|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **User Guide – Steps to create CSR** | | | |  |  |  | |   **Draft Version**  **18-August-2015** |
|  |

|  |  |
| --- | --- |
| Version: | 1.0 |
| Date: | 26 March 2019 |
| Approval State: |  |
| Confidentiality Category: | For Internal Use Only |

This book is copyright reserved by Tetherfi and may not be photocopied or taken out of the physical boundaries of the organization.

# Create Self Signed certificate:

Run command: keytool -genkey -alias tomcat -keyalg RSA -keystore projectname.jks

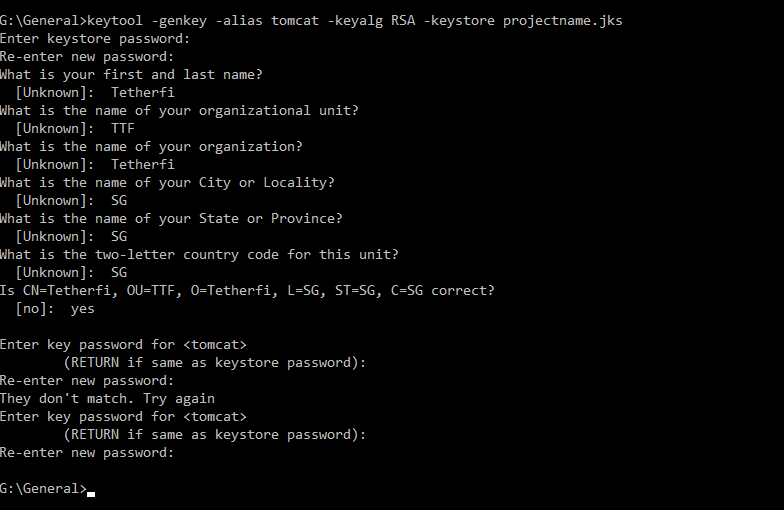
Alias name and jks name can be updated based on project requirement. It is provided by customer.

Java is set in my classpath and so I can run it directly. Otherwise full path has to be given Ex: “C:\Java\jdk1.8.0\_151\bin\keytool.exe”.

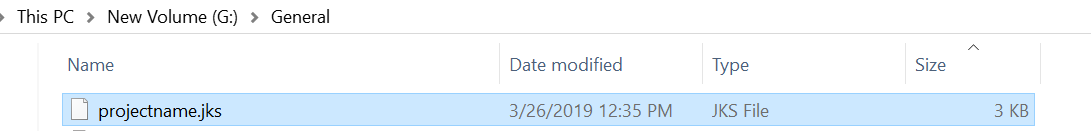
**If require SAN detail as part of CSR**, add attribute SAN

keytool -genkey -alias tomcat -keyalg RSA -keystore projectname.jks -ext SAN=dns:commonname

It prompts for password to protect keystore. Take note on the password given.



Jks file is created.



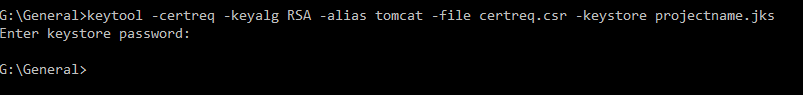
# Create CSR file:

Run below command to create CSR from JKS.

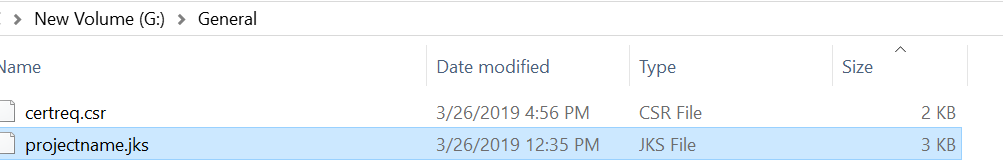
keytool -certreq -keyalg RSA -alias tomcat -file certreq.csr -keystore projectname.jks

**If require SAN detail as part of CSR**, add attribute SAN

keytool -certreq -keyalg RSA -alias tomcat -file certreq.csr -keystore projectname.jks -ext SAN=dns:commonname



It creates the csr file.



It has to be shared with customer to sign the certificate.

Once the cert is signed, root, intermediate and server certificate will be shared.

It has to be imported one by one in JKS to make it signed.

# Command to import cert

keytool -import -trustcacerts -file Root-CA.crt -alias root -keystore projectname.jks

It would prompt for password. Enter password and yes to import it.

