

hw5: MicroServices & Security

微服务

HTTPS

增加HTTPS通信功能后，运行的区别

reference

请你在大二开发的E-Book系统的基础上，完成下列任务：

1. 下面两项任务你可以选择一项完成：

- ① 开发一个微服务，输入为书名，输出为书的作者。将此微服务单独部署，并使用netflix-zuul进行路由，在你的E-Book系统中使用该服务来完成作者搜索功能。
- ② 开发一个函数式服务，输入为订单中每种书的价格和数量，输出为订单的总价。将此函数式服务单独部署，并在你的E-Book系统中使用该服务来完成作者搜索功能。

2. 在你的工程中增加HTTPS通信功能，并且观察程序运行时有什么不同。请你编写文档，将程序运行时的过程截图，并解释为什么会出现和之前不同的差异。

-请将你的工程所有源代码和资源文件压缩后上传，请勿压缩编译后生成文件和依赖的第三方包

-关于第2点的文档一并压缩提交。

微服务

客户端添加eureka的dependency



```
<version>8.2.0</version>
</dependency>
<dependency>
<groupId>org.springframework.cloud</groupId>
<artifactId>spring-cloud-starter-netflix-eureka-client </artifactId>
<version>2.2.1.RELEASE</version>
</dependency>
</dependencies>
```

这里考虑了一本书名可能有好几个版本的情况，因此测试了数据库放了两本同书名的书。

GET <http://localhost:11230/getAuthorByBookName?bookName=Three Body> Send

Params ● Auth Headers (8) Body ● Pre-req. Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edi
<input checked="" type="checkbox"/>	bookName	Three Body			
	Key	Value	Description		

Body ▾

Pretty Raw Preview Visualize JSON

```

1  []
2    "Liu Cixion",
3    "Liu Cixion"
4  []

```

访问独立端口

Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
BOOK-SERVICE	n/a (1)	(1)	UP (1) - 192.168.16.107:book-service:11230
GATEWAY	n/a (1)	(1)	UP (1) - 192.168.16.107:gateway:8080
OTHER-SERVICE	n/a (1)	(1)	UP (1) - 192.168.16.107:other-service:9090

注册到eureka上面

通过gateway访问

GET

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> username	reins
<input checked="" type="checkbox"/> password	123
Key	Value

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   "Liu Cixion",
3   "Liu Cixion"
4 ]

```

其他服务 (Book Store)也通过gateway进行了路由，相应的前端页面的url都要改了。

http://localhost:8080/user/getUser

GET

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> username	reins			
<input checked="" type="checkbox"/> password	123			
Key	Value	Description		

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```

1 [
2   {
3     "userId": 1,
4     "name": "echo",
5     "type": 1,
6     "email": "819601183@qq.com"
7   },
8   {
9     "userId": 2,
10    "name": "admin",
11    "type": 0,
12    "email": "819601192@qq.com"
13  }
14 ]

```

HTTPS

1. 命令行生成keystore, 秘钥库指令是自己输入的, 这里是testkey.

```
→ 20200703_bookstore_backend git:(main) ✘ keytool -genkey -v -alias testKey -keyalg RSA -validity 3650 -keystore ./key/test.keystore
输入密钥库口令：
再次输入新口令：
您的名字与姓氏是什么？
[Unknown]: Olivia
您的组织单位名称是什么？
[Unknown]: SJTU
您的组织名称是什么？
[Unknown]: SJTU
您所在的城市或区域名称是什么？
[Unknown]: Shanghai
您所在的省/市/自治区名称是什么？
[Unknown]: Shanghai
该单位的双字母国家/地区代码是什么？
[Unknown]: CC
CN=Olivia, OU=SJTU, O=SJTU, L=Shanghai, ST=Shanghai, C=CC是否正确？
[否]: y

正在为以下对象生成 2,048 位RSA密钥对和自签名证书 (SHA256withRSA) (有效期为 3,650 天):
CN=Olivia, OU=SJTU, O=SJTU, L=Shanghai, ST=Shanghai, C=CC
[正在存储 ./key/test.keystore]
```

查看证书

```
→ key git:(main) ✘ keytool -list -v -keystore test.keystore
输入密钥库口令：
密钥库类型：PKCS12
密钥库提供方：SUN

您的密钥库包含 1 个条目

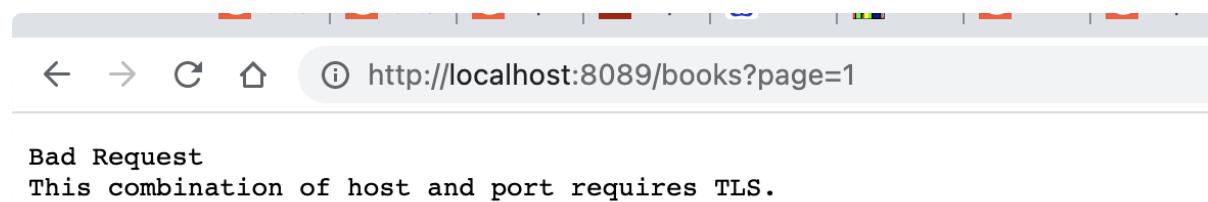
别名：testkey
创建日期：2021年10月24日
条目类型：PrivateKeyEntry
证书链长度：1
证书 [1]：
所有者：CN=Olivia, OU=SJTU, O=SJTU, L=Shanghai, ST=Shanghai, C=CC
发布者：CN=Olivia, OU=SJTU, O=SJTU, L=Shanghai, ST=Shanghai, C=CC
序列号：dc8511ef312d0191
生效时间：Sun Oct 24 12:54:53 CST 2021, 失效时间：Wed Oct 22 12:54:53 CST 2031
证书指纹：
SHA1: 8C:DC:38:4B:53:41:EF:9D:71:01:4C:FA:8C:BA:DC:BC:F3:10:61:21
SHA256: 06:5F:87:CC:31:6A:04:DD:CC:91:C2:BE:17:5A:8A:6F:C5:0D:60:2E:7D:C2:A1:2A:6B:C5:17:28:1A:AD:A5:43
签名算法名称：SHA256withRSA
主体公共密钥算法：2048 位 RSA 密钥
版本：3

扩展：

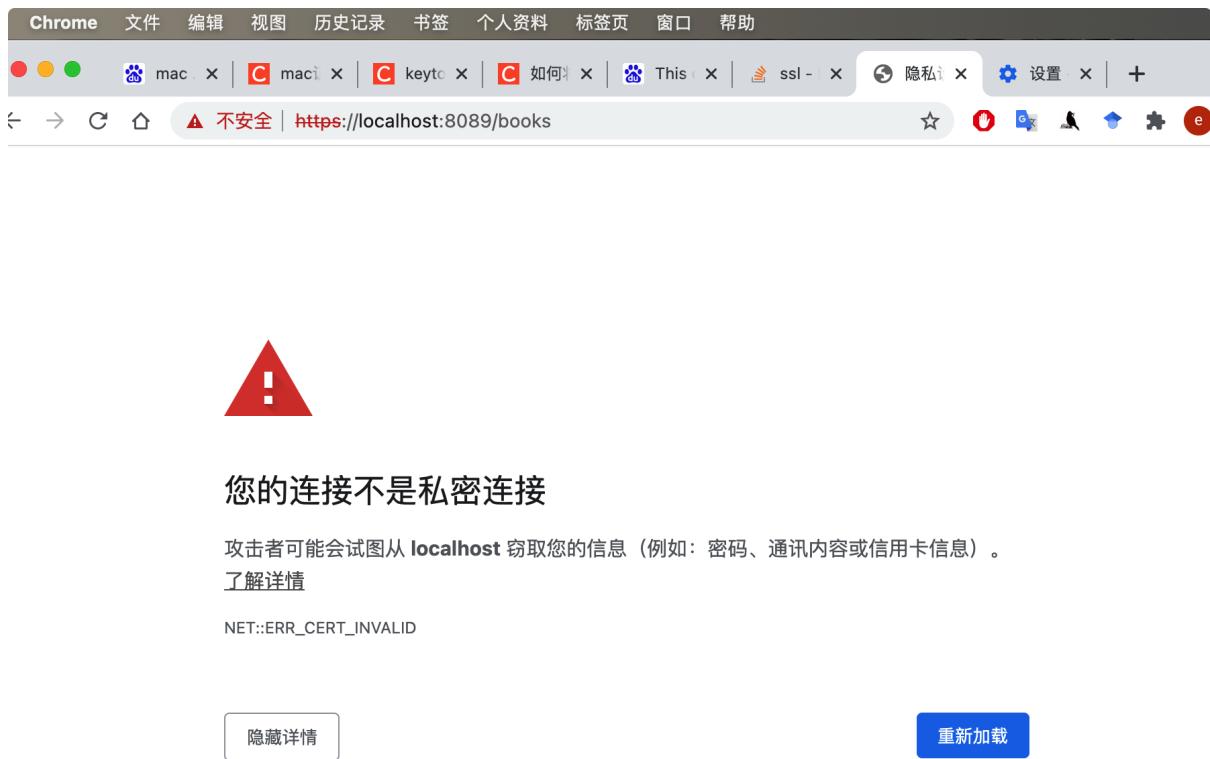
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 5B B1 D9 E9 C5 1F 15 59  31 73 77 49 2F FD 7D C6  [.....Y1swI/...
0010: 89 04 E6 89                               ....
]
]

*****
*****
```

这是访问http显示如下，需要改用https

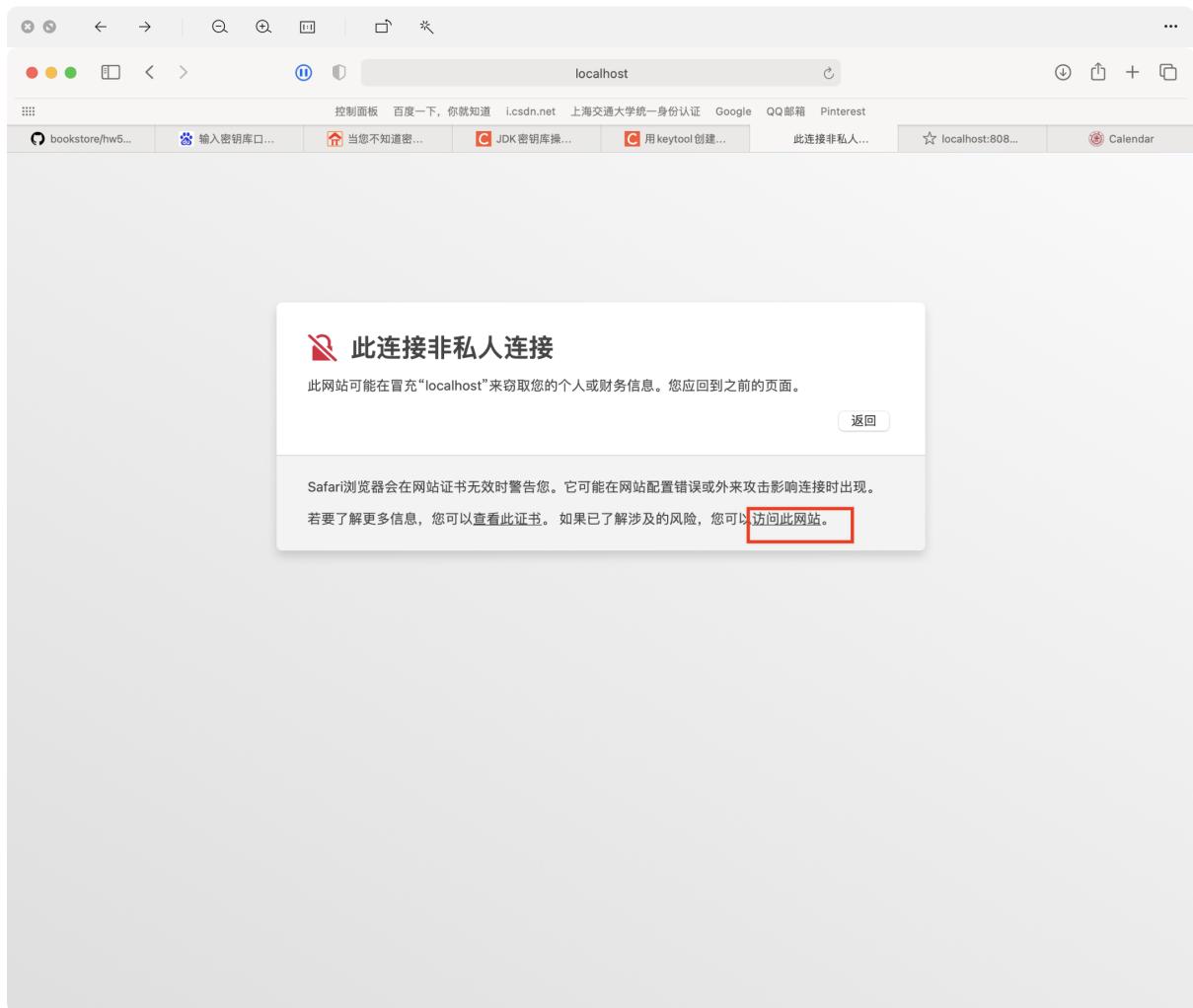


即使改到了https 也还不能访问 (safari可以点击信任证书)



localhost 通常会使用加密技术来保护您的信息。Chrome 此次尝试连接到 localhost 时，该网站发回了异常的错误凭据。这可能是因为有攻击者在试图冒充 localhost，或者 Wi-Fi 登录屏幕中断了此次连接。请放心，您的信息仍然是安全的，因为 Chrome 尚未进行任何数据交换便停止了连接。

您目前无法访问localhost，因为此网站发送了Chrome无法处理的杂乱凭据。网络错误和攻击通常是暂时的，因此，此网页稍后可能会恢复正常。



2. 导出.cer 证书文件

```
keytool -export -alias testKey -keystore test.keystore -rfc -file keystore.cer
```

-alias testKey, testKey是别名，这个和之前的设置证书的-alias要一样。

-keystore test.keystore 对哪个文件进行导出

-rfc 指定可以查看编码的方式输出

-file keystore.cer 设置导出后的文件名

```
→ key git:(main) ✘ keytool -export -alias testKey -keystore test.keystore -rfc -file keystore.cer
输入密钥库口令：
存储在文件 <keystore.cer> 中的证书
```

```
Web → key git:(main) ✘ ls  
keystore.cer test.keystore
```



keystore.cer

参考：keyTool操作合集

keytool常用操作_这是一个懒人的博客-CSDN博客
genkey 在用户主目录 -genkey 在用户主目录中创建一个默认文件".keystore",还会产生一个mykey的别名， mykey中包含用户的公钥、私钥和证书(在没有指定生成位置的情况下,keystore会存在
C https://blog.csdn.net/qq_30062125/article/details/86717827?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1.no_

原创

The screenshot shows a Java KeyStore management interface. On the left, there's a sidebar with categories: '默认钥匙串' (Default Keystore), '登录' (Login), '本地项目' (Local Project) which is selected, '系统钥匙串' (System Keystore), '系统' (System), and '系统根证书' (System Root Certificate). The main area is titled '钥匙串访问' (Keystore Access) and has tabs for '所有项目' (All Projects), '密码' (Password), '安全备注' (Security Notes), '我的证书' (My Certificates), '密钥' (Keys), and '证书' (Certificates), with '证书' being the active tab. A search bar at the top right contains the text 'oli'. Below the tabs is a table with columns: '名称' (Name), '种类' (Type), '钥匙串' (Keystore), and '过期时间' (Expiration Date). The table contains one row for a certificate named 'Olivia'.

名称	种类	钥匙串	过期时间
Olivia	证书	登录	2031年10月22日 下午12:...



Olivia

自行签名的根证书

过期时间: 2031年10月22日 星期三 中国标准时间 下午12:

⚠ 此证书尚未经过第三方验证

名称

种类



Olivia

新建证书偏好设置...

拷贝“Olivia”

删除“Olivia”

导出“Olivia”...

显示简介

评估“Olivia”...



```

[{"id": 1, "bookId": 1, "name": "Elon Musk", "type": "biographies", "author": "Ashlee Vance", "price": 3299, "description": "Elon Reeve Musk FRS (/i:lɒn/ EE-lon; born June 28, 1971) is an entrepreneur and business magnate. He is the founder, CEO, and chief engineer at SpaceX; early stage investor, [note 1] CEO, and product architect of Tesla, Inc.; founder of The Boring Company; and co-founder of Neuralink and OpenAI. A centibillionaire, Musk is one of the richest people in the world.\n\nMusk was born to a Canadian mother and South African father and raised in Pretoria, South Africa. He briefly attended the University of Pretoria before moving to Canada aged 17 to attend Queen's University. He transferred to the University of Pennsylvania two years later, where he received bachelor's degrees in economics and physics. He moved to California in 1995 to attend Stanford University but decided instead to pursue a business career, co-founding the web software company Zip2 with his brother Kimbal. The startup was acquired by Compaq for $307 million in 1999. Musk co-founded online bank X.com that same year, which merged with Confinity in 2000 to form PayPal. The company was bought by eBay in 2002 for $1.5 billion.", "inventory": 12315, "image": "http://r.photo.store.qq.com/psc?/V11fv0hk0pCaQ1/TmEUgtj9EK6.7V8ajmQrEJ1jWqo10KoaK32vU0MRaxhtyMkCxyDvO9gaoBk+p5pD8upIZ3Lbd5dF26nGzngyyHB.sr5lJmKeU56GEwMg4!/r"}, {"id": 2, "bookId": 5, "isbn": "978-7-18618-5", "name": "A Promise Land", "type": "biographies", "author": "Barack Obama", "price": 1233, "description": "Barack Obama was the 44th president of the United States, elected in November 2008 and holding office for two terms. He is the author of two previous New York Times bestselling books, Dreams from My Father and The Audacity of Hope, and the recipient of the 2009 Nobel Peace Prize. He lives in Washington, D.C., with his wife, Michelle. They have two daughters, Malia and Sasha.", "inventory": 23445, "image": "http://r.photo.store.qq.com/psc?/V11fv0hk0pCaQ1/TmEUgtj9EK6.7V8ajmQrEJrA8.qbAN0xfZS2HF3hy0GdwhvB73TwUD.6ki.mnv4M2mv5QZ95HsF6.un5Ru1Edz7JIDXMwGa2AJGPJB*Y9p0!/r"}]

```

增加HTTPS通信功能后，运行的区别

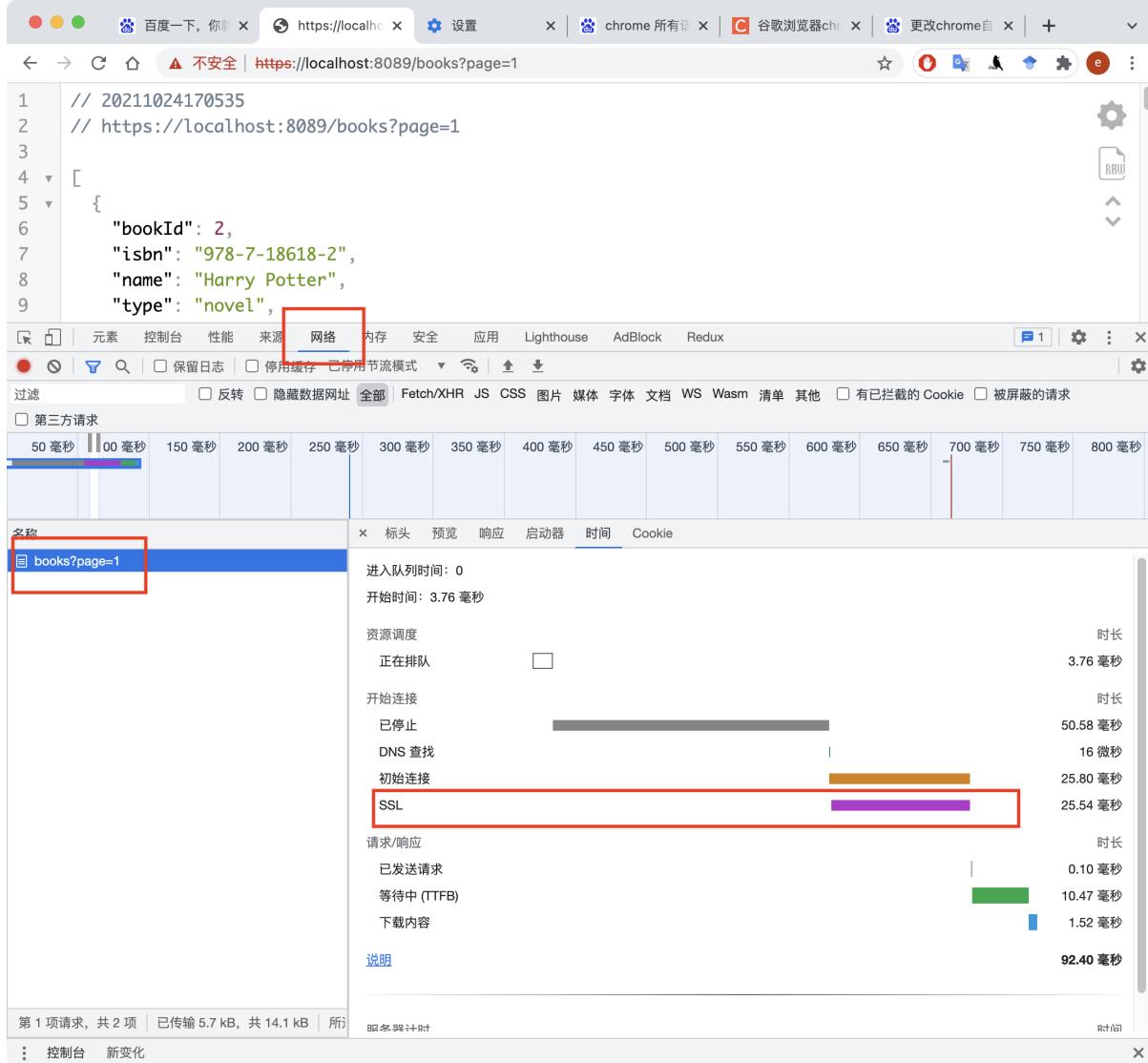
1. 输入域名之前是http,现在变成了https

见下图，http升级为https的方法是部署SSL证书。而SSL证书可以正规的CA机构申请，也可以自己导出。本次作业使用jdk自带的工具生成本地的ssl证书

HTTPS协议和HTTP协议的区别：

- https协议需要到ca申请证书，一般免费证书很少，需要交费。
- http是超文本传输协议，信息是明文传输，https 则是具有安全性的ssl加密传输协议。
- http和https使用的是完全不同的连接方式用的端口也不一样,前者是80,后者是443。
- http的连接很简单,是无状态的。
- HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，要比http协议安全。

2. 连接过程中会多一个SSL握手



类似于TCP的3次握手建立TCP连接，SSL握手是用于建立SSL（Security Socket Layer）层的连接。SSL握手的场景很多，比如最常见的HTTPS，在进行HTTPS的应用数据传递之前，需要建立SSL的连接。

上图就进行了一次SSL握手。

TLS/SSL中使用了非对称加密，对称加密以及HASH算法。握手过程的具体描述如下：

- 1) 浏览器将自己支持的一套加密规则发送给网站。
- 2) 网站从中选出一组加密算法与HASH算法，并将自己的身份信息以证书的形式发回给浏览器。证书里面包含了网站地址，加密公钥，以及证书的颁发机构等信息。

- 3) 浏览器获得网站证书之后浏览器要做以下工作：a) 验证证书的合法性（颁发证书的机构是否合法，证书中包含的网站地址是否与正在访问的地址一致等），如果证书受信任，则浏览器栏里面会显示一个小锁头，否则会给出证书不受信的提示。b) 如果证书受信任，或者是用户接受了不受信的证书，浏览器会生成一串随机数的密码，并用证书中提供的公钥加密。c) 使用约定好的HASH算法计算握手消息，并使用生成的随机数对消息进行加密，最后将之前生成的所有信息发送给网站。
- 4) 网站接收浏览器发来的数据之后要做以下的操作：a) 使用自己的私钥将信息解密取出密码，使用密码解密浏览器发来的握手消息，并验证HASH是否与浏览器发来的一致。b) 使用密码加密一段握手消息，发送给浏览器。
- 5) 浏览器解密并计算握手消息的HASH，如果与服务端发来的HASH一致，此时握手过程结束，之后所有的通信数据将由之前浏览器生成的随机密码并利用对称加密算法进行加密。

这里浏览器与网站互相发送加密的握手消息并验证，目的是为了保证双方都获得了一致的密码，并且可以正常的加密解密数据，为后续真正数据的传输做一次测试。另外，HTTPS一般使用的加密与HASH算法如下：

- 非对称加密算法：RSA， DSA/DSS
- 对称加密算法：AES， RC4， 3DES
- HASH算法：MD5， SHA1， SHA256

3. 需要增加证书认证

在你的初始尝试通过安全连接与webserver通信时，该服务器将以“证书”的形式向你的web浏览器提供一组凭证，作为该网站声称是**谁和什么网站**的证明。

HTTPS核心的一个部分是数据传输之前的握手，握手过程中确定了数据加密的密码。在握手过程中，网站会向浏览器发送SSL证书，SSL证书和我们日常用的身份认证类似，是一个支持HTTPS网站的身份证明，SSL证书里面包含了网站的域名，证书有效期，证书的颁发机构以及用于加密传输密码的公钥等信息，由于公钥加密的密码只能被在申请证书时生成的私钥解密，因此浏览器在生成密码之前需要先核对当前访问的域名与证书上绑定的域名是否一致，同时还要对证书的颁发机构进行验证，如果验证失败浏览器会给出证书错误的提示。

在本次实现中，就导出了自己的证书，放入了本地的证书保存库，然后设置为始终信任，这样就使得证书被导入了浏览器（mac是这样）



keystore.cer

Olivia
自行签名的根证书
过期时间: 2031年10月22日 星期三 中国标准时间 下午12:54:53
此证书已标记为受此帐户信任

信任

使用此证书时: 始终信任 ?

加密套接字协议层 (SSL)	始终信任
安全邮件 (S/MIME)	始终信任
可扩展认证协议 (EAP)	始终信任
IP 安全 (IPsec)	始终信任
代码签名	始终信任
时间戳	始终信任
X.509 基本策略	始终信任

细节

主题名称 _____
国家或地区 CC
省/市/自治区 Shanghai
所在地 Shanghai
组织 SJTU

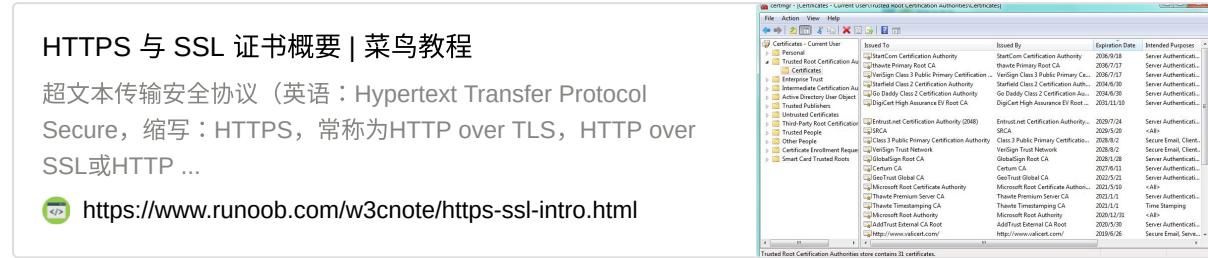
reference

mac 导入证书方式

mac下chrome导入burp证书

MacOS上使用BurpSuite拦截HTTPS流量请求会出现站点不信任的情况，需要导入BurpSuite的证书。
Proxy >> Options >> Import/export CA certificate => 如下 保存为.crt文件后缀，如burp.crt。谷歌
浏览器切换到证书管理： 将导出的证书拖拽到证书Tab，此时根证书是不被信任的。 右键选中证书
🔗 <https://www.cnblogs.com/Hi-blog/p/How-To-Import-BurpSuite-Certificate-To-Chrome-On-MacOS.html>

HTTPS 与 SSL 证书概要



The screenshot shows the Windows Certificates dialog box. The left pane lists categories: Personal, Trusted Root Certification Authorities, Enterprise Trust, Internet Explorer Certification Authorities, Active Directory User Objects, Trusted Publishers, Untrusted Publishers, and Third Party Root Certificates. The right pane displays a detailed list of certificates, each with columns for Issued To, Issued By, Expiration Date, and Intended Purpose. The list includes well-known authorities like StartCom, Thawte, GeoTrust, and DigiCert.

Issued To	Issued By	Expiration Date	Intended Purpose
StartCom Certification Authority	StartCom Certification Authority	2036/9/18	Server Authentication
Thawte Primary Root CA	Thawte Primary Root CA	2036/7/31	Server Authentication
VeriSign Class 3 Public Primary Certification Authority	VeriSign Class 3 Public Primary Certification Authority	2036/5/31	Server Authentication
GeoTrust Class 2 Certification Authority	GeoTrust Class 2 Certification Authority	2034/6/30	Server Authentication
Go Daddy Class 2 Certification Authority	Go Daddy Class 2 Certification Authority	2034/6/30	Server Authentication
DigiCert High Assurance EV Root CA	DigiCert High Assurance EV Root CA	2032/12/31	Server Authentication
Entrust.net Certification Authority (208)	Entrust.net Certification Authority	2029/7/24	Server Authentication
Class 3 Public Primary Certification Authority	Class 3 Public Primary Certification Authority	2028/8/2	Secure Email, Client
VeriSign Trust Network	VeriSign Trust Network	2028/8/2	Secure Email, Client
GeoTrust Global Root CA	GeoTrust Global Root CA	2028/6/2	Server Authentication
Cetain CA	Cetain CA	2027/6/31	Server Authentication
GeoTrust Global CA	GeoTrust Global CA	2022/5/21	Server Authentication
Thawte Premium Certificate Authority	Thawte Premium Certificate Authority	2021/12/31	Server Authentication
Thawte Premium Server CA	Thawte Premium Server CA	2021/3/1	Server Authentication
Thawte Timestamping CA	Thawte Timestamping CA	2021/1/1	Time Stamping
GeoTrust Global	GeoTrust Global	2020/6/30	Server Authentication
AdTrust External CA Root	AdTrust External CA Root	2020/5/29	Server Authentication
Http://www.vaciet.com/	Http://www.vaciet.com/	2019/6/26	Secure Email, Service

HTTPS 与 SSL 证书概要 | 菜鸟教程

超文本传输安全协议（英语：Hypertext Transfer Protocol）

Secure，缩写：HTTPS，常称为HTTP over TLS，HTTP over SSL或HTTP ...

 <https://www.runoob.com/w3cnote/https-ssl-intro.html>