luv2code

Secure Communication with HTTPS

Secure Communication

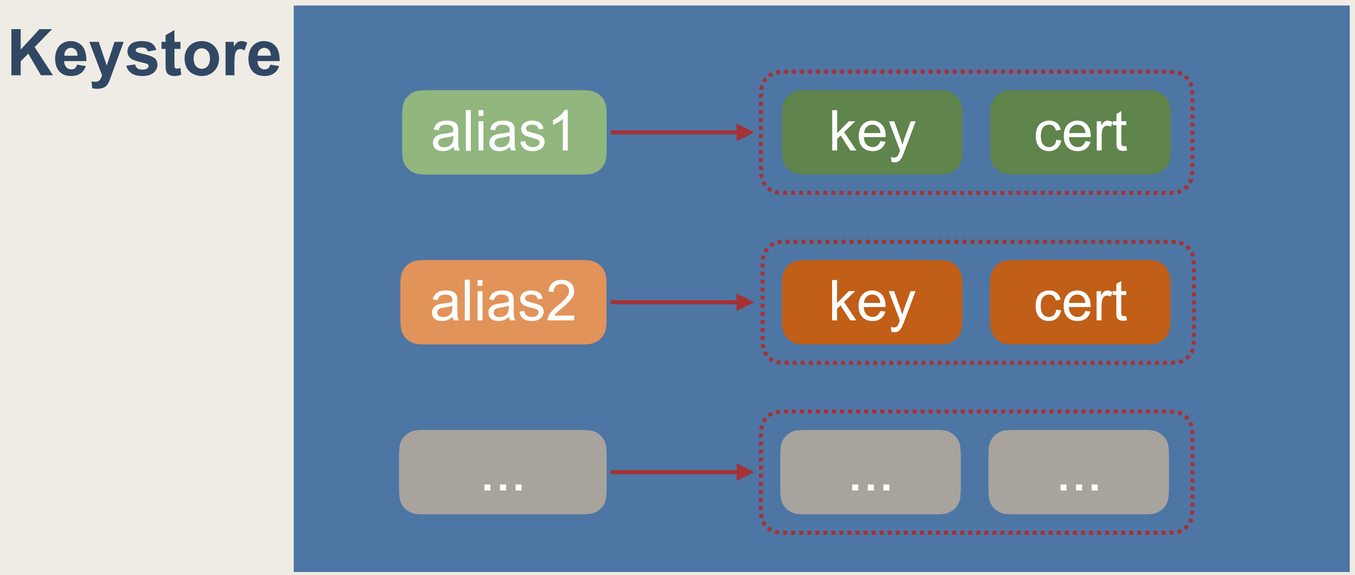
- We have secured Angular frontend
- Now let's secure the Spring Boot backend

Development Process

1. Generate key and self-signed certificate
2. Update application.properties with security configs

Java Keystore

- Java provides support for keys and certificates
- A keystore is a file that contains keys and certificates
- The entries are associated with an alias and password



Step 1: Generate key and self-signed certificate

- Use the JDK utility: `keytool`

```
$ keytool -genkeypair -noprompt \
  -alias luv2code \
  -keypass secret \
  -keystore src/main/resources/luv2code-keystore.p12 \
  -storeType PKCS12 -storepass secret \
  -keyalg RSA -keysize 2048 -validity 365 \
  -dname "C=US, ST=Pennsylvania, L=Philadelphia, O=luv2code, OU=Training Backend, CN=localhost" \
  -ext "SAN=dns:localhost"
```

Location of Java keystore file

Key info: alg, size, valid for 365 days

www.luv2code.com/java-keytool-docs

Step 2: Update application.properties with security configs

File: application.properties

```
# Server web port
server.port=8443

# Enable HTTPS support (only accept HTTPS requests)
server.ssl.enabled=true

# Alias that identifies the key in the key store
server.ssl.key-alias=luv2code

# Keystore location
server.ssl.key-store=classpath:luv2code-keystore.p12

# Keystore password
server.ssl.key-store-password=secret

# Keystore format
server.ssl.key-store-type=PKCS12
```

Based on keytool command in previous slide

```
$ keytool -genkeypair -noprompt \
  -alias luv2code \
  -keypass secret \
  -keystore src/main/resources/luv2code-keystore.p12 \
  -storeType PKCS12 -storepass secret \
  -keyalg RSA -keysize 2048 -validity 365 \
  -dname "C=US, ST=Pennsylvania, L=Philadelphia, O=luv2code, OU=Tra
  -ext "SAN=dns:localhost"
```