
Python SPOJ Documentation

Release 0.5.3

Dheerendra Rathor

January 02, 2015

1	Contents:	3
1.1	Installation	3
1.2	Configuration	3
1.3	Language	4
1.4	Submit	6
1.5	Status	6
1.6	About	6
2	Basic Commands	9

This is a command line tool to help you submit your problems easily to [SPOJ](#) using your linux terminal. Using this package is as easy as using any other command line tool.

Contents:

1.1 Installation

Note: Supported Python version is 2.7.x

To install this tool, you can use any of the following commands

```
$ pip install --upgrade python_spoj
$ easy_install python_spoj
```

OR you can download the source code from project [Github Repository](#) and use the following command.

```
$ python setup.py install
```

After installing the package, you need to configure it (Configuration is not necessary but suggested). For configuration refer [Configuration](#) page.

1.2 Configuration

If you want to avoid entering credentials for every submit or want to setup your favourite language, it is advised to properly configure after installation.

To see help about configuration use **spoj config -help**. This will display output similar to following

```
Usage: spoj config [OPTIONS]
```

```
Options:
```

```
-l, --language [1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31|32|33|34|35|36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|62|63|64|65|66|67|68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|96|97|98|99|100]
                                Choose Default Language. See `spoj
                                language`

-c, --credential
--help                        Show this message and exit.
```

1.2.1 Credentials

To store your credentials use **spoj config -c/-credential**. After running this command, few prompt will appear

- **Username:** - Enter your SPOJ username here
- **Password:** - Enter your SPOJ password here

- **Confirm Password:** - Enter your SPOJ password again for confirmation.

After these three prompts, your password will be verified with SPOJ server and after verification your credentials will be remembered and will not be asked again while submitting.

Note: Your credentials are stored on your PC in a configuration file. We **do not** save or store your credentials on any server on internet.

1.2.2 Language

To mark your favourite programming language, use **spoj config -l/-language <language code>**. The language code should be an integer and should be available in the list produced by **spoj language**. For details see command *spoj language*.

1.3 Language

This page display the supported languages for submitting problems.

1.3.1 Languages

<Language Name> : <Language Code>

- ADA 95 (gnat 4.3.2) : 7
- Assembler (nasm 2.03.01) : 13
- Awk (gawk-3.1.6) : 104
- Bash (bash-4.0.37) : 28
- Brainf**k (bff 1.0.3.1) : 12
- C (gcc 4.3.2) : 11
- C# (gmcs 2.0.1) : 27
- C++ (g++ 4.0.0-8) : 1
- C++ (g++ 4.3.2) : 41
- C99 strict (gcc 4.3.2) : 34
- Clips (clips 6.24) : 14
- Clojure (clojure 1.1.0) : 111
- Common Lisp (sbcl 1.0.18) : 31
- Common Lisp (clisp 2.44.1) : 32
- D (gdc 4.1.3) : 20
- Erlang (erl 5.6.3) : 36
- F# (fsharp 2.0.0) : 124
- Fortran 95 (gfortran 4.3.2) : 5
- Go (gc 2010-07-14) : 114

- Haskell (ghc 6.10.4) : 21
- Icon (iconc 9.4.3) : 16
- Intercal (ick 0.28-4) : 9
- JAR (JavaSE 6) : 24
- Java (JavaSE 6) : 10
- JavaScript (rhino 1.7R1-2) : 35
- Lua (luac 5.1.3) : 26
- Nemerle (ncc 0.9.3) : 30
- Nice (nicec 0.9.6) : 25
- Node.js (0.8.11) : 56
- Ocaml (ocamlopt 3.10.2) : 8
- Pascal (fpc 2.2.4) : 22
- Pascal (gpc 20070904) : 2
- Perl (perl 5.12.1) : 3
- Perl 6 (rakudo-2010.08) : 54
- PHP (php 5.2.6) : 29
- Pike (pike 7.6.112) : 19
- Prolog (swipl 5.6.58) : 15
- Python (python 2.7) : 4
- Python (PyPy 2.4) : 99
- Python 3 (python 3.2.3) : 116
- Python 3 nbc (python 3.2.3 nbc) : 126
- Python3.4 (Python 3.4) : 98
- Ruby (ruby 1.9.3) : 17
- Scala (scala 2.8.0) : 39
- Scheme (stalin 0.11) : 18
- Scheme (guile 1.8.5) : 33
- Sed (sed-4.2) : 46
- Smalltalk (gst 3.0.3) : 23
- Tcl (tclsh 8.5.3) : 38
- TECS () : 42
- Text (plain text) : 62
- Whitespace (wspace 0.3) : 6

For submitting problem use **language code** to specify a language.

1.4 Submit

This command is useful for submitting problem to spoj. By running **spoj submit --help** you will see the similar output:

Usage: spoj submit [OPTIONS] FILENAME

Options:

-p, --problem TEXT	Problem code
-l, --language [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100]	Language of problem. See 'spoj language'
--help	Show this message and exit.

1.4.1 Required Arguments

- **FILENAME** Specify the path of the file you want to submit. Make sure this file is readable and exists. If this file doesn't exist then you will get similar error message

```
Error: Invalid value for "filename": Could not open file: <FILENAME>: No such file or directory
```

1.4.2 Optional Arguments

- **-p, --problem TEXT** Specify the problem code for you want to submit the solution. If this argument is not present then the filename will be considered as problem code.

Example:

```
filename: ~/username/spoj/myfile.py
problem name used: MYFILE
```

```
filename: /path/to/someFile
problem name used: SOMEFILE
```

- **-l, --language** Specify the language code for the problem you are submitting. If this field is not present then the language is taken from configuration file if already configured. If the language is not configured then a prompt will ask for language code.

See *spoj language* for supported language and codes

1.5 Status

This command do nothing in current version and is reserved for future versions. In future versions this command will show logs of file submitted and their status.

1.6 About

This project is a self motivated project and targeted for programmers who are active on [SPOJ](#).

This project is built over following packages and modules:

- **click** Click is a Python package for creating beautiful command line interfaces in a composable way with as little code as necessary. It's the "Command Line Interface Creation Kit". It's highly configurable but comes with sensible defaults out of the box.

- **mechanize** Stateful programmatic web browsing, after Andy Lester's Perl module WWW::Mechanize.
- **beautifulsoup4** Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

If you want to contribute to this project or want to report bugs/issues, please create an issue on [Github](#)

For hugs and bugs, write to author at dheeru.rathor14@gmail.com

Basic Commands

- **spoj -help** Display the help content for the main command
- **spoj -version** Display the version string of the project
- **spoj <subcommand> -help** Display help content for subcommand. Available subcommands are:
 - *config*
 - *language*
 - *submit*
 - *status*