

Trust Nobody

Dimitry-Janos Lux

```

::ROUTINE enter public
  check = LINEIN(user.txt, 1) --Retrieve stored user password (fully encrypted)

  /*Retrieve user input */
  /* @get(pwfield logininfo) */
  secret = pwfield~text
  .my.app~secret = secret

  /* Use jasypt to authenticate user */
  crypt=bsf.loadClass("org.jasypt.util.password.StrongPasswordEncryptor")~newInstance
  login = crypt~checkPassword(pwfield~text, check)

  IF login = 1 THEN DO
    CALL LINEOUT log.txt, date() || "/" || time() || ":" "User logged in successfully" --write to log
    menuUrl=.bsf~new("java.net.URL", "file:menu.fxml")
    menuNode=bsf.loadClass("javafx.fxml.FXMLLoader")~load(menuUrl)
    menuScene=.bsf~new("javafx.scene.Scene", menuNode)
    .my.app~primaryStage~setScene(menuScene)
  END

  ELSE DO
    logininfo~text="Password entered is not correct"
    CALL LINEOUT log.txt, date() || "/" || time() || ":" "Wrong password was entered"
  END
END

```

Script Engine!

- Jasypt → Einfach

- SHA-256 Hash

```

user.txt
1 9791LYwBy33vnT0C0vgt1HN3RrmJnaPxe7RuvYEyuDCFiax9SQZeQsA6XeyfsabI
2

```

```

::ROUTINE general public
  /* Access the general section */
  /* Get access to Script Context */
  /* @get(mainText) */

  slotDir=arg(arg())
  sc=slotDir~scriptContext
  sc~setAttribute("doc", "general", sc~global_scope)

  cipher =LINEIN(general.txt)

  IF cipher <> "" THEN DO
    crypt=bsf.loadClass("org.jasypt.util.text.BasicTextEncryptor")~newInstance
    crypt~setpassword(.my.app~secret)
    plaintext=crypt~decrypt(cipher)
    mainText~text=plaintext
  END

  IF cipher = "" THEN DO
    mainText~text=""
    CALL LINEOUT log.txt, date() || "/" || time() || ":" "General section accessed" --write to log
  END

```

•PBEWithMD5AndDES

•(Salt)

```

:: ROUTINE save public
  /* Save changes */
  /* @get(doc mainText) */

  crypt=bsf.loadClass("org.jasypt.util.text.BasicTextEncryptor")~newInstance
  crypt~setpassword(.my.app~secret)
  plaintext=mainText~text
  cipher=crypt~encrypt(plaintext)

  IF doc = "general" THEN DO
    CALL LINEOUT general.txt, cipher, 1
    CALL LINEOUT log.txt, date() || "/" || time() || ":" "General section edited" --write to log
  END

  IF doc = "ideas" THEN DO
    CALL LINEOUT ideas.txt, cipher, 1
    CALL LINEOUT log.txt, date() || "/" || time() || ":" "Ideas section edited" --write to log
  END

```

General

The Top 10 Java Packages of 2019

Ideas

- #1 – org.apache.hadoop
- #2 – org.apache.camel
- #3 – org.apache.commons
- #4 – org.apache.ignite
- #5 – org.apache.http
- #6 – org.springframework.boot
- #7 – com.amazonaws.services
- #8 – com.google.common
- #9 – org.apereo.cas
- #10 – com.fasterxml.jackson

Save

Back

```
::ROUTINE create public
/* @get(newPw newPwConf pwInfo) */

crypt=bsf.loadClass("org.jasypt.util.password.StrongPasswordEncryptor")~newInstance

c = newPw~text = newPwConf~text

IF c = 0 THEN DO
    pwInfo~text= "Not matching!"
    IF c = 0 THEN CALL LINEOUT log.txt, date() || "/" || time() || ":" "Reset failed, 2 different passwords entered"
END

IF c = 1 THEN DO
    CALL LINEOUT log.txt, date() || "/" || time() || ":" "Reset has been successful" --write to log
    pw = crypt~encryptPassword(newPw~text)
    CALL LINEOUT user.txt, pw, 1
    CALL LINEOUT general.txt, "", 1
    CALL LINEOUT ideas.txt, "", 1
    CALL back
END
```

```
CALL LINEOUT log.txt, date() || "/" || time() || ":" "User logged in successfully" --write to log
```

Trust Nobody

```
23 Jun 2020/18:01:44: User accessed his private notes
23 Jun 2020/18:01:51: General section edited
23 Jun 2020/18:02:09: App was closed
23 Jun 2020/19:18:06: User logged in successfully
23 Jun 2020/19:55:28: User logged in successfully
23 Jun 2020/19:55:30: User accessed his private notes
23 Jun 2020/19:55:37: General section edited
23 Jun 2020/19:55:40: Ideas section edited
23 Jun 2020/19:55:59: Reset has been successful
23 Jun 2020/19:56:00: User logged in successfully
23 Jun 2020/19:56:01: User accessed his private notes
23 Jun 2020/19:56:07: General section edited
23 Jun 2020/19:56:07: Ideas section accessed
23 Jun 2020/19:56:11: Ideas section edited
23 Jun 2020/19:56:16: User logged in successfully
23 Jun 2020/19:56:16: User accessed his private notes
23 Jun 2020/19:56:25: App was closed
23 Jun 2020/19:57:42: User logged in successfully
23 Jun 2020/19:57:43: User accessed his private notes
23 Jun 2020/19:58:26: User logged in successfully
23 Jun 2020/19:58:26: User accessed his private notes
23 Jun 2020/19:58:39: General section edited
23 Jun 2020/20:00:02: User logged in successfully
23 Jun 2020/20:00:03: User accessed his private notes
23 Jun 2020/20:00:15: General section edited
23 Jun 2020/20:00:48: User logged in successfully
23 Jun 2020/20:00:49: User accessed his private notes
23 Jun 2020/20:01:32: User logged in successfully
23 Jun 2020/20:01:33: User accessed his private notes
23 Jun 2020/20:01:57: User logged in successfully
23 Jun 2020/20:01:58: User accessed his private notes
23 Jun 2020/20:02:07: Wrong password was entered
23 Jun 2020/20:02:42: Wrong password was entered
```

Design

```
xDesktop      = UNO.createDesktop()
xComponentLoader = xDesktop~XComponentLoader
url = ConvertToURL(directory() || "\general.odt")
writerComponent = xComponentLoader~loadComponentFromURL(url, "_blank", 0, .UNO~noProps)
xText = writerComponent~XTextDocument~getText
text =xText~getString
```

- Performance
 - Version OpenOffice / Version .txt
 - Verschlüssen: Dateien oder Strings?
- Usability
 - Direkt im Programm arbeiten
- Fonts
- App Icon

```
/* Adding the app icon. */
img=.bsf~new("javafx.scene.image.Image", "file:icon.png")
primaryStage~getIcons~add(img)
```

Verbesserung

- Seltener Bug → .txt file → neue Zeile
- Automatisch speichern nach jedem Input
- Anzahl Dateien und Controller → 17 Files
 - Duplikate → Back Button
- Best Practice → SAY/say, FXML kommentieren?