

Security

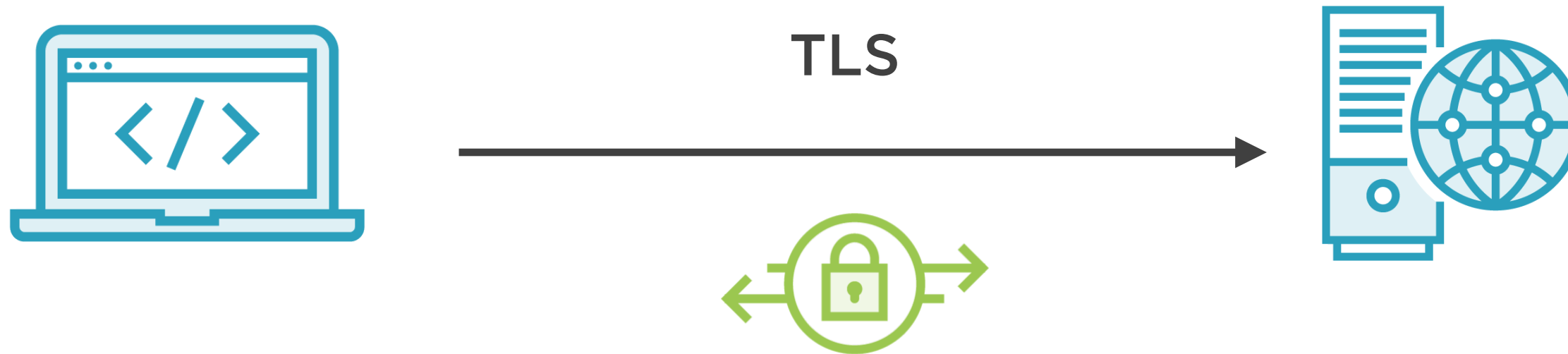


Sander Mak

FELLOW & SOFTWARE ARCHITECT

@Sander_Mak

Secure Connections



Self-signed certificates

Mutual authentication

SSLContext

```
HttpClient.newBuilder()  
    .sslContext(SSLContext.getDefault())  
    .build()
```

- Djavax.net.ssl.trustStore
- Djavax.net.ssl.keyStore

SSLParameters

```
SSLParameters params = new SSLParameters(  
    new String[] { "TLSv1.2" },  
    new String[] { "TLS_AES_128_GCM_SHA256" }  
);
```

```
HttpClient.newBuilder()  
    .sslParameters(params)  
    .build()
```

Self-signed Certificates

Add Certificate Authority to a trust store

```
$JAVA_HOME/bin/keytool
```

```
-Djavax.net.ssl.trustStore /path/to/truststore
```



Create custom SSLContext with trust store

Mutual Authentication

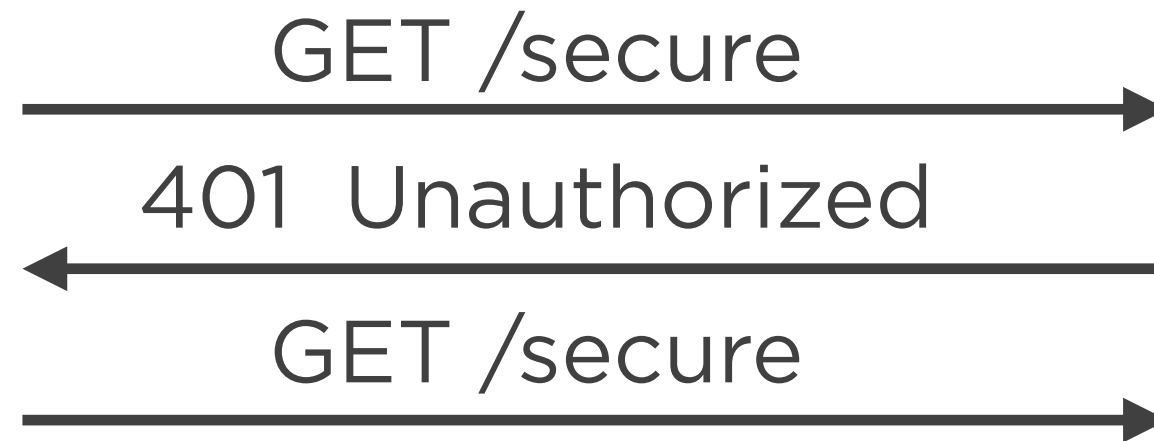
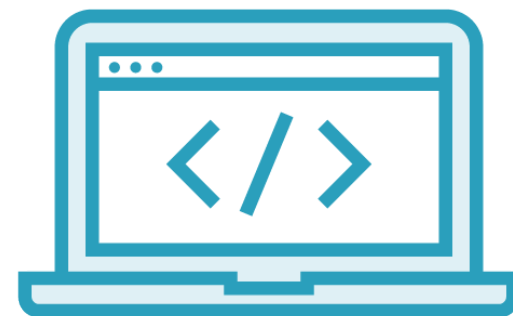
Add client certificate to a key store

```
$JAVA_HOME/bin/keytool  
-Djavax.net.ssl.keyStore /path/to/keystore
```



Create custom SSLContext with key store

HTTP Authentication



Authorization: Basic <credentials>

HTTP Basic Authentication

credentials == base64(user + ":" + password)

Authenticator

```
HttpClient.newBuilder().authenticator(  
    new Authenticator() {  
  
        protected PasswordAuthentication getPasswordAuthentication() {  
            return new PasswordAuthentication("user", "passwd".toCharArray());  
        }  
  
    }  
).build()
```


Proxy

```
HttpClient.newBuilder().proxy(  
    ProxySelector.of(new InetSocketAddress("proxyserver.com", 8080))  
).build()
```

```
ProxySelector.getDefault()
```

Summary



Secure connections



Basic authentication