

NAME

jupyter-run – Helper to run jupyter apps from the command line

DESCRIPTION

Run Jupyter kernel code.

OPTIONS

Arguments that take values are actually convenience aliases to full Configurables, whose aliases are listed on the help line. For more information on full configurables, see `'--help-all'`.

--generate-config

generate default config file

--debug

set log level to logging.DEBUG (maximize logging output)

--no-confirm-exit

Don't prompt the user when exiting. This will terminate the kernel if it is owned by the frontend, and leave it alive if it is external. This can also be set in the config file by setting `'c.JupyterConsoleApp.confirm_exit'`.

--confirm-exit

Set to display confirmation dialog on exit. You can always use `'exit'` or `'quit'`, to force a direct exit without any confirmation. This can also be set in the config file by setting `'c.JupyterConsoleApp.confirm_exit'`.

-y

Answer yes to any questions instead of prompting.

--existing

Connect to an existing kernel. If no argument specified, guess most recent

--transport=<CaselessStrEnum> (JupyterConsoleApp.transport)

Default: `'tcp'` Choices: [`'tcp'`, `'ipc'`]

--hb=<Int> (JupyterConsoleApp.hb_port)

Default: 0 set the heartbeat port [default: random]

--ssh=<Unicode> (JupyterConsoleApp.sshserver)

Default: `''` The SSH server to use to connect to the kernel.

--ip=<Unicode> (JupyterConsoleApp.ip)

Default: `''` Set the kernel's IP address [default localhost]. If the IP address is something other than localhost, then Consoles on other machines will be able to connect to the Kernel, so be careful!

--kernel=<Unicode> (JupyterConsoleApp.kernel_name)

Default: `'python'` The name of the default kernel to start.

--log-level=<Enum> (Application.log_level)

Default: 30 Choices: (0, 10, 20, 30, 40, 50, `'DEBUG'`, `'INFO'`, `'WARN'`, `'ERROR'`, `'CRITICAL'`)
Set the log level by value or name.

--stdin=<Int> (JupyterConsoleApp.stdin_port)

Default: 0 set the stdin (ROUTER) port [default: random]

--config=<Unicode> (JupyterApp.config_file)

Default: `''` Full path of a config file.

--iopub=<Int> (JupyterConsoleApp.iopub_port)

Default: 0 set the iopub (PUB) port [default: random]

--shell=<Int> (JupyterConsoleApp.shell_port)

Default: 0 set the shell (ROUTER) port [default: random]

-f <Unicode> (JupyterConsoleApp.connection_file)

Default: " JSON file in which to store connection info [default: kernel-<pid>.json] This file will contain the IP, ports, and authentication key needed to connect clients to this kernel. By default, this file will be created in the security dir of the current profile, but can be specified by absolute path.

--existing=<CUnicode> (JupyterConsoleApp.existing)

Default: " Connect to an already running kernel

CLASS PARAMETERS

Parameters are set from command-line arguments of the form: '**--Class.trait=value**'. This line is evaluated in Python, so simple expressions are allowed, e.g.: '**--C.a=range(3)**'. For setting C.a=[0,1,2].

RUNAPP OPTIONS

--RunApp.answer_yes=<Bool>

Default: False Answer yes to any prompts.

--RunApp.config_file=<Unicode>

Default: " Full path of a config file.

--RunApp.config_file_name=<Unicode>

Default: " Specify a config file to load.

--RunApp.confirm_exit=<CBool>

Default: True Set to display confirmation dialog on exit. You can always use 'exit' or 'quit', to force a direct exit without any confirmation.

--RunApp.connection_file=<Unicode>

Default: " JSON file in which to store connection info [default: kernel-<pid>.json] This file will contain the IP, ports, and authentication key needed to connect clients to this kernel. By default, this file will be created in the security dir of the current profile, but can be specified by absolute path.

--RunApp.control_port=<Int>

Default: 0 set the control (ROUTER) port [default: random]

--RunApp.existing=<CUnicode>

Default: " Connect to an already running kernel

--RunApp.generate_config=<Bool>

Default: False Generate default config file.

--RunApp.hb_port=<Int>

Default: 0 set the heartbeat port [default: random]

--RunApp.iopub_port=<Int>

Default: 0 set the iopub (PUB) port [default: random]

--RunApp.ip=<Unicode>

Default: " Set the kernel's IP address [default localhost]. If the IP address is something other than localhost, then Consoles on other machines will be able to connect to the Kernel, so be careful!

--RunApp.kernel_name=<Unicode>

Default: 'python' The name of the default kernel to start.

--RunApp.kernel_timeout=<Float>

Default: 60 Timeout for giving up on a kernel (in seconds). On first connect and restart, the console tests whether the kernel is running and responsive by sending kernel_info_requests. This sets the timeout in seconds for how long the kernel can take before being presumed dead.

--RunApp.log_datefmt=<Unicode>

Default: '%Y-%m-%d %H:%M:%S' The date format used by logging formatters for %(asctime)s

--RunApp.log_format=<Unicode>

Default: '[%(name)s]%(highlevel)s %(message)s' The Logging format template

--RunApp.log_level=<Enum>

Default: 30 Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL')
Set the log level by value or name.

--RunApp.shell_port=<Int>

Default: 0 set the shell (ROUTER) port [default: random]

--RunApp.sshkey=<Unicode>

Default: '' Path to the ssh key to use for logging in to the ssh server.

--RunApp.sshserver=<Unicode>

Default: '' The SSH server to use to connect to the kernel.

--RunApp.stdin_port=<Int>

Default: 0 set the stdin (ROUTER) port [default: random]

--RunApp.transport=<CaselessStrEnum>

Default: 'tcp' Choices: ['tcp', 'ipc']