Configure Flask dev server to be visible across the network

Asked 13 years, 4 months ago Modified 4 months ago

Viewed 871k times

723

I'm not sure if this is Flask specific, but when I run an app in dev mode (`http://localhost:5000`), I cannot access it from other machines on the network (with `http://[dev-host-ip]:5000`). With Rails in dev mode, for example, it works fine. I couldn't find any docs regarding the Flask dev server configuration. Any idea what should be configured to enable this?

python flask werkzeug

Share Follow

edited Oct 11, 2017 at 0:19

davidism

127k ● 30 ● 414 ● 347

asked Aug 11, 2011 at 8:40

sa125

28.9k ● 38 ● 113 ● 157

19 Answers

Sorted by: Highest score (default)

1071

While this is possible, you should not use the Flask dev server in production. The Flask dev server is not designed to be particularly secure, stable, or efficient. See the docs on deploying for correct solutions.

The `--host` option to `flask run`, or the `host` parameter to `app.run()`, controls what address the development server listens to. By default it runs on `localhost`, change it to `flask run --host=0.0.0.0` (or `app.run(host="0.0.0.0")`) to run on all your machine's IP addresses.

`0.0.0.0` is a special value that you can't use in the browser directly, you'll need to navigate to the actual IP address of the machine on the network. You may also need to adjust your firewall to allow external access to the port.

The Flask quickstart docs explain this in the "Externally Visible Server" section:

> If you run the server you will notice that the server is only accessible from your own computer, not from any other in the network. This is the default because in debugging mode a user of the application can execute arbitrary Python code on your computer.
>
> If you have the debugger disabled or trust the users on your network, you can make the server

> publicly available simply by adding `--host=0.0.0.0` to the command line:
>
> ```
> $ flask run --host=0.0.0.0
> ```
>
> This tells your operating system to listen on all public IPs.

Share Follow

edited Jun 11, 2021 at 9:19

HackDolphin
176 ● 11

answered Aug 11, 2011 at 13:53

Shawn Swaner
11.3k ● 1 ● 22 ● 14

2 For me it was not working before after disabling firewall, it worked for me. – Kunal Soni Jan 18, 2023 at 9:06

1 @Kunal-Soni Somehow disabling an entire firewall is not a good idea. Creating an Exception that lets a specific app communicate with your computer however is completely different and is a far better solution. If the latter is what you mean by "disabling firewall", then that is good. – Stev Jul 18, 2023 at 8:08

This worked great for me! I did have to disconnect from my VPN, so an FYI to anyone who has a VPN. – Tyler Petrov Feb 12 at 15:44

If you use the `flask` executable to start your server, use `flask run --host=0.0.0.0` to change the default from

178 127.0.0.1 and open it up to non-local connections.

> If you run the server you will notice that the server is only accessible from your own computer, not from any other in the network. This is the default because in debugging mode a user of the application can execute arbitrary Python code on your computer.
>
> If you have the debugger disabled or trust the users on your network, you can make the server publicly available simply by adding `--host=0.0.0.0` to the command line:
>
> ```
> $ flask run --host=0.0.0.0
> ```
>
> This tells your operating system to listen on all public IPs.

Reference: https://flask.palletsprojects.com/quickstart/

Share Follow

edited Mar 29, 2021 at 15:18

davidism

127k ● 30 ● 414 ● 347

answered Oct 29, 2016 at 1:28

Aria

1,937 ● 1 ● 12 ● 8

Try this if the 0.0.0.0 method doesn't work

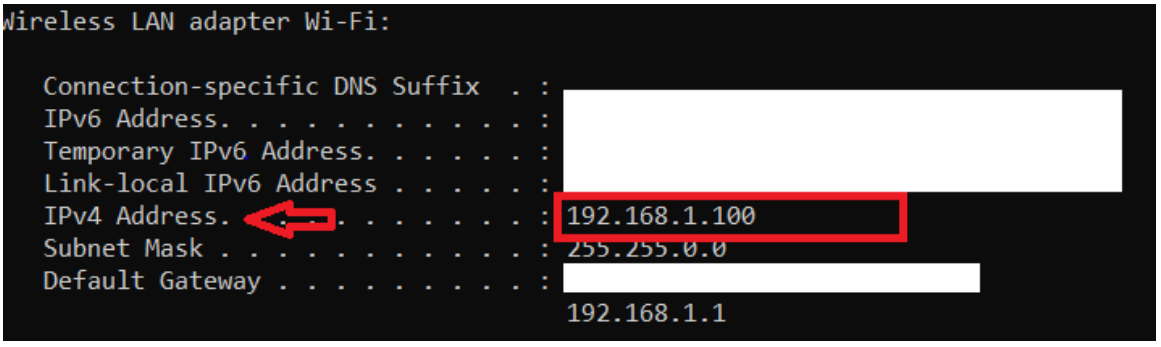
99

Boring Stuff

I personally battled a lot to get my app accessible to other devices(laptops and mobile phones) through a local-server. I tried the 0.0.0.0 method, but no luck. Then I tried changing the port, but it just didn't work. So, after trying a bunch of different combinations, I arrived to this one, and it solved my problem of deploying my app on a local server.

Steps

1. Get the local IPv4 address of your computer. This can be done by typing `ipconfig` on Windows and `ifconfig` on Linux and Mac.



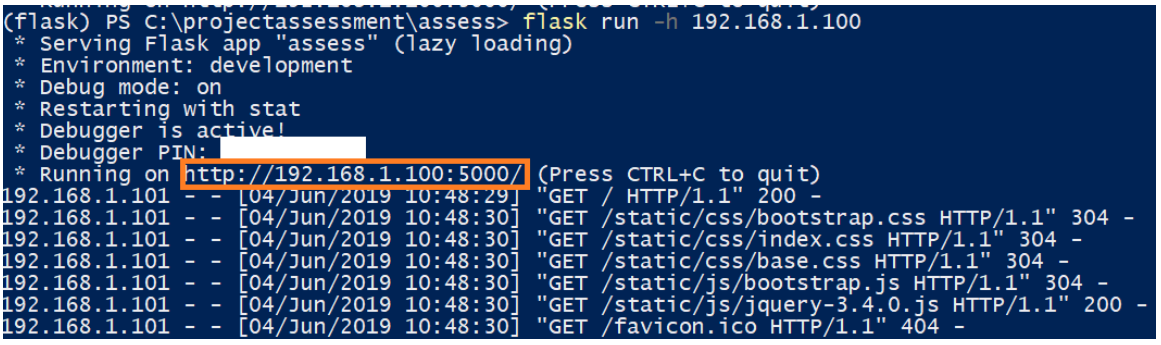
Please note: The above step is to be performed on the machine you are serving the app on, and on not the machine on which you are accessing it. Also note, that the IPv4 address might change if you disconnect and reconnect to the network.

2. Now, simply run the flask app with the acquired IPv4 address.

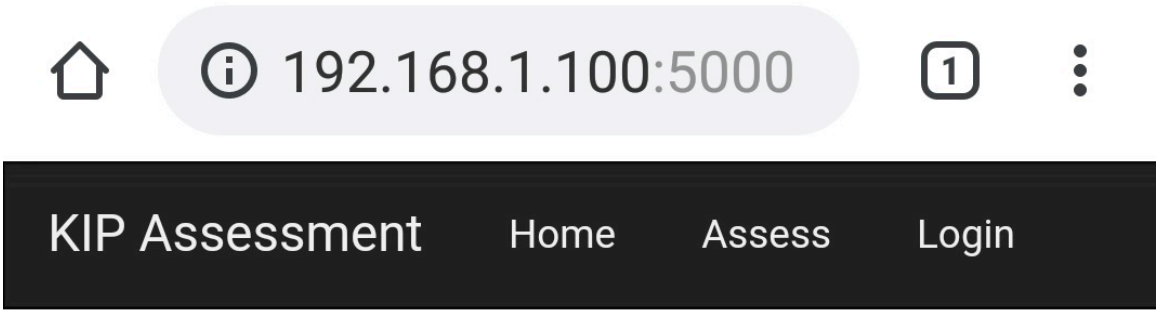
   `flask run -h 192.168.X.X`

   E.g. In my case (see the image), I ran it as:

```
flask run -h 192.168.1.100
```



On my mobile device



Optional Stuff

If you are performing this procedure on Windows and using Power Shell as the CLI, and you still aren't able to access the website, try a CTRL + C command in the shell that's running the app. Power Shell gets frozen up sometimes and it needs a pinch to revive. Doing this

might even terminate the server, but it sometimes does the trick.

That's it. Give a thumbs up if you found this helpful.😉

Some more optional stuff

I have created a short Powershell script that will get you your IP address whenever you need one:

```
$env:getIp = ipconfig
if ($env:getIp -match '(IPv4[\sa-zA-Z.]+:\s[0-9.]+)')
    if ($matches[1] -match '([^a-z\s][\d]+[.\d]+)'){
        $ipv4 = $matches[1]
    }
}
echo $ipv4
```

Save it to a file with .ps1 extension (for PowerShell), and run it on before starting your app. You can save it in your project folder and run it as:

```
.\getIP.ps1; flask run -h $ipv4
```

Note: I saved the above shellcode in getIP.ps1.

Cool.👌

Share Follow

edited Aug 11, 2020 at 7:28

Abhishek kumar

139 • 8

answered Jun 4, 2019 at 6:52

nwillo
1,284 • 9 • 11

1 On macOS or Linux you can get the same ip by entering `ipconfig getifaddr en0` into the command line and use that IP as your `--host` value. – W1ck3d Jan 27, 2022 at 6:41

5 Heewoon this IP address is local to his network. Chances are you have it too. – Westsi May 2, 2022 at 14:19

86

Add `host='0.0.0.0' to` app.run`.

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

If you get `OSError: [WinError 10013] An attempt was made to access a socket in a way forbidden by its access permissions` on Windows, you either don't have permission to use the port, or something else is using it which you can find with `netstat -na|findstr 5000`.

Share Follow

edited Mar 29, 2021 at 15:21

davidism
127k • 30 • 414 • 347

answered Jan 4, 2018 at 9:18

Jeevan Chaitanya
1,374 • 11 • 12

27

Check whether the particular port is open on the server to serve the client or not?

in Ubuntu or Linux distro

```
sudo ufw enable
sudo ufw allow 5000/tcp //allow the server to handle t
```

Configure the application to handle remote requests

```
app.run(host='0.0.0.0' , port=5000)


python3 app.py & #run application in background
```

Share Follow

answered Sep 1, 2018 at 19:12

Rajesh Prasad Yadav

908 ● 11 ● 9

25

If your `cool` app has it's configuration loaded from an external file, like in the following example, then don't forget to update the corresponding config file with HOST="0.0.0.0"

```
cool.app.run(
    host=cool.app.config.get("HOST", "localhost"),
    port=cool.app.config.get("PORT", 9000)
)
```

Share Follow

answered May 7, 2015 at 20:52

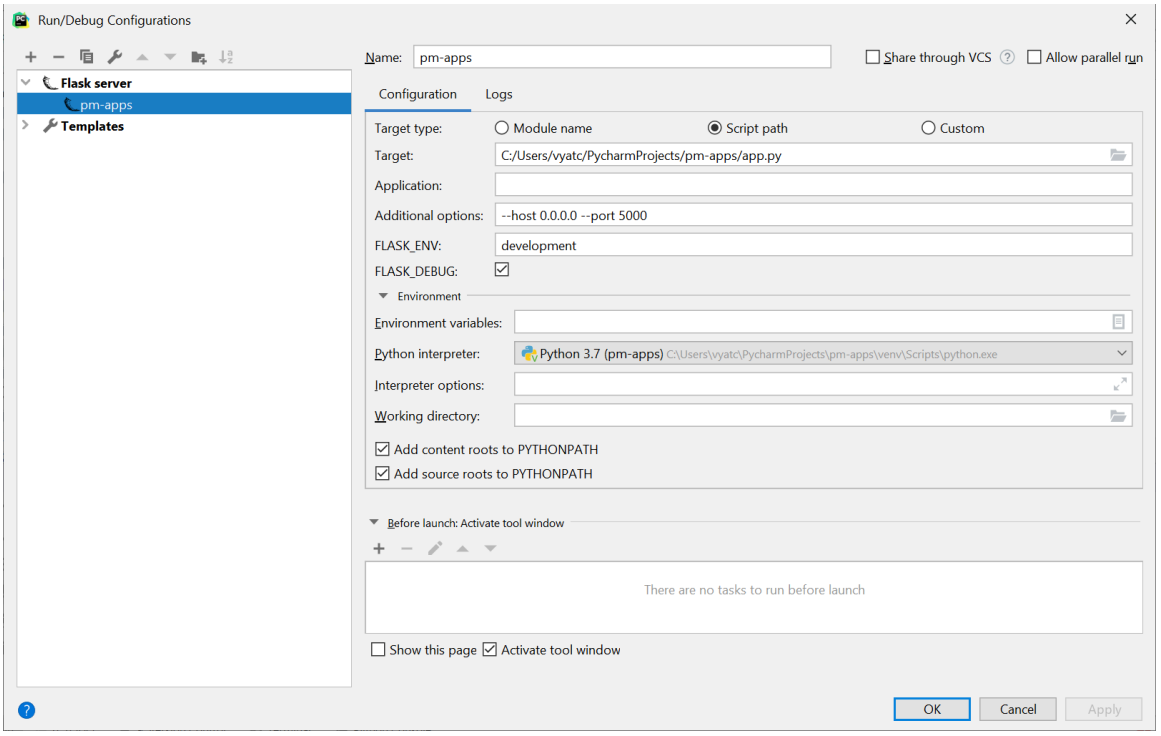

mircealungu
6,971 ● 7 ● 36 ● 45

20

If you're having troubles accessing your Flask server, deployed using **PyCharm**, take the following into account:

PyCharm doesn't run your main .py file directly, so any code in `if __name__ == '__main__':` won't be executed, and any changes (like `app.run(host='0.0.0.0', port=5000)`) won't take effect.

Instead, you should configure the Flask server using Run Configurations, in particular, placing `--host 0.0.0.0 --port 5000` into **Additional options** field.



More about [configuring Flask server in PyCharm]

Share Follow

answered Aug 14, 2019 at 14:58

Viacheslav Zhukov
1,340 11 16

1 Thanks, for anyone working with PyCharm this is a very good info. – Claudiu Aug 30, 2022 at 13:02

This should be one of the top answers. – DEEPAK S.V. Apr 29, 2023 at 22:10

15

You can also set the host (to expose it on a network facing IP address) and port via environment variables.

```
$ export FLASK_APP=app.py
$ export FLASK_ENV=development
$ export FLASK_RUN_PORT=8000
$ export FLASK_RUN_HOST=0.0.0.0

$ flask run
 * Serving Flask app "app.py" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Running on https://0.0.0.0:8000/ (Press CTRL+C to q
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 329-665-000
```

See How to get all available Command Options to set environment variables?

Share Follow

edited Aug 22, 2020 at 12:13

arshovon
13.7k 9 53 72

answered Apr 3, 2020 at 21:48

Everett Toews
10.9k ● 10 ● 47 ● 48

7

Go to your project path on CMD(command Prompt) and execute the following command:-

set FLASK_APP=ABC.py

SET FLASK_ENV=development

flask run -h [yourIP] -p 8080

you will get following o/p on CMD:-

- Serving Flask app "expirement.py" (lazy loading)
  - Environment: development
  - Debug mode: on
  - Restarting with stat
  - Debugger is active!
  - Debugger PIN: 199-519-700
  - Running on http://[yourIP]:8080/ (Press CTRL+C to quit)

Now you can access your flask app on another machine using http://[yourIP]:8080/ url

Share Follow

answered Jun 26, 2019 at 10:27

Ibz.Shaikh
325 ● 2 ● 10

To simplify, please note that you have to use your public IPv4 address. – Andrew Feb 7, 2022 at 13:45

got timeout error on my alt computer – myrccar May 28, 2022 at 17:53

7

For me i followed the above answer and modified it a bit:

1. Just grab your ipv4 address using ipconfig on command prompt
2. Go to the file in which flask code is present
3. In main function write app.run(host= 'your ipv4 address')

Eg:

```
if __name__ == '__main__':
    app.run(host='192.168.0.107')
```

Share Follow

answered Jan 5, 2020 at 15:46

Arsalan Fakhar Siddiqui

146 ● 3 ● 9

6

Create file `.flaskenv` in the project root directory.

The parameters in this file are typically:

```
FLASK_APP=app.py
FLASK_ENV=development
FLASK_RUN_HOST=[dev-host-ip]
FLASK_RUN_PORT=5000
```

If you have a virtual environment, activate it and do a `pip install python-dotenv` .

This package is going to use the `.flaskenv` file, and declarations inside it will be **automatically** imported across terminal sessions.

Then you can do `flask run`

Share Follow

edited Jun 1, 2020 at 20:20

answered May 20, 2020 at 21:33

Saurabh

6,831 ● 5 ● 47 ● 57

4

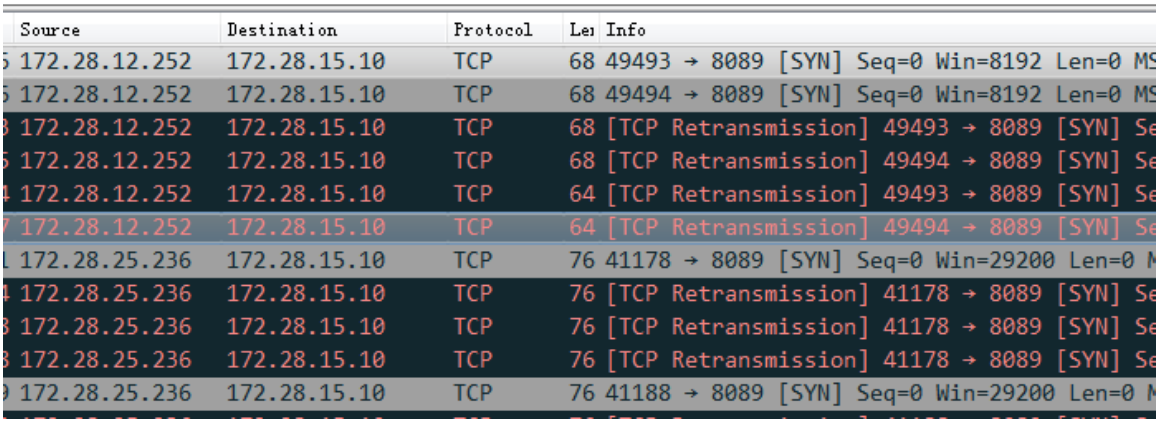
This answer is not solely related with flask, but should be applicable for all *cannot connect service from another host* issue.

1. use `netstat -ano | grep <port>` to see if the address is 0.0.0.0 or ::. If it is 127.0.0.1 then it is only

for the local requests.

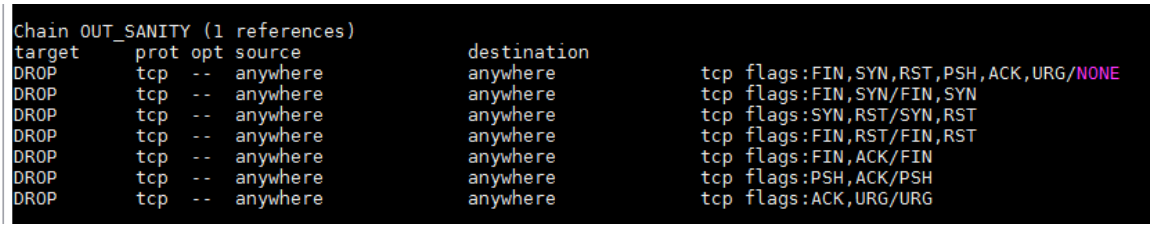
2. use tcpdump to see if any packet is missing. If it shows obvious imbalance, check routing rules by iptables.

Today I run my flask app as usual, but I noticed it cannot connect from other server. Then I run `netstat -ano | grep <port>`, and the local address is `::` or `0.0.0.0` (I tried both, and I know 127.0.0.1 only allows connection from the local host). Then I used `telnet host port`, the result is like `connect to ...`. This is very odd. Then I thought I would better check it with `tcpdump -i any port <port> -w w.pcap`. And I noticed it is all like this:



Source	Destination	Protocol	Le	Info
172.28.12.252	172.28.15.10	TCP	68	49493 → 8089 [SYN] Seq=0 Win=8192 Len=0 MS
172.28.12.252	172.28.15.10	TCP	68	49494 → 8089 [SYN] Seq=0 Win=8192 Len=0 MS
172.28.12.252	172.28.15.10	TCP	68	[TCP Retransmission] 49493 → 8089 [SYN] Se
172.28.12.252	172.28.15.10	TCP	68	[TCP Retransmission] 49494 → 8089 [SYN] Se
172.28.12.252	172.28.15.10	TCP	64	[TCP Retransmission] 49493 → 8089 [SYN] Se
172.28.12.252	172.28.15.10	TCP	64	[TCP Retransmission] 49494 → 8089 [SYN] Se
172.28.25.236	172.28.15.10	TCP	76	41178 → 8089 [SYN] Seq=0 Win=29200 Len=0 M
172.28.25.236	172.28.15.10	TCP	76	[TCP Retransmission] 41178 → 8089 [SYN] Se
172.28.25.236	172.28.15.10	TCP	76	[TCP Retransmission] 41178 → 8089 [SYN] Se
172.28.25.236	172.28.15.10	TCP	76	[TCP Retransmission] 41178 → 8089 [SYN] Se
172.28.25.236	172.28.15.10	TCP	76	41188 → 8089 [SYN] Seq=0 Win=29200 Len=0 M

Then by checking `iptables --list` OUTPUT section, I could see several rules:



```
Chain OUT_SANITY (1 references)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN,RST,PSH,ACK,URG/NONE
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,SYN/FIN,SYN
DROP       tcp  --  anywhere             anywhere             tcp flags:SYN,RST/SYN,RST
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,RST/FIN,RST
DROP       tcp  --  anywhere             anywhere             tcp flags:FIN,ACK/FIN
DROP       tcp  --  anywhere             anywhere             tcp flags:PSH,ACK/PSH
DROP       tcp  --  anywhere             anywhere             tcp flags:ACK,URG/URG
```

these rules forbid output tcp vital packets in handshaking. By deleting them, the problem is gone.

Share Follow

edited Apr 12, 2019 at 2:27

answered Mar 24, 2019 at 8:34

Tiina

4,727 ● 7 ● 50 ● 84

4

This finally worked for me.

```
import os
```

Then place this at the end of your python app.py or main file.

```
if __name__ == "__main__":
    port = int(os.environ.get("PORT", 5000))
    app.run(host='0.0.0.0', port=port)
```

Share Follow

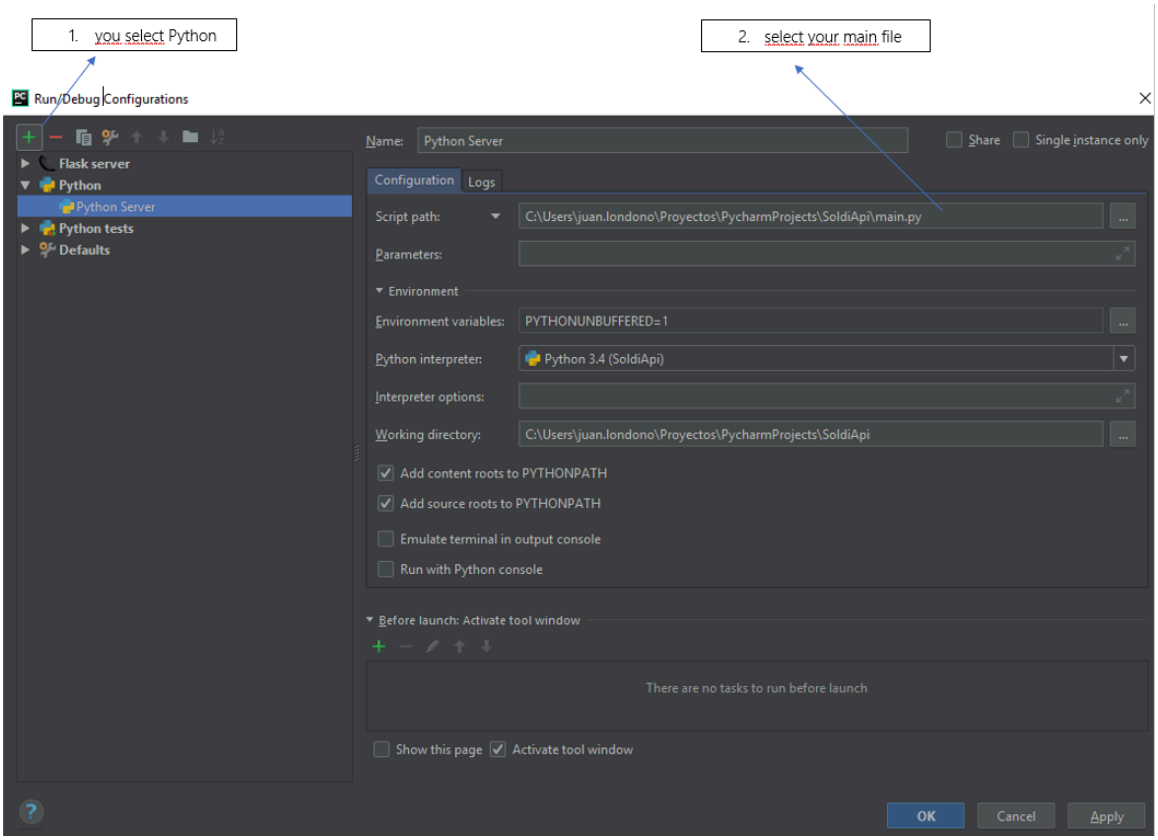
answered Jun 29, 2022 at 0:23

Ronny K

3,731 ● 4 ● 35 ● 43

3

I had the same problem, I use PyCharm as an editor and when I created the project, PyCharm created a Flask Server. What I did was create a server with Python in the following way;



basically what I did was create a new server but flask if not python

I hope it helps you

Share Follow

edited Jul 4, 2018 at 13:46

answered Jul 4, 2018 at 2:02

Juanes30

2,546 ● 2 ● 27 ● 40

2

In my case the problem was the windows firewall, just disable it "while testing".

It's not the optimal way but if you are testing before deploying on a secure server, it could work for you.

Also I'll attach the flask minimal server to test:

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello, World 2!</p>"


if __name__ == '__main__':
    app.run(host="0.0.0.0",debug=True,port=9600)
```

Share Follow

edited Aug 30, 2023 at 1:57

Richard E

3,393 ● 3 ● 17 ● 34

answered Aug 25, 2023 at 23:45

Gman

41 ● 5

1

go to project path set FLASK_APP=ABC.py SET FLASK_ENV=development

flask run -h [yourIP] -p 8080 you will following o/p on CMD:- * Serving Flask app "expirement.py" (lazy loading) * Environment: development * Debug mode: on * Restarting with stat * Debugger is active! * Debugger PIN: 199-519-700 * Running on http://[yourIP]:8080/ (Press CTRL+C to quit)

Share Follow

answered Jun 26, 2019 at 10:21

Ibz.Shaikh

325 ● 2 ● 10

1

In case you need to test your app from an **external** network. Simply serve it to the whole Internet with ngrok.com which will deploy it like a dev server but in no time and locally, saved me a lot of time, and no, I'm not related to that company :)

Just make sure to change the port in your flask app:

```
app.run(host='0.0.0.0', port=80)
```

Share Follow

edited Jul 25, 2022 at 9:58

answered Jul 24, 2022 at 23:35

IdoS

561 ● 1 ● 11 ● 22

0

If none of the above solutions are working, **try manually adding "http://" to the beginning of the url.**

Chrome can distinguish "[ip-address]:5000" from a search query. But sometimes that works for a while, and then stops connecting, seemingly without me changing anything. My hypothesis is that the browser might sometimes automatically prepend http**s**:// (which it shouldn't, but this fixed it in my case).

Share Follow

edited Dec 24, 2020 at 15:58

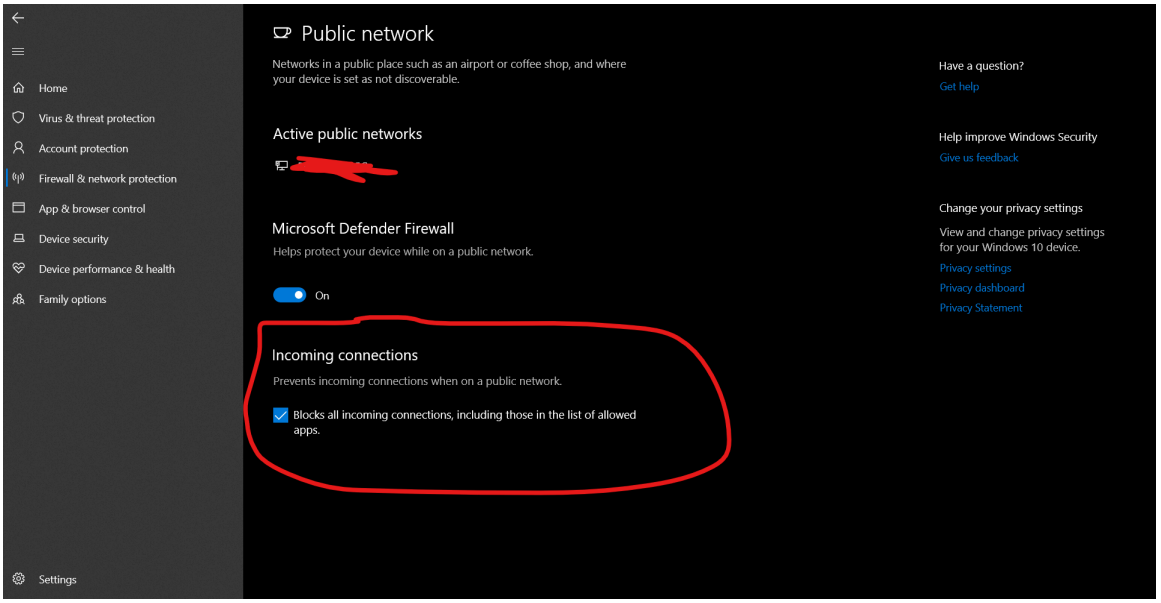
answered Dec 24, 2020 at 15:51

Azeezah M

164 ● 2 ● 9

0

If you are on public Network As I was you have to turn off "block all connections"



Share Follow

answered Aug 18 at 5:54

Coding Dog

18 ● 4

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.