

Simple index.http file

```
#!/usr/bin/env httpx --httpfile
### get my internet ip
# @name myip
GET https://httpbin.org/ip
```

HTTP syntax: <https://www.jetbrains.com/help/idea/exploring-http-syntax.html>

Where to load http file?

1. Assigned by --justfile option
2. Find index.http in current directory
3. Try to find index.http in parent dir until root
4. Load default \$HOME/.httpx/index.http

Test services in http file

```
$ httpx --list
$ httpx myip
$ echo hello | httpx post
$ httpx post --data
@/path/to/file
```

Zsh completion

```
$ httpx --completions zsh
```

Generate HTTP file from OpenAPI

```
httpx -f demo.http --import
https://api.example.com/openapi.jsonn "name": "Jackie "
```

REST API

```
### get my internet ip
# @name myip
GET https://httpbin.org/ip
### inspection http post
# @name post
POST https://httpbin.org/post
```

REST API (cont)

```
> Content-Type: application/json
[1]
```

Email

```
### send an email by Gmail
//@name mail
MAIL mailto:demo@example.com
Host: tls://smtp.gmail.com:587
Authorization: Basicyour_name@gmail.com:google_app_password
From: your_name@gmail.com
Subject: e-nice to meet you
Content-Type: text/plain
Hi Master:
    bala bala ...
Best regards
Yours sincerely Zombie
```

gRPC

```
### grpc call SayHello
//@name SayHello
GRPC localhost:50052 /Greeter/SayHello
{
    "name": "Jackie "
}
```

GraphQL

```
### graphql query
GRAPHQL https://countries.es.torvalds.com/
Content-Type: application/graphql

query { continents { name } }
### graphql query over WebSocket
GRAPHQL ws://localhost:4000/graphql
```

GraphQL (cont)

```
> Content-Type: application/graphql
subscription { greetings }
### graphql over RSocket
GRAPHQL rsocketws://localhost:8080/rsocket/graphql
Content-Type: application/graphql

query { continents { name } }
```

RSocket

```
### RSocket Request
//@name rsocket-request
RSocket com.example.UseRService.findById
Host: 127.0.0.1:42252
Content-Type: application/json
1
### RSocket Request
//@name rsocket-stream
STREAM com.example.UseRService.findAll
Host: ws://127.0.0.1:8080/rsocket
Content-Type: application/json
"vip"
```

Apache Dubbo

```
### Dubbo sayHi
//@name sayHi
DUBBO 127.0.0.1:20880 /Greeter/sayHi(-java.lang.String)
Content-Type: application/json
"jackie "
```

ZeroMQ

```
### zeromq request
//@name zero-req
ZEROREQ 127.0.0.1 :5555
Content-Type: application/json
" Jackie "
### subscribe zeromq
//@name zero-sub
SUB topic1
Host: zeromq :// localhost -
ost :5555
```

Apache Thrift

```
### thrift request
THRIFT 127.0.0.1 :9090/ getUser
Content-Type: application/json
-
{
    " 1": {"i3 2": 1}
}
```

Kafka

```
### publish kafka message
//@name kafka-pub
PUB topic-1
Host: kafka: //localhost:9092
Content-Type: application/json
{
    " name": " Jackie "
}
### subscribe kafka topic
//@name kafka-sub
SUB topic-1
Host: kafka: //localhost:9092
```

Apache Pulsar

```
### publish kafka message
//@name pulsar-pub
PUB topic-1
Host: pulsar :// localhost -
ost :6650
Content-Type: application/json
{
    " name": " Jackie "
}
### subscribe Pulsar topic
//@name pulsar-sub
SUB topic-1
Host: pulsar :// localhost -
ost :6650
```

RabbitMQ

```
### publish RabbitMQ message
//@name rabbit-pub
PUB queue1
Host: amqp:/localhost:5672
Content-Type: application/json
{
    " name": " Jackie "
}
### Subscribe RabbitMQ queue
//@name rabbit-sub
SUB queue1
Host: amqp:/localhost:5672
```

Redis Pub/Sub

```
### publish redis message
//@name redis-pub
PUB channel1
Host: redis: //localhost:6379
```

Redis Pub/Sub (cont)

```
> Content-Type: application/json
{
    "name": "Jackie"
}
### subscribe redis
//@name redis-sub
SUB channel1
Host: redis://localhost:6379
```

Pub/Sub more

- * Nats
- * AMQP, MQTT, Stomp
- * Aliyun MNS/Event Bridge
- * AWS SNS/SQS/EventBridge



By **linux_china**

cheatography.com/linux-china/

Published 6th February, 2022.

Last updated 18th July, 2022.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>