasarathy](#) (Meeting Organizer)
-  [Rajul Shrivastava](#)
-  Santhoshi Chepuru (Presenter)
-  [Bhupender Agarwal](#)
-  [Vasu Nakka](#)
-  Deekshith kumar Lingoju
-  Swamy Kommu
-  Shiva Kumar Macharla

Notes from the Meeting:

VS Code Advantages over Aptana

1. Free, extensible, and customizable
2. Common IDE for both Protractor and Selenium
3. JavaScript **IntelliSense** support - JavaScript IntelliSense allows Visual Studio Code to provide you with useful hints and auto-completion features while you code.
4. Embedded **Git** control - Visual Studio Code has integrated Git control, guaranteeing speed, data integrity, and support for distributed, non-linear workflows.
5. Custom snippets support - Snippets are templates that will insert text for you and adapt it to their context, and in VSC they are highly customizable - <https://code.visualstudio.com/docs/editor/userdefinedsnippets>
6. Ruby suggestions are quick
7. Search Files / Strings
8. Huge community behind it - The ease of getting assistance and finding tutorials is increasing as the community grows.

Actions:

1. Ask scripters to start using VSCode: [Getting started with VS code editor for Selenium](#)
2. Shiva and team to explore Rubymine for 2-3 weeks and see if this is any better than VS Code
3. Based on the Pros - We probably can negotiate JetBrains for a package for both WebStorm and RubyMine
4. Somebody to explore RubyLint?
 - a. <https://github.com/YorickPeterse/ruby-lint>
5. Start following yard documentation for all new test cases from now on : [Doc Generator for ruby scripts](#)

Automation Issues

Monday, March 12, 2018 4:22 PM