ColabHost Documentation

Release 0.1.9

Puneetha Pai

Sep 19, 2020

Contents:

CHAPTER 1

ColabHost

1.1 Host any (python) application in colab or kaggle notebook environment

Google Colab and Kaggle notebook environments are great. You have powerful compute, but just using their notebook environment feels restrictive. Given the hardware you should be able to do more.

Inspired from Abhishek Thakur, and his work on colab code this package extends the idea.

CHAPTER 2

Features

2.1 Supported IDEs:

- Jupyter Notebook
- Jupyter Lab
- For VScode you can use colabcode

2.2 Supported Applications:

- Flask and Gunicron applications
- FastAPI and Uvicorn applications

This is purely for developement and testing purpose. You can use supported IDEs to seamlessly develop your idea and also host them for testing purpose. It is not advised to use for production purpose.

2.3 Installation

```
$ pip install colab_host
```

- Free software: MIT license
- Documentation: https://colab-host.readthedocs.io.

2.4 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

2.4.1 ColabHost

Host any (python) application in colab or kaggle notebook environment

Google Colab and Kaggle notebook environments are great. You have powerful compute, but just using their notebook environment feels restrictive. Given the hardware you should be able to do more.

Inspired from Abhishek Thakur, and his work on colab code this package extends the idea.

2.4.2 Features

Supported IDEs:

- Jupyter Notebook
- Jupyter Lab
- For VScode you can use colabcode

Supported Applications:

- Flask and Gunicron applications
- FastAPI and Uvicorn applications

This is purely for developement and testing purpose. You can use supported IDEs to seamlessly develop your idea and also host them for testing purpose. It is not advised to use for production purpose.

Installation

```
$ pip install colab_host
```

- Free software: MIT license
- Documentation: https://colab-host.readthedocs.io.

Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

2.4.3 Installation

Stable release

To install ColabHost, run this command in your terminal:

```
$ pip install colab_host
```

This is the preferred method to install ColabHost, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

From sources

The sources for ColabHost can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/PuneethaPai/colab_host
```

Or download the tarball:

```
$ curl -OJL https://github.com/PuneethaPai/colab_host/tarball/master
```

Once you have a copy of the source, you can install it with:

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

2.4.4 Usage

To start a simple HTTP File Server:

```
from colab_host import SimpleHttpServer

SimpleHttpServer(port=1000)
```

To start Jupyter Notebook in colab:

```
from colab_host import JupyterNotebook

JupyterNotebook(port=1000)
```

To start Jupyter Lab in colab:

```
from colab_host import JupyterLab

JupyterLab(port=1000)
```

To start Flask Application in colab:

```
from colab_host import FlaskApp

FlaskApp(
    port=1000,
    app="main:app",
    git_url="https://github.com/PuneethaPai/colab_host_flask_demo.git",
    requirements_file="requirements.txt"
)
```

To start Uvicorn App in colab:

```
from colab_host import UvicornApp

UvicornApp(
    port=1000,
    app="main:app",
    git_url="https://github.com/PuneethaPai/colab_host_uvicorn_demo.git",
    requirements_file="requirements.txt"
)
```

2.4.5 colab_host

colab_host package

Submodules

colab_host.colab_host module

class `colab_host.colab_host.`**`FlaskApp`**(*port: int = 1000*, *app='main:app'*, *git_url='https://github.com/PuneethaPai/colab_host_flask_demo'*, *requirements_file: str = 'requirements.txt'*)

Bases: *`colab_host.colab_host.Host`*

Class to expose python Flask or Gunicorn application.

Parameters:

port : int, optional

app [str, optional] Definition of your python gunicorn app. (Defaults to *"main:app"*).

git_url [str, optional] Git URL to clone your repo containing application. (Defaults to *"https://github.com/PuneethaPai/colab_host_flask_demo"*).

requirements_file: str, optional Name of file in repo *git_url* containing requirements for hosting the application. (Defaults to *"requirements.txt"*).

class `colab_host.colab_host.`**`Host`**(*port: int = 1000*, *requirements: list = None*, *git_url: str = None*)

Bases: `object`

Base class for hosting any python application.

Given *port* number it will expose the port to internet. Given *requirements* will install them using *pip install*. Given *git_url* it will clone the repo for you.

Parameters:

port : int, optional

requirements [List[str] or str, optional] List[str]: list of package requirements for hosting. str: requirements file path to install requirements from.

git_url : str, optional

class `colab_host.colab_host.`**`JupyterLab`**(*port: int = 1000*, *requirements: list = ['jupyterlab']*)

Bases: *`colab_host.colab_host.Host`*

Class to expose Jupyter Lab IDE on browser.

Parameters:

port : int, optional

requirements [List[str], optional] Defaults to *["jupyterlab"]* and you can include other packages to include with this. For example notebook extension, theme, etc

class colab_host.colab_host.**JupyterNotebook**(*port: int = 1000, requirements: list = ['notebook']*)

Bases: *colab_host.colab_host.Host*

Class to expose Jupyter Notebook IDE on browser.

Parameters:

port : int, optional

requirements [List[str], optional] Defaults to *["notebook"]* and you can include other packages to include with this. For example notebook extension, theme, etc

class colab_host.colab_host.**SimpleHttpServer**(*port: int = 1000*)

Bases: *colab_host.colab_host.Host*

Class to expose simple file server application.

Parameters:

port : int, optional

class colab_host.colab_host.**UvicornApp**(*port: int = 1000, app='main:app', git_url='https://github.com/PuneethaPai/colab_host_flask_demo', requirements_file: str = 'requirements.txt'*)

Bases: *colab_host.colab_host.Host*

Class to expose python FastApi or Uvicorn application.

Parameters:

port : int, optional

app [str, optional] Definition of your python gunicorn app. (Defaults to *"main:app"*).

git_url [str, optional] Git URL to clone your repo containing application. (Defaults to *"https://github.com/PuneethaPai/colab_host_uvicorn_demo"*).

requirements_file: str, optional Name of file in repo *git_url* containing requirements for hosting the application. (Defaults to *"requirements.txt"*).

Module contents

Top-level package for ColabHost.

2.4.6 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/PuneethaPai/colab_host/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

Write Documentation

ColabHost could always use more documentation, whether as part of the official ColabHost docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/PuneethaPai/colab_host/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here’s how to set up *colab_host* for local development.

1. Fork the *colab_host* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/colab_host.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv colab_host
   $ cd colab_host/
   $ python setup.py develop
   ```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 colab_host tests
$ python setup.py test or pytest
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 3.5, 3.6, 3.7 and 3.8, and for PyPy. Check https://travis-ci.com/PuneethaPai/colab_host/pull_requests and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ python -m unittest tests.test_colab_host
```

Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bump2version patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

2.4.7 Credits

Development Lead

- Puneetha Pai <puneethapai29@gmail.com>

Contributors

None yet. Why not be the first?

2.4.8 History

0.1.0 (2020-09-10)

- First release on PyPI.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index