



Use Python to

build a
modern
communication
system



用python构建一个现代通信系统
by: yingshaoxo

ABSTRACT

Python is a good language. Almost every developer know or master this kind of language. In this book, I'll try my best to show you guys how to use Python to build a modern communication system.

In this process, I'll talk about the IP, TCP, Socket, HTTP, WebSocket.

Communication System is based on protocols. Without those protocols, communication can be arbitrary and unreliable.

This book step by step demonstrates how to build a modern communication system with the support of multiple user management. It uses a technique we called RESTful API. So it can almost be used for any programming language and devices like java in android, swift in iPhone, javascript on a web-page.

Keywords: Communication System; Python; RESTful API; Multi-User management

Index

Acknowledgments	VI
About the author	VII
Chapter 1 What is communication	1
1.1 In the ancient	1
1.2 In nowadays	1
1.3 The principle of communication	1
1.3.1 A table for the principle of communication.....	1
1.3.2 Exceptions	2
1.4 Classification of communications	2
1.4.1 Distinguish by the number of end devices	2
1.4.2 Distinguish by the type of information.....	2
1.5 The machine language	3
1.6 Reference	3
Chapter 2 Main protocols in programming for communication	4
2.1 Ethernet	4
2.2 IP	4
2.3 UDP	4
2.4 TCP	4
2.5 HTTP	5
2.6 HTTPS	5
2.7 socket	5
2.8 WebSocket	5
Chapter 3 Socket Communication with Python	6
3.1 A simple program.....	6
3.1.1 server.py	6
3.1.2 client.py.....	6

3.2 A problem about process blocking	7
3.2.1 Thread	7
3.2.2 Multiprocessing	11
Chapter 4 Asynchronous Communication	14
4.1 asyncio	14
4.1.1 A simple example	14
4.2 A simple server and client program	15
4.2.1 server	15
4.2.2 client	15
4.2.3 summary	16
4.3 Keep the Connection or Not	16
Chapter 5 HTTP based Communication	17
5.1 What is API	17
5.2 How to create an API server with Python	17
5.2.1 API server example	17
5.2.2 Test that server	18
Chapter 6 Database based communication	20
6.1 What is MongoDB	20
6.2 How to use MongoDB with Python	20
6.2.1 PyMongo	20
6.2.2 Simple Example	20
6.3 Why we have to use a database	21
Chapter 7 User management system	23
7.1 The basic idea of user management	23
7.2 Implementation	23
7.3 sign up	25
7.4 log in	25
7.5 log out	25
7.6 account deletion	25

Chapter 8 Encryption for communication	26
8.1 Level One: HTTPS	26
8.2 Level Two: Sha256	26
8.2.1 What the server needs to do	26
8.2.2 What the client needs to do	27
8.2.3 Server.py.....	27
8.2.4 Client.py	29
8.2.5 Conclusion	31
Chapter 9 API design	33
9.1 Verification	33
9.2 Get contacts	33
9.3 Send Message	34
9.3.1 Step 1	34
9.3.2 Step 2	34
9.4 Receive Message	36
9.4.1 Step 1	36
9.4.2 Step 2	36
Chapter 10 UI design	38
10.1 Choose a framework.....	38
10.2 Main pages.....	38
10.2.1 Home page	39
10.2.2 Contacts page	40
10.2.3 Chat page	41
Chapter 11 Audio Chat	43
11.1 Voice recording	43
11.1.1 HTML	43
11.1.2 Javascript	43
11.2 Audio transferring	44
11.2.1 Change in data structure	44
11.2.2 Change in API requiring.....	45
11.3 Screenshot	47

11.3.1 User 1	47
11.3.2 User 2	48
References	49

Acknowledgments

I really have to thank **Google and Open Source Community** who did a lot to let knowledge be equally delivered to everyone in this world who wants to learn. Without them, I can't be as knowledgeable as I am today.

Thanks to **my parents** who give me food so I can stay away from starving.

Thanks to **TeX**, it saves me a lot of time for the writing of this book.

About the author

Resume

Name: yingshaoxo (or 胡英杰)

Gender: Male

Birth Day: 1998.3.29

I'm a self-taught product manager, front end designer, back end developer, automation engineer, operations engineer.

I love freedom, hate censorship and authority.

Links

Email: yingshaoxo@gmail.com

Personal Website: <https://yingshaoxo.xyz>

Github: <https://github.com/yingshaoxo>

Twitter: <https://twitter.com/yingshaoxo>

Youtube: <https://www.youtube.com/channel/UCbT9GDmkqf555ATReJor6ag>

Source Code for This Book

<https://github.com/yingshaoxo/use-python-to-build-a-modern-communication-system>

Chapter 1 What is communication

We'll talk about the definition of communication.

1.1 In the ancient

It is an action for people to communicate with each other. One man talks to another man.

1.2 In nowadays

I would say it's a concept of information exchange.

1.3 The principle of communication

If I have to use one word to describe it, that'll be 'Stable'.

During the transmission process, you shouldn't lose any information that was originally sent.

And also, it should be as fast as possible. Say, if you have an urgent message to send to others, you definitely don't wanna others to receive that message ten days later.

And also, security is important too. You don't wanna send your message to a wrong guy, it'll cause you a lot of trouble.

1.3.1 A table for the principle of communication

principles	description
integrity	never lose information
speed	provide information at time
security	make sure no one else could see it!

1.3.2 Exceptions

But there are still have a few communication types that only follow 1 or 2 principles of communication.

For example, broadcast. As long as you send the message out, whether the audience gets that message or not, it's not your business. Because the user have a choice about when to listen to your broadcast, when not to.

Radio for entertainment doesn't need security, but radio for the military does need to secure its message.

1.4 Classification of communications

1.4.1 Distinguish by the number of end devices

Some people like to classify communication with the number of sender or receiver.



It's OK for doing that classification. 'One to one' also called 'Peer to Peer', 'One to many' also called broadcast.

1.4.2 Distinguish by the type of information

Some other people like to classify communication with the type of information being transferred.



1.5 The machine language

At the above Explanation, we only covered the basics of communication. Or how humans understood that.

In this section, I'll talk about how a machine sees that problem. Because in the end, in modern society, communication is based on computers, not humans.

For a computer, he itself is a computer, the one that he communicates with, is also a computer. So the speed of communication can be very high. At least it'll faster than human-being.

Then the problem left is how a computer could understand each other, and most importantly, to cooperate in the right way so information can flow freely and correctly.

Example 1.1: For example, if we got a phone and a computer, how do we communicate with each other? (A phone itself is also a computer actually.)

We need to create a certain method for communicating, that method is called *protocol*.

Example 1.2: How to make sure all the information we sent is correctly received by others?

We need to create a certain method to make sure the integrity of information in transferring, that method is a subsection of a *protocol*.

Question 1.1: Is there a best *protocol* for using?

No! For different scenarios, there are different transmission protocols to choose from.

1.6 Reference

For more information, check this Wikipedia ^[1]