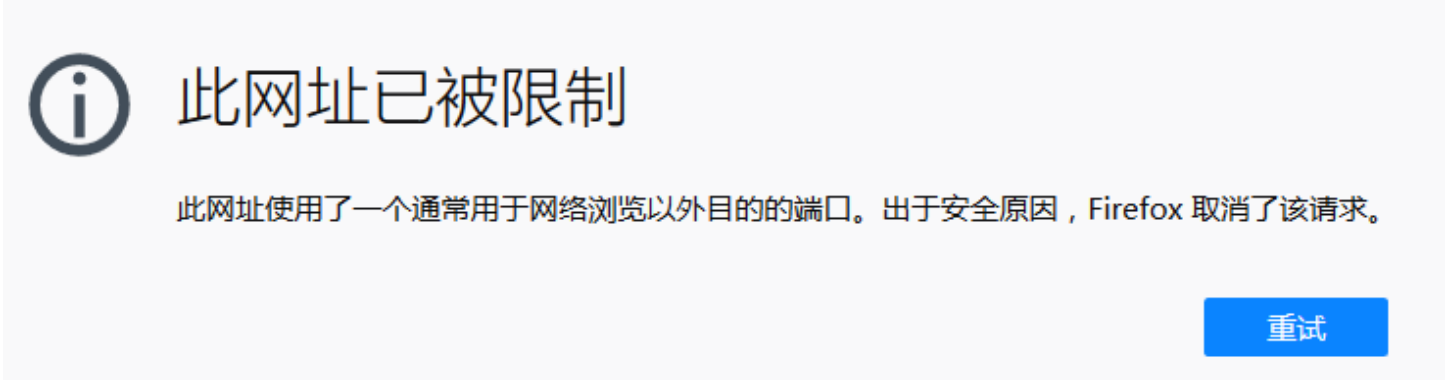
浏览器阻止的不安全端口

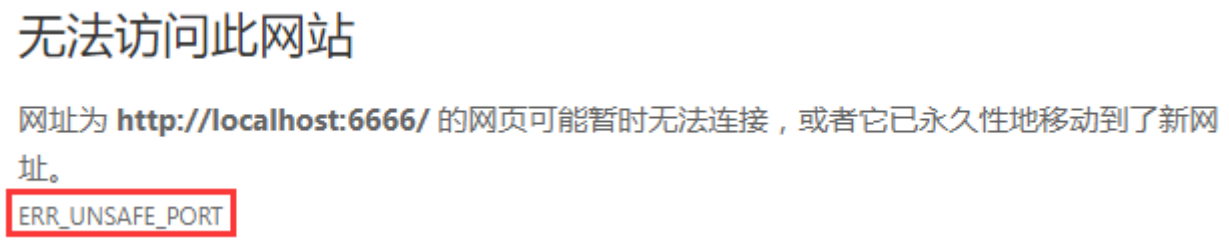
今天在写一个http服务的时候，随手用了一个6666的端口，然而在Chrome中访问时却显示**无法访问此网站**。
首先我怀疑可能是http服务没有正常启动，但是并没有发现程序报错，并且我在cmd下运行：

```
netstat -ano|findstr "6666"
```

发现6666端口处于监听状态。
那么换一个端口试试吧，我随便换了一个端口，再访问就正常了，感觉是6666这个端口有问题，什么原因呢？
我又改回6666，改用Firefox访问:



Firefox的提示比较明显，浏览器出于安全考虑而阻止了一些可能不安全的端口。我又回去看了一下Chrome，原来是我太粗心，没有仔细看报错信息：



在superuser (https://superuser.com/questions/188058/which-ports-are-considered-unsafe-on-chrome)上找到了Chrome的不安全端口列表，然后Firefox的在这里(https://developer.mozilla.org/en-US/docs/Mozilla/Mozilla_Port_Blocking)。

这个限制并不是强制的，可以通过修改浏览器的设置来取消对这些端口的阻止，但是对外提供的服务，不可能让用户去修改他们浏览器中的这些设置，因此应该避免在程序中使用这些端口。
为了通用性，避免使用的端口集合应该是上面两份（或者更多）列表的并集，我用 `kotlin` 写了一段代码，进行了合并操作，虽然写代码可能比手工合并用的时间还长，但是这么做更像一个程序员，哈哈。

代码如下：

```
val chromeBlock = """
  1,    // tcpmux
  7,    // echo
  9,    // discard
  11,   // systat
  13,   // daytime
  15,   // netstat
  17,   // qotd
  19,   // chargen
  20,   // ftp data
  21,   // ftp access
  22,   // ssh
  23,   // telnet
  25,   // smtp
  37,   // time
  42,   // name
  43,   // nicname
  53,   // domain
  77,   // priv-rjs
  79,   // finger
  87,   // ttylink
  95,   // supdup
  101,  // hostriame
  102,  // iso-tsap
  103,  // gppitnp
  104,  // acr-nema
  109,  // pop2
  110,  // pop3
  111,  // sunrpc
  113,  // auth
  115,  // sftp
  117,  // uucp-path
  119,  // nntp
  123,  // NTP
  135,  // loc-srv /epmap
  139,  // netbios
  143,  // imap2
  179,  // BGP
  389,  // ldap
  465,  // smtp+ssl
  512,  // print / exec
  513,  // login
  514,  // shell
```

```
  515,  // printer
  526,  // tempo
  530,  // courier
  531,  // chat
  532,  // netnews
  540,  // uucp
  556,  // remotefs
  563,  // nntp+ssl
  587,  // stmp?
  601,  // ??
  636,  // ldap+ssl
  993,  // ldap+ssl
  995,  // pop3+ssl
  2049, // nfs
  3659, // apple-sasl / PasswordServer
  4045, // lockd
  6000, // X11
  6665, // Alternate IRC [Apple addition]
  6666, // Alternate IRC [Apple addition]
  6667, // Standard IRC [Apple addition]
  6668, // Alternate IRC [Apple addition]
  6669, // Alternate IRC [Apple addition]
""".trimIndent()

val firefoxList = """
  1     tcpmux
  7     echo
  9     discard
  11    systat
  13    daytime
  15    netstat
  17    qotd
  19    chargen
  20    ftp data
  21    ftp control
  22    ssh
  23    telnet
  25    smtp
  37    time
  42    name
  43    nicname
  53    domain
  77    priv-rjs
  79    finger
```

```
  87    ttylink
  95    supdup
  101   hostriame
  102   iso-tsap
  103   gppitnp
  104   acr-nema
  109   POP2
  110   POP3
  111   sunrpc
  113   auth
  115   sftp
  117   uucp-path
  119   NNTP
  123   NTP
  135   loc-srv / epmap
  139   netbios
  143   IMAP2
  179   BGP
  389   LDAP
  465   SMTP+SSL
  512   print / exec
  513   login
  514   shell
  515   printer
  526   tempo
  530   courier
  531   chat
  532   netnews
  540   uucp
  556   remotefs
  563   NNTP+SSL
  587   submission
  601   syslog
  636   LDAP+SSL
  993   IMAP+SSL
  995   POP3+SSL
  2049  nfs
  4045  lockd
  6000  X11
""".trimIndent()

data class Block(val id: String, var name: String)

fun addBlock(map: HashMap<String, Block>, block: Block) {
```

```
    map[block.id]?.let {
      if (it.name.compareTo(block.name, true) != 0) {
        it.name += " / " + block.name
      }
    } ?: {
      map[block.id] = block
    }()
  }

  fun main(args: Array<String>) {

    val map: HashMap<String, Block> = hashMapOf()
    chromeBlock.split("\n").forEach {
      val str = it.replace(",", "").replace("//", "")
      val arr = str.split("\\s+".toRegex(), 2)
      if (arr.size != 2) {
        println("chromeBlock split error, need size 2, get size: ${arr.size}")
        return
      }
      addBlock(map, Block(arr[0], arr[1]))
    }
    firefoxList.split("\n").forEach {
      val arr = it.split("\\s+".toRegex(), 2)
      if (arr.size != 2) {
        println("firefoxList split error, need size 2, get size: ${arr.size}")
        return
      }
      addBlock(map, Block(arr[0], arr[1]))
    }
    map.entries.sortedBy { it.key.toInt() }.forEach {
      println("${it.key} ${it.value.name}")
    }
  }
```

最终得到的列表如下：

1 tcpmux
7 echo
9 discard
11 systat
13 daytime
15 netstat
17 qotd
19 chargen
20 ftp data
21 ftp access / ftp control
22 ssh
23 telnet
25 smtp
37 time
42 name
43 nicname
53 domain
77 priv-rjs
79 finger
87 ttylink
95 supdup
101 hostriame
102 iso-tsap
103 gppitnp
104 acr-nema
109 pop2
110 pop3
111 sunrpc
113 auth
115 sftp
117 uucp-path
119 nntp
123 NTP
135 loc-srv /epmap / loc-srv / epmap
139 netbios
143 imap2
179 BGP
389 ldap
465 smtp+ssl
512 print / exec
513 login
514 shell
515 printer

```
526 tempo
530 courier
531 chat
532 netnews
540 uucp
556 remotefs
563 nntp+ssl
587 stmp? / submission
601 ?? / syslog
636 ldap+ssl
993 ldap+ssl / IMAP+SSL
995 pop3+ssl
2049 nfs
3659 apple-sasl / PasswordServer
4045 lockd
6000 X11
6665 Alternate IRC [Apple addition]
6666 Alternate IRC [Apple addition]
6667 Standard IRC [Apple addition]
6668 Alternate IRC [Apple addition]
6669 Alternate IRC [Apple addition]
```

最后，我用Microsoft Edge又试了一次，上面列表中的大部分端口都可以访问，这算优点还是缺点呢？

TAGS

network (https://dafengge0913.github.io/tags/network/)

TAG
golang (https://dafengge0913.github.io/tags/golang)
docker (https://dafengge0913.github.io/tags/docker)
java (https://dafengge0913.github.io/tags/java)
network (https://dafengge0913.github.io/tags/network)

windows (https://dafengge0913.github.io/tags/windows)

algorithm (https://dafengge0913.github.io/tags/algorithm)

jvm (https://dafengge0913.github.io/tags/jvm)

linux (https://dafengge0913.github.io/tags/linux)

mongodb (https://dafengge0913.github.io/tags/mongodb)

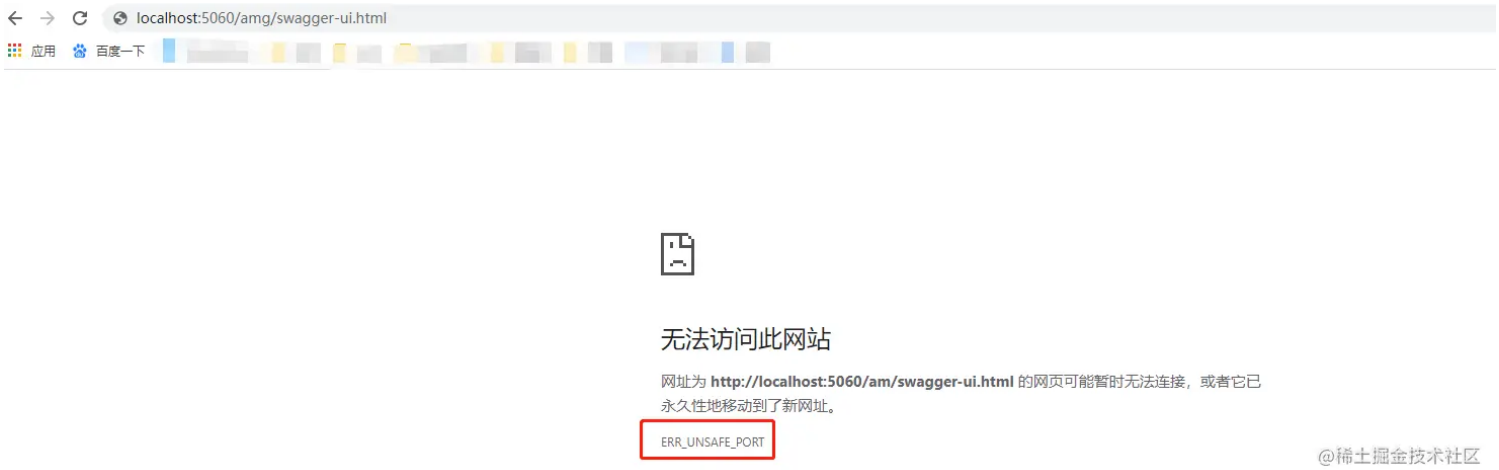
protobuf (https://dafengge0913.github.io/tags/protobuf)

ERR_UNSAFE_PORT 非安全端口

预立科技 2021-08-04 09:38 264

+ 关注

chrome访问地址：http://localhost:5060，提示"ERR_UNSAFE_PORT"



但是在Edge里面可以打开

原因是chrome浏览器将一些端口号(5060,6666等)默认为非安全端口，禁止访问

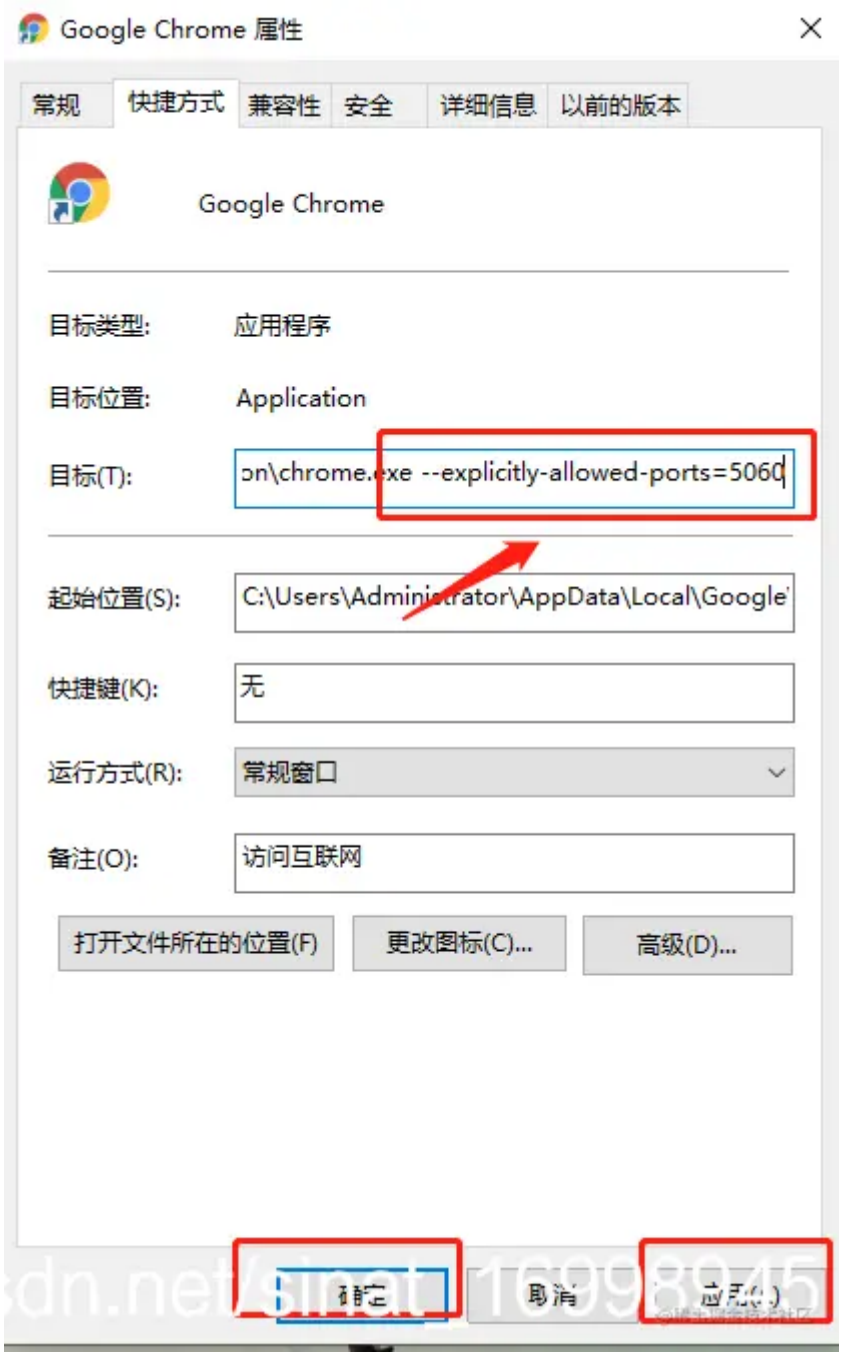
2种解决办法：

1.更换端口号，如果不能更换端口号，可以尝试第二种

2.桌面右键chrome浏览器，点击属性，目标路径后，追加，空格 + “–explicitly-allowed-ports=5060”
如果多个端口用逗号分割例如：

```java
--explicitly-allowed-ports=5060,6666
```

"-- "前注意需要空格，空格不能缺



重启浏览器，重新访问

标签： Java