AUTHENTICATION MANAGER

13 Jan 22

OBJECTIVES

- 1. Time Synchronization
- 2. Checking availability of keystore at local storage & with Authy Server
- 3. Creation of new Self Signed Certificate
- 3. Getting Authy Server's signature on user's Self Signed Certificate
- 4. Creation of Keystore
- 5. Storage of Keystore with password & alias
- 6. Verification of available Keystore
- 7. Keystore Recovery

STEPS

1. Time Synchronization

- a. Connect to http://172.20.82.6:8080/b4server
- b. Get B4 server's date + time
- c. Get user's current system date + time
- d. Calculate time offset
- e. Add offset to user's current time to get updated user's time
- f. Update the same in config file

2. Checking availability of user's keystore at local storage & at Authy Server

- a. At local storage: check for .jks file
- b. With Authy server:
 - i. Get user's email_id from browser
 - ii. Est http connection with b4 server
 - iii. Send user's email to b4 server as http post request
 - iv. Get server's response
 - v. Close http connection

3. Creation new self-signed certificate

- a. Get user form details from browser
- b. Generate Pub & Pvt Keys
- c. Generate x509 certificate with user's data & keyPairs.
- d. Sign the certificate with private key.

GETTING USER INPUT FROM WEB BROWSER: EMAIL ID

```
AM:
String email_id = GC.getEmail ()
                                                            GC:
GC:
getEmail() {
                                                            getEmail() {
// connect to web server
                                                            // receives response from web server
// Pass URL to web server
                                                            // process response and retrieve
                                                            email_id from it
// request server to open URL
                                                            // Returns email_id to AM
}
                                                           Web Server:
Web Server:
                                                           // Process http request
// processes request
                                                           // Retrieves user's email_id from request
// Opens URL page on browser
                                                           // Sends email_id to GC as response
Browser / react
// User enters email in text field
// saves email_id
// prepare http post request with email_id add to it
// Sends http request to web server
```