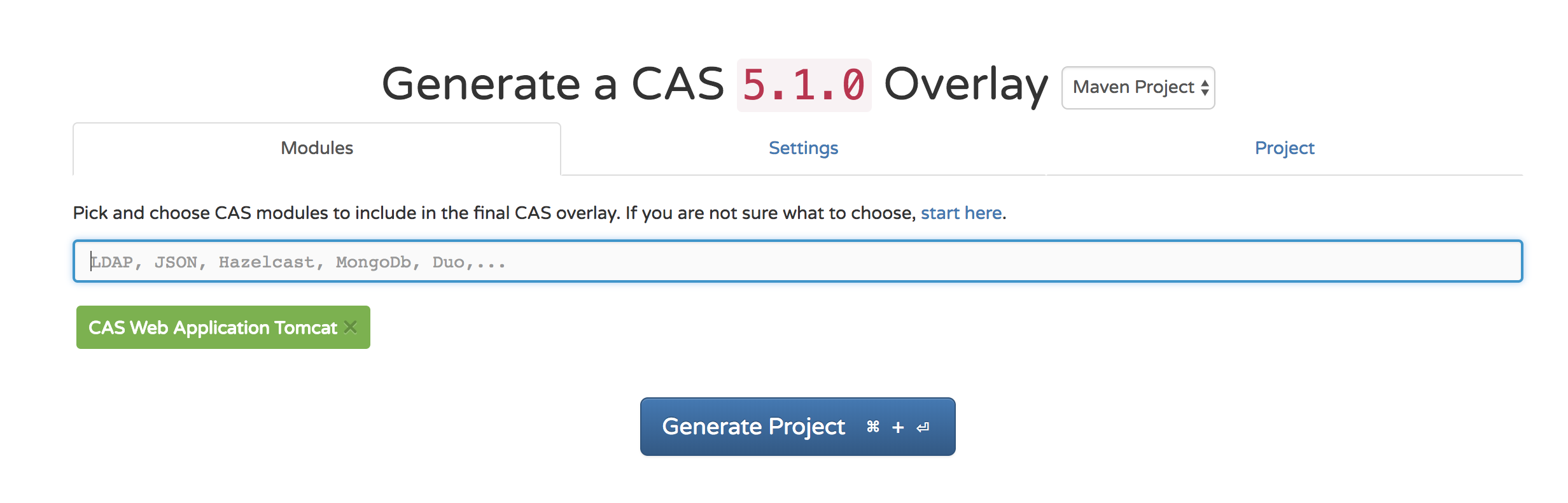
https://casinitializr.herokuapp.com/

利用骨架创建cas服务器项目 maven

# 1、骨架网站 ，类似于spring boot项目网站

## <https://casinitializr.herokuapp.com/>

## 1、tomcat





## 2、project 添加项目的配置 sso-server，妈的我打开不开，所以就没配置，直接下载的别人的成熟的代码

# 2、添加一个父工程

*<?*xml version="1.0" encoding="UTF-8"*?>*<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.carl.auth</groupId>  
 <artifactId>sso</artifactId>  
 <version>1.0.0</version>  
 <packaging>pom</packaging>  
  
 <modules>  
 <module>sso-server</module>  
 </modules>  
  
  
 <properties>  
 <java.version>1.8</java.version>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 <cas.version>5.1.3</cas.version>  
 <springboot.version>1.5.3.RELEASE</springboot.version>  
 </properties>  
  
  
</project>

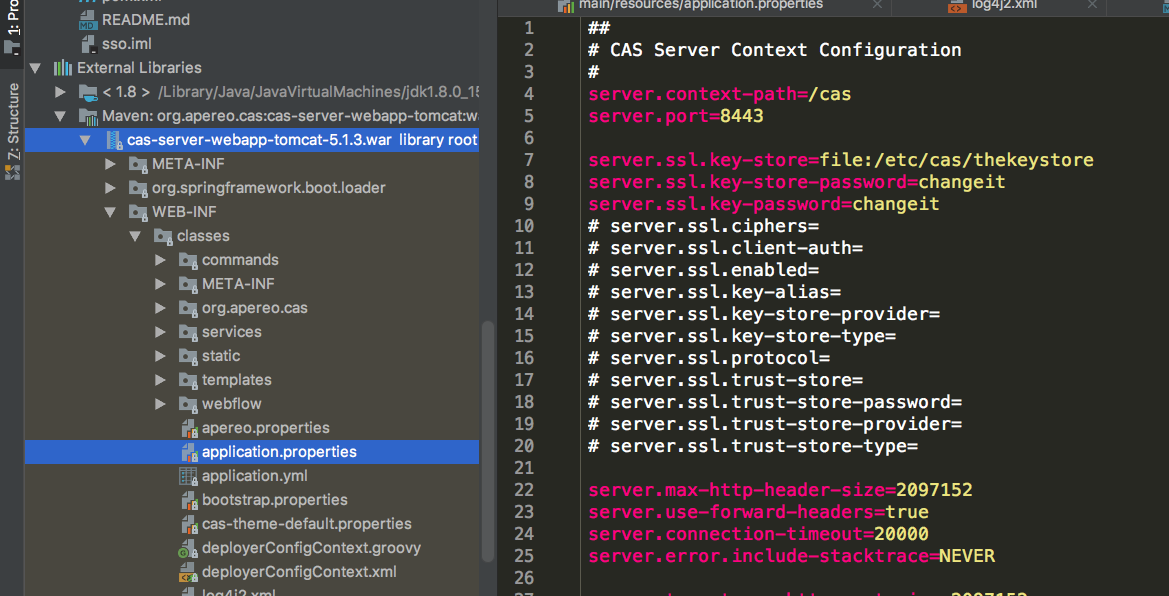
# 3、创建子工程（也就是上面骨架生成的工程sso-server）

## 1、依赖如下

*<?*xml version="1.0" encoding="UTF-8"*?>*<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <parent>  
 <groupId>com.carl.auth</groupId>  
 <artifactId>sso</artifactId>  
 <version>1.0.0</version>  
 <relativePath>../pom.xml</relativePath>  
 </parent>  
  
 <artifactId>sso-server</artifactId>  
 <packaging>war</packaging>  
  
 <name>sso-server</name>  
 <description>CAS认证服务，负责各系统的鉴权的鉴权</description>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.apereo.cas</groupId>  
 <artifactId>cas-server-webapp-tomcat</artifactId>  
 <version>${cas.version}</version>  
 <type>war</type>  
 <scope>runtime</scope>  
 </dependency>  
 </dependencies>  
 <dependencyManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.apereo.cas</groupId>  
 <artifactId>cas-server-support-bom</artifactId>  
 <version>${cas.version}</version>  
 <type>pom</type>  
 <scope>import</scope>  
 </dependency>  
 </dependencies>  
 </dependencyManagement>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>com.rimerosolutions.maven.plugins</groupId>  
 <artifactId>wrapper-maven-plugin</artifactId>  
 <version>0.0.5</version>  
 <configuration>  
 <verifyDownload>true</verifyDownload>  
 <checksumAlgorithm>MD5</checksumAlgorithm>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 <version>${springboot.version}</version>  
 <configuration>  
 <mainClass>org.springframework.boot.loader.WarLauncher</mainClass>  
 <addResources>true</addResources>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>3.1.0</version>  
 <configuration>  
 <warName>cas</warName>  
 <failOnMissingWebXml>false</failOnMissingWebXml>  
 <recompressZippedFiles>false</recompressZippedFiles>  
 <archive>  
 <compress>false</compress>  
 <manifestFile>${project.build.directory}/war/work/org.apereo.cas/cas-server-webapp-tomcat/META-INF/MANIFEST.MF</manifestFile>  
 </archive>  
 <overlays>  
 <overlay>  
 <groupId>org.apereo.cas</groupId>  
 <artifactId>cas-server-webapp-tomcat</artifactId>  
 </overlay>  
 </overlays>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.3</version>  
 </plugin>  
 </plugins>  
 <finalName>cas</finalName>  
 </build>  
</project>

## 2、application.properties  配置

### 将cas-server-webapp-tomcat/5.1.3/cas-server-webapp-tomcat-5.1.3.war!/WEB-INF/classes/application.properties  拷贝到src/main/resources 下并且进行调整，



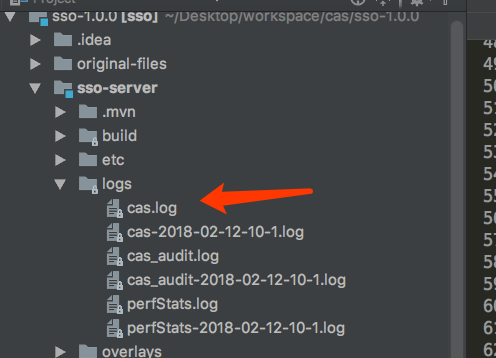
## 因为没有证书现在，所以先将证书设置为false，等将来有了证书再设置为true

server.context-path=/cas  
server.port=8444  
  
server.ssl.enabled=false  
  
server.max-http-header-size=2097152  
server.use-forward-headers=true  
server.connection-timeout=20000  
server.error.include-stacktrace=NEVER  
  
server.tomcat.max-http-post-size=2097152  
server.tomcat.basedir=build/tomcat  
server.tomcat.accesslog.enabled=true  
server.tomcat.accesslog.pattern=%t %a "%r" %s (%D ms)  
server.tomcat.accesslog.suffix=.log  
server.tomcat.max-threads=10  
server.tomcat.port-header=X-Forwarded-Port  
server.tomcat.protocol-header=X-Forwarded-Proto  
server.tomcat.protocol-header-https-value=https  
server.tomcat.remote-ip-header=X-FORWARDED-FOR  
server.tomcat.uri-encoding=UTF-8  
  
spring.http.encoding.charset=UTF-8  
spring.http.encoding.enabled=true  
spring.http.encoding.force=true  
  
##  
# CAS Cloud Bus Configuration  
#  
spring.cloud.bus.enabled=false  
  
endpoints.enabled=false  
endpoints.sensitive=true  
  
endpoints.restart.enabled=false  
endpoints.shutdown.enabled=false  
  
management.security.enabled=true  
management.security.roles=ACTUATOR,ADMIN  
management.security.sessions=if\_required  
management.context-path=/status  
management.add-application-context-header=false  
  
security.basic.authorize-mode=role  
security.basic.enabled=false  
security.basic.path=/cas/status/\*\*  
  
##  
# CAS Web Application Session Configuration  
#  
server.session.timeout=300  
server.session.cookie.http-only=true  
server.session.tracking-modes=COOKIE  
  
##  
# CAS Thymeleaf View Configuration  
#  
spring.thymeleaf.encoding=UTF-8  
spring.thymeleaf.cache=true  
spring.thymeleaf.mode=HTML  
##  
# CAS Log4j Configuration  
#  
# logging.config=file:/etc/cas/log4j2.xml  
server.context-parameters.isLog4jAutoInitializationDisabled=true  
  
##  
# CAS AspectJ Configuration  
#  
spring.aop.auto=true  
spring.aop.proxy-target-class=true  
  
##  
# CAS Authentication Credentials  
#  
cas.authn.accept.users=casuser::Mellon

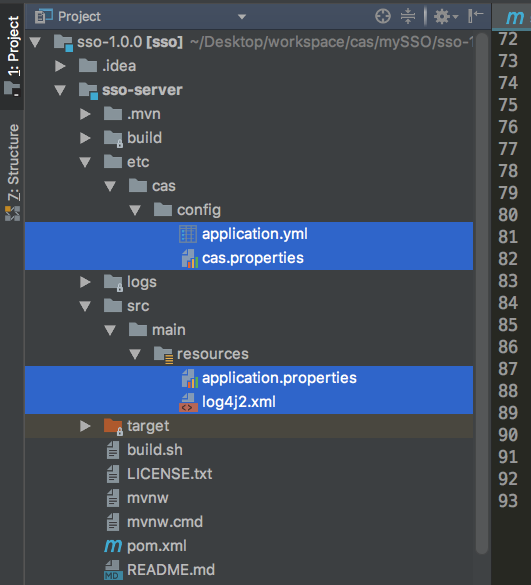
## 3、日志文件调整：

### 1、把etc/cas/config/log4j2.xml 放到项目的src/main/resources 下，

### 2、内容由<Property name="cas.log.dir" >.</Property> 调整成 <Property name="cas.log.dir" >logs</Property>  源码就不贴出来了，需要的自行下载

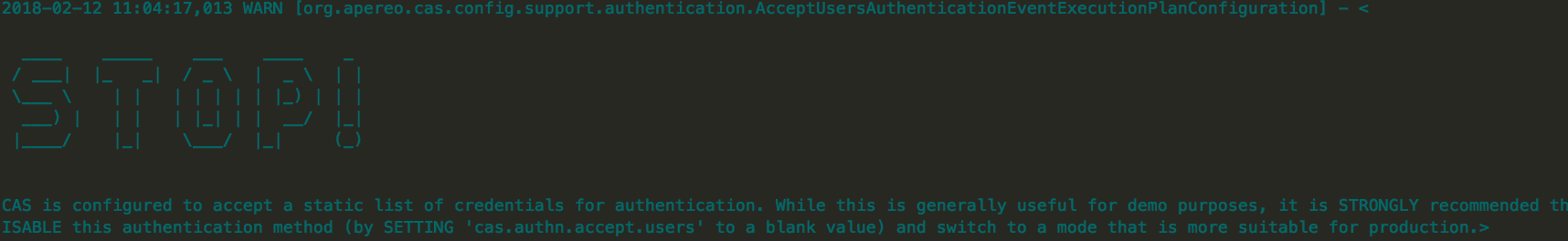


### 3、项目中etc 不能删掉，但是可以删掉里面没有用的东西，剩余项目的代码如下



# 4、启动这个工程（mac上启动）

## sudo ./build.sh run



## 2、浏览器访问 （用户名密码默认用户名/密码为：casuser/Mellon）

##  
# CAS Authentication Credentials  
#  
cas.authn.accept.users=casuser::Mellon

### http://localhost:8444/cas/login

