



Smart Design  
scalable solutions

# Robot Plugin

For IntelliJ IDEA

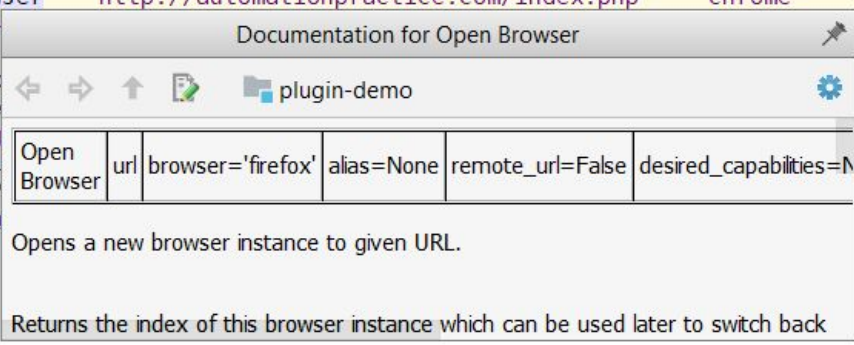
Andrey Chernyshev

[andrey.chernyshev@smddev.com](mailto:andrey.chernyshev@smddev.com)

[www.smddev.com](http://www.smddev.com)

# Syntax highlight, keyword documentation

```
1  *** Settings ***
2  Library           ElementLocatorsLibrary  web_demo.locators
3  Library           Selenium2Library
4
5  *** Variables ***
6  ${PARENT_ELEMENT}
7
8  *** Keywords ***
9  Teardown
10     Capture Page Screenshot
11     Close Browser
12  *** Test Cases ***
13  Locator Test
14     Open Browser    http://automationpractice.com/index.php    chrome
15     ${button}
16     ${input}
17     Input Text
18     Click Ele
19     ${warning}
20     Wait Unti
21     Capture P
22     [Teardown]
23
```



The image shows a Selenium IDE interface with a keyword documentation popup for the 'Open Browser' keyword. The popup window is titled 'Documentation for Open Browser' and contains the following information:

Open Browser	url	browser='firefox'	alias=None	remote_url=False	desired_capabilities=N
--------------	-----	-------------------	------------	------------------	------------------------

Opens a new browser instance to given URL.

Returns the index of this browser instance which can be used later to switch back

# Keyword explorer

The screenshot shows the Keyword Explorer interface. On the left is a file tree for a 'plugin-demo' project. The 'resources' folder is selected. The main area displays a table of keywords with columns for Keyword, Component, and Source. The table lists several keywords from the 'requests.robot' library. Below the table are navigation icons and a 'Keyword documentation' link.

Keyword	Component	Source
Append By Path		requests.robot (resources/rest)
Append By Path From Json String		requests.robot (resources/rest)
CSV Body Processing		requests.robot (resources/rest)
Can Have Body		requests.robot (resources/rest)

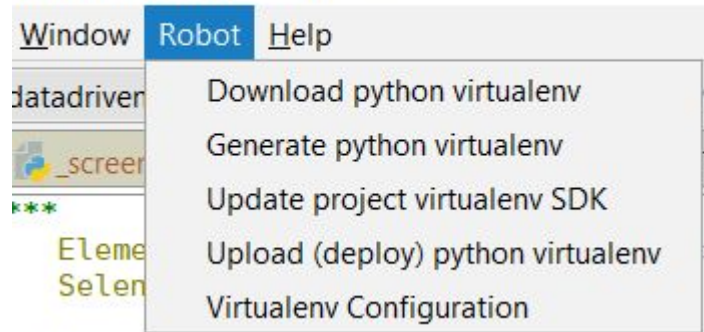
Keyword documentation

# Keyword autocompletion

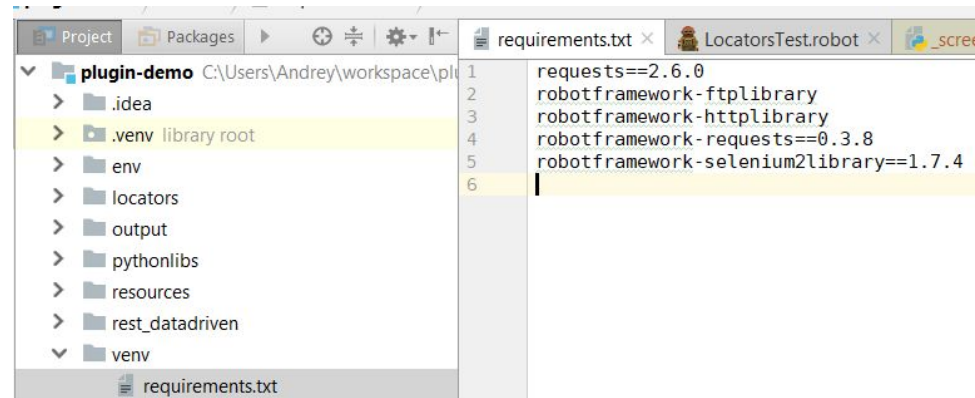
```
*** Test Cases ***  
Locator Test  
Open  
0 .venv.Lib.site-packages.Selenium2Library.Open Browser  
$ .venv.Lib.site-packages.Selenium2Library.Open Co,,  
$  
I v ${PARENT_ELEMENT}  
C v ${TEST_ENV}  
$ Hit Ctrl+Space to show not imported keywords  
Wait Until Page Contains Element    ${warning}  
Capture Page Screenshot  
[Teardown] Teardown
```

# Manage Python virtual environments

plugin-demo] - IntelliJ IDEA



given a requirements.txt:



# Pulling the Robot test dev comfort up to the level of popular langs with:

- Syntax Highlight
- Code Completion
- Jump to Source
- Code Inspections
- Refactorings
- Content suggestions
- Python support
- Keyword explorer tool window
- Run configurations
- Debug Robot test suites
- Robot test runner console
- Python virtual environments management
- Table view/format

# Requirements

- Python 2.7.9
- IntelliJ IDEA 2017.3 (Community edition)
- PyCharm plugin

# Robot debugger

Breakpoint  
In test or  
keyword

The screenshot displays the Robot Framework debugger interface. The top pane shows the test file `rest.robot` with a breakpoint set at line 438. The code at this line is `Send Data Driven Request` with arguments `$(in)`. The code below the breakpoint shows various keywords and their return statuses, such as `Run Keyword If`, `Set Variable`, `Run Keyword And Return Status`, and `Run Keyword If`.

The bottom pane shows the debugger's state. The `Frames` pane displays the current execution context, including the `keyword, requests.Send Data Driven Request $(in)` frame. The `Variables` pane shows the current values of variables, including `$(BASE_PATH)`, `$(CURRENT_SESSION)`, `$(HOST_BA)`, `$(PORT_BA)`, `$(TEST_DOCUMENTATION)`, `$(TEST_NAME)`, `$(Test Files_EXPECTED_RESULT)`, and `$(Test Files_ISSUE_KEY)`.

Keyword  
stack

Variables  
& values



# Questions?

Drop us a note at:

[info@smddev.com](mailto:info@smddev.com)