Activity 22.4: Streaming Web Sockets (60:00)

Start Assignment

Due Nov 13 by 11:29am **Points** 20 **Submitting** a file upload
Available after Nov 6 at 8am

- (https://classroom.emeritus.org/courses/9296/modules/items/1556599)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556601)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556602)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556603)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556604)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556606)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556607)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556608)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556609)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556610)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556611)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556612)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556613)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556614)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556615)
- (https://classroom.emeritus.org/courses/9296/modules/items/1556616)
- ps://classroom.emeritus.org/courses/9296/modules/items/1556617)

(https://classroom.emeritus.org/courses/9296/modules/items/1556618)

(https://classroom.emeritus.org/courses/9296/modules/items/1556619)

(https://classroom.emeritus.org/courses/9296/modules/items/1556621)

(https://classroom.emeritus.org/courses/9296/modules/items/1556622)

(https://classroom.emeritus.org/courses/9296/modules/items/1556623)

(https://classroom.emeritus.org/courses/9296/modules/items/1556624)

(https://classroom.emeritus.org/courses/9296/modules/items/1556625)

(https://classroom.emeritus.org/courses/9296/modules/items/1556626)

(https://classroom.emeritus.org/courses/9296/modules/items/1556628)

(https://classroom.emeritus.org/courses/9296/modules/items/1556629)

(https://classroom.emeritus.org/courses/9296/modules/items/1556630)

(https://classroom.emeritus.org/courses/9296/modules/items/1556631)

(https://classroom.emeritus.org/courses/9296/modules/items/1556632)

(https://classroom.emeritus.org/courses/9296/modules/items/1556633)

(https://classroom.emeritus.org/courses/9296/modules/items/1556634)

(https://classroom.emeritus.org/courses/9296/modules/items/1556636)

Learning Outcome Addressed:

6. *Stream* data through web sockets.

In this module, you learned about *streaming* data implementation strategies and how web sockets are beneficial in implementing live data connection apps.

In this activity, you will implement web socket communication by using the Python sockets *library*. Through the steps of this activity, you will execute *packet* transfer from *server* to *client* to help you better understand how *packets* can be created by putting the message length as a header.

To complete this activity, follow these steps:

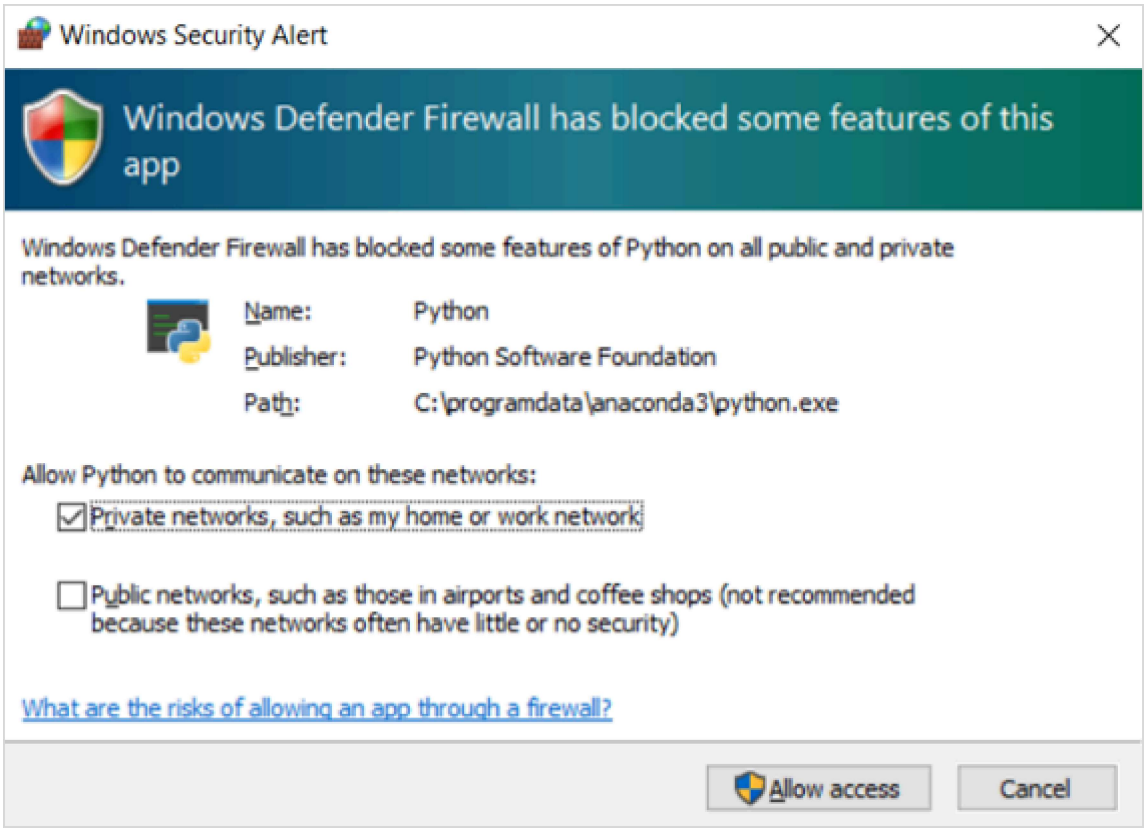
1. Download and extract the Python code from the **johntango Streaming GitHub repository (https://github.com/johntango/Streaming)** to your local machine. Provide a screenshot to show that you have successfully downloaded and extracted the johntango Streaming GitHub repository on your local machine.

2. Open the `streamServer.py` file in VS Code. Provide a screenshot to show that you successfully opened the `streamServer.py` file in VS Code.

3. To start a web socket *server*, execute the `python streamServer.py` command from the Terminal window in VS Code. This will not *print* anything until the *client* connects to the *server*. Provide a screenshot to show that you successfully opened the command prompt in the `streamServer.py` folder and ran the `python streamServer.py` command without errors.

   For Windows users:

   If you are prompted, allow the private networks execution permissions to access web communication port, as shown in the image below:



4. To start a web socket *client*, open a new command line interface in the VS Code Terminal and execute the `python streamClientEx1.py` command from the Terminal. This will establish a connection with the *server* and receive *packets* from the *server*. Provide a screenshot to show that you successfully opened the command prompt in the `streamClientEx1.py` folder, ran the `python streamClientEx1.py` command, and received the message from the *server*.

5. Review the *server* console that indicates that the connection has been established and shows that *packets* that are being sent to the *client*. Provide a screenshot to show that after running the *client* code, you can view the sent messages output over the *server* command line interface.

6. To review another *client* communication, open a new command line interface in the VS Code Terminal and execute the `python streamClient01Soln.py` command from the Terminal. This

will establish another connection with the *server* and receive *packets* from the *server*. Provide a screenshot to show that you successfully opened the command prompt in the `streamClient01Soln.py` file, ran the `python streamClient01Soln.py` command, and can view the received message from the *server*.

7. Open the `streamClient01Soln.py` file. Add a debugging checkpoint at line 13 and run the code to debug it. Provide a screenshot to show that you successfully opened the *client* code in the `streamClient01Soln.py` folder and added a debugging checkpoint at line 13.

8. The previous step will establish a connection with the *server* and receive *packets* from the *server*. The variables listed in the left panel in VS Code when debugging your code show that the value of the Python `msg` variable is "`'29 The ti'`". Provide a screenshot to show that you were able to run the *client* code in debugging mode and see that the value of the Python `msg` variable is "`'29 The ti'`" in VS Code.

You have now completed this activity and practiced implementing web sockets using the socket *library*.

Submission Instructions:

Your submission for this activity should be a Word document that includes the following screenshots, each labeled for the step that the screenshot represents:

1. Provide a screenshot to show that you have successfully downloaded and extracted the johntango Streaming GitHub repository on your local machine.
2. Provide a screenshot to show that you successfully opened the `streamServer.py` file in VS Code.
3. Provide a screenshot to show that you successfully opened the command prompt in the `streamServer.py` folder and ran the `python streamServer.py` command without errors.
4. Provide a screenshot to show that you successfully opened the command prompt in the `streamClientEx1.py` folder, ran the `python streamClientEx1.py` command, and received the message from the *server*.
5. Provide a screenshot to show that after running the *client* code, you can view the sent messages output over the *server* command line interface.
6. Provide a screenshot to show that you successfully opened the command prompt in the `streamClient01Soln.py` file, ran the `python streamClient01Soln.py` command, and can view the received message from the *server*.
7. Provide a screenshot to show that you successfully opened the *client* code in the `streamClient01Soln.py` folder and added a debugging checkpoint at line 13.
8. Provide a screenshot to show that you were able to run the *client* code in debugging mode and see that the value of the Python `msg` variable is "`'29 The ti'`" in VS Code.

Reference

johntango. “Streaming.” *GitHub*. 2021. https://github.com/johntango/Streaming. (https://github.com/johntango/Streaming)

Additional Details:

- **Estimated time:** 60 minutes
- *This is a required activity and counts toward course completion.*

Activity 22.4

Criteria	Ratings		Pts
1. Provide a screenshot to show that you have successfully downloaded and extracted the johntango Streaming GitHub repository on your local machine.	**2 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
2. Provide a screenshot to show that you successfully opened the streamServer.py file in VS Code.	**2 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
3. Provide a screenshot to show that you successfully opened the command prompt in the streamServer.py folder and ran the python streamServer.py command without errors.	**2 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
4. Provide a screenshot to show that you successfully opened the command prompt in the streamClientEx1.py folder, ran the python streamClientEx1.py command, and received the message from the server.	**2 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	2 pts
5. Provide a screenshot to show that after running the client code, you can view the sent messages output over the server command line interface.	**3 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	3 pts
6. Provide a screenshot to show that you successfully opened the command prompt in the streamClient01Soln.py file, ran the python	**3 pts** **Complete**	**0 pts** **Incomplete**	3 pts

Criteria	Ratings		Pts
streamClient01Soln.py command, and can view the received message from the server.	The correct screenshot has been included in the submission.	The screenshot has not been included in the submission or is the incorrect screenshot.	
7. Provide a screenshot to show that you successfully opened the client code in the streamClient01Soln.py folder and added a debugging checkpoint at line 13.	**3 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	3 pts
8. Provide a screenshot to show that you were able to run the client code in debugging mode and see that the value of the Python msg variable is “'29 The ti'” in VS Code.	**3 pts** **Complete** The correct screenshot has been included in the submission.	**0 pts** **Incomplete** The screenshot has not been included in the submission or is the incorrect screenshot.	3 pts
			Total Points: 20

Top Questions

😔 😐 😔

It's all empty here!

If you have any questions ask one