Server SSL certificates checking while Websocket connection establishing

- Not recommended: Trust all the certificates
- Recommended: Add self-signed certificate to application resources

By default, Android SDK delegates SSL certificates checking to the system level while establishing secure Websocket connestion to a server. On the system level, in its turn, server certificate is compared with system certificate storage content.

In this case, if the server uses self-signed certificate (for debugging purposes), this certificate will not pass the checking. Use the following ways to bypass this depending on Android SDK build.

Not recommended: Trust all the certificates

Since build 1.1.0.18the session option SessionOptions.trustAllCertificates is added, false by default. To accept any certificates including self-signed ones, tis option should be set to true

```
SessionOptions sessionOptions = new SessionOptions(url);
sessionOptions.trustAllCertificates(true);
```

Usage example:

code

```
                    private CheckBox mTrustAllCer;
                    ...
                    mTrustAllCer = (CheckBox) findViewById(R.id.trust_all_certificates_default);
                    ...
                    /**
                     * The options for connection session are set.
                     * WCS server URL is passed when SessionOptions object is created.
                     * SurfaceViewRenderer to be used to display video from the camera is set with method
SessionOptions.setLocalRenderer().
                     * SurfaceViewRenderer to be used to display preview stream video received from the server
is set with method SessionOptions.setRemoteRenderer().
                     */
                    SessionOptions sessionOptions = new SessionOptions(url);
                    sessionOptions.setLocalRenderer(localRender);
                    sessionOptions.setRemoteRenderer(remoteRender);
                    sessionOptions.trustAllCertificates(mTrustAllCer.isChecked());
```

Today, Google Play security requirements does not allow to publish an application with such code. Use the recommended way.

Recommended: Add self-signed certificate to application resources

Since Android SDK build1.1.0.56 X509TrustManager class implementation is removed fromAndroid SDK code. For testing purposes, self-signed certificate must be added to application resources. Also, the configuration file`network_security_config.xml` containing certificate file description must be added:

code

network_security_config.xml

```
<network-security-config>
    <base-config>
        <trust-anchors>
            <certificates src="@raw/my_ca"/>
            <certificates src="system"/>
        </trust-anchors>
    </base-config>
</network-security-config>
```