
dockerjudge

Jul 01, 2021

CONTENTS:

1	Badges	1
2	Project links	3
2.1	Installation	3
2.2	<code>dockjudge.main</code> - Main	4
2.3	<code>dockjudge.processor</code> - Processors	6
2.4	<code>dockjudge.status</code> - Statuses	8
3	Indices and tables	9
	Python Module Index	11
	Index	13

BADGES

Service	Status
AppVeyor	
Azure Pipelines	
Code Climate	
FOSSA	
GitHub Actions	
GitLab CI/CD	
Read the Docs	
Travis CI	

PROJECT LINKS

- [GitHub Repository](#)
- [GitLab Repository](#)
- [PyPI Project](#)
- [Transifex Translation](#)

2.1 Installation

2.1.1 Docker

To run `dockerjudge`, [Docker Engine](#) is required.

Install using the convenience script (for servers)

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

See [Install Docker Engine | Docker Documentation](#) for more information.

2.1.2 Package `dockerjudge`

From the Python Package Index (PyPI)

`dockerjudge` · [PyPI](#)

- [PyPI](#)
- [Alibaba Open Source Mirror](#)
- [Tsinghua Open Source Mirror](#)

Via pip

```
pip install dockerjudge
```

Via Easy install (deprecated)

```
easy_install dockerjudge
```

From source on GitHub

- HTTPS: <https://github.com/wxh06/dockerjudge.git>
- SSH: <git@github.com:wxh06/dockerjudge.git>

```
git clone https://github.com/wxh06/dockerjudge.git
cd dockerjudge

make pip && make # python3 -m pip install -Ur requirements.txt && python3 setup.py build
sudo make install # python3 setup.py install
```

2.2 dockerjudge.main - Main

dockerjudge main functions

2.2.1 Judge

`dockerjudge.main.judge(processor, source, tests, config=None, client=None)`

Main function

Parameters

- **processor** (*dockerjudge.processor.Processor*, list or tuple) – Programming language processor
- **source** (*str*) – Source code
- **tests** (*list*) – Test cases
- **config** (*dict*) – Configuration

Key		Description	De- fault	Value type
callback	compile	Compilation callback	None	<i>function</i>
	judge	Callback after judging		
demux	compile	Return <i>stdout</i> and <i>stderr</i> of compiler separately	False	<i>bool</i>
iofilename	in	Input filename	<i>stdin</i>	<i>str</i>
	out	Output filename	<i>stdout</i>	
limit	time	Time limit	1	<i>int</i> or <i>float</i>
network		Network enabled	False	<i>bool</i>
threads		Thread limit	None	<i>int</i>

- **client** (*docker.client.DockerClient*) – Docker client

Returns Result

Return type

list

Key	Value type	Description
<i>0</i>	<i>list</i>	Result of each test case
<i>1</i>	<i>byte</i>	Compiler output

Test case

Key	Value type	Description
<i>0</i>	<i>Status</i>	Status code
<i>1</i>	<i>tuple</i>	<i>stdout</i> and <i>stderr</i>
<i>2</i>	<i>float</i>	Time spent

2.2.2 Callback

Compile

Parameter	Type	Description
<i>0</i>	<i>int</i>	Return value of the compiler
<i>1</i>	<i>byte</i> or <i>tuple</i>	Output of compiler, value type depends on <i>demux</i>

Judge

Parameter	Type	Description
<i>0</i>	<i>int</i>	Test case id, starting from <i>0</i>
<i>1</i>	<i>Status</i>	Status
<i>2</i>	<i>tuple</i>	Output (<i>stdout</i> , <i>stderr</i>)
<i>3</i>	<i>float</i>	Time used

2.3 dockerjudge.processor - Processors

2.3.1 Available built-in processors

Processors

Processor	Language(s) ⁰	Required Docker image
<i>*Bash</i>	Shell	<i>bash</i>
<i>Clang</i>	<ul style="list-style-type: none">• C (c)• C++ (cpp)	<i>clangbuiltlinux/ubuntu</i>
<i>GCC</i>	<ul style="list-style-type: none">• C (c)• C++ (cpp)• Go (go)	<i>gcc</i>
<i>Go</i>	Go	<i>golang</i>
<i>Mono</i>	<ul style="list-style-type: none">• Visual Basic (vb)• C# (csharp)	<i>mono</i>
<i>Node</i>	Node.js	<i>node</i>
<i>OpenJDK</i>	Java	<i>openjdk</i>
<i>PHP</i>	PHP	<i>php</i>
<i>PyPy</i>	Python	<i>pypy</i>
<i>Python</i>	Python	<i>python</i>
<i>Ruby</i>	Ruby	<i>ruby</i>
<i>Swift</i>	Swift	<i>swift</i>

class `dockerjudge.processor.Bash`(*version=None*)

Bash is the GNU Project's Bourne Again SHell

Parameters **version** (*str*, *int* or *float*) – Tag name of Docker image *bash*

class `dockerjudge.processor.Clang`(*language=None*, *version=None*, *filenames=None*, *options=None*)

Clang C Language Family Frontend for LLVM

Parameters

- **language** (*dockerjudge.processor.Clang.Language* or *str*) – Programming language (C/c or C++/cpp), C++ by default
- **version** (*str*, *int* or *float*) – Tag name of Docker image *clangbuiltlinux/ubuntu*
- **filenames** (*dict*) – Filenames of source code and binary file, C++ default: {'src': 'a.cpp', 'bin': None}
- **options** (*list* or *str*) – Compiler options

class `dockerjudge.processor.Clang.Language`(*value*)

Programming language, C (c) or C++ (cpp)

C `Clang.Language.c`, `Clang.Language['c']` or `Clang.Language('C')`

C++ `Clang.Language.cpp`, `Clang.Language['cpp']` or `Clang.Language('C++')`

⁰ Emboldened language by default.

class `dockerjudge.processor.GCC`(*language=None, version=None, filenames=None, options=None*)
GNU project C, C++ and Go compiler

Parameters

- **language** (*dockerjudge.processor.GCC.Language* or *str*) – Programming language (C/c, C++/cpp or Go/go), C++ by default
- **version** (*str, int* or *float*) – Tag name of Docker image *gcc*
- **filenames** (*dict*) – Filenames of source code and binary file, C++ default: {'src': 'a.cpp', 'bin': None}
- **options** (*list* or *str*) – Compiler options

class `dockerjudge.processor.GCC.Language`(*value*)
Programming language, C (c), C++ (cpp) or Go (go)

C `GCC.Language.c`, `GCC.Language['c']` or `GCC.Language('C')`

C++ `GCC.Language.cpp`, `GCC.Language['cpp']` or `GCC.Language('C++')`

Go `GCC.Language.go`, `GCC.Language['go']` or `GCC.Language('Go')`

class `dockerjudge.processor.Go`(*version=None, filenames=None, options=None*)
The Go Programming Language

Parameters

- **version** (*str, int* or *float*) – Tag name of Docker image *golang*
- **filenames** (*dict*) – Filenames of source code and binary file, default: {'src': 'main.go', 'bin': None}

class `dockerjudge.processor.Mono`(*language=None, version=None*)
Mono is a software platform designed to allow developers to easily create cross platform applications part of the .NET Foundation.

Sponsored by [Microsoft](#), Mono is an open source implementation of Microsoft's .NET Framework based on the [ECMA](#) standards for [C#](#) and the [Common Language Runtime](#).

Parameters

- **language** (*dockerjudge.processor.Mono.Language* or *str*) – Programming language (Visual Basic/vb or C#/csharp), C# by default
- **version** (*str, int* or *float*) – Tag name of Docker image *mono*

class `dockerjudge.processor.Mono.Language`(*value*)
Programming language, *Visual Basic* (vb) or *C#* (csharp)

Visual Basic `GCC.Language.vb`, `GCC.Language['vb']` or `GCC.Language('Visual Basic')`

C# `GCC.Language.csharp`, `GCC.Language['csharp']` or `GCC.Language('C#')`

class `dockerjudge.processor.Node`(*version=None*)
Node.js®

class `dockerjudge.processor.OpenJDK`(*version=None*)
Open Java Development Kit

class `dockerjudge.processor.PHP`(*version=None*)

class `dockerjudge.processor.PyPy`(*version=None*)

```
class dockerjudge.processor.Python(version=None)
    CPython
```

```
class dockerjudge.processor.Ruby(version=None)
```

```
class dockerjudge.processor.Swift(version=None)
```

2.3.2 Customize

```
class dockerjudge.processor.Processor
```

Defines the operations of a multi-version programming language processor

Data	Type
image	<i>str</i>
workdir	PurePosixPath or <i>str</i>
source	<i>str</i>
before_compile	<i>str</i> or <i>list</i>
compile	<i>str</i> or <i>list</i>
after_compile	<i>str</i> or <i>list</i>
before_judge	<i>str</i> or <i>list</i>
judge	<i>str</i>
after_judge	<i>str</i> or <i>list</i>

2.4 dockerjudge.status - Statuses

The collection of judge statuses

```
class dockerjudge.status.Status(value)
```

Enumeration of judge statuses

Name	Value
AC	<i>Accepted</i>
WA	<i>Wrong Answer</i>
ONF	<i>Output Not Found</i>
RE	<i>Runtime Error</i>
TLE	<i>Time Limit Exceeded</i>
UE	<i>Unknown Error</i>
CE	<i>Compilation Error</i>

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

d

`dockerjudge.main`, 4
`dockerjudge.processor`, 6
`dockerjudge.status`, 8

INDEX

B

Bash (*class in dockerjudge.processor*), 6

C

Clang (*class in dockerjudge.processor*), 6

D

dockerjudge.main

module, 4

dockerjudge.processor

module, 6

dockerjudge.status

module, 8

G

GCC (*class in dockerjudge.processor*), 6

Go (*class in dockerjudge.processor*), 7

J

judge() (*in module dockerjudge.main*), 4

L

Language (*class in dockerjudge.processor.Clang*), 6

Language (*class in dockerjudge.processor.GCC*), 7

Language (*class in dockerjudge.processor.Mono*), 7

M

module

dockerjudge.main, 4

dockerjudge.processor, 6

dockerjudge.status, 8

Mono (*class in dockerjudge.processor*), 7

N

Node (*class in dockerjudge.processor*), 7

O

OpenJDK (*class in dockerjudge.processor*), 7

P

PHP (*class in dockerjudge.processor*), 7

Processor (*class in dockerjudge.processor*), 8

PyPy (*class in dockerjudge.processor*), 7

Python (*class in dockerjudge.processor*), 7

R

Ruby (*class in dockerjudge.processor*), 8

S

Status (*class in dockerjudge.status*), 8

Swift (*class in dockerjudge.processor*), 8