MATURITNÍ PRÁCE

Aplikace pro generování a ověřování jednorázových hesel

Uživatelská příručka

Petr Michalík

Gymnázium, Praha 6, Nad Alejí 1952

2016/2017

Prohlášení

Prohlašuji, že jsem na maturitním projektu pracoval samostatně pouze za pomoci uvedených zdrojů.

V Praze 11. dubna 2017

Podpis

Úvod

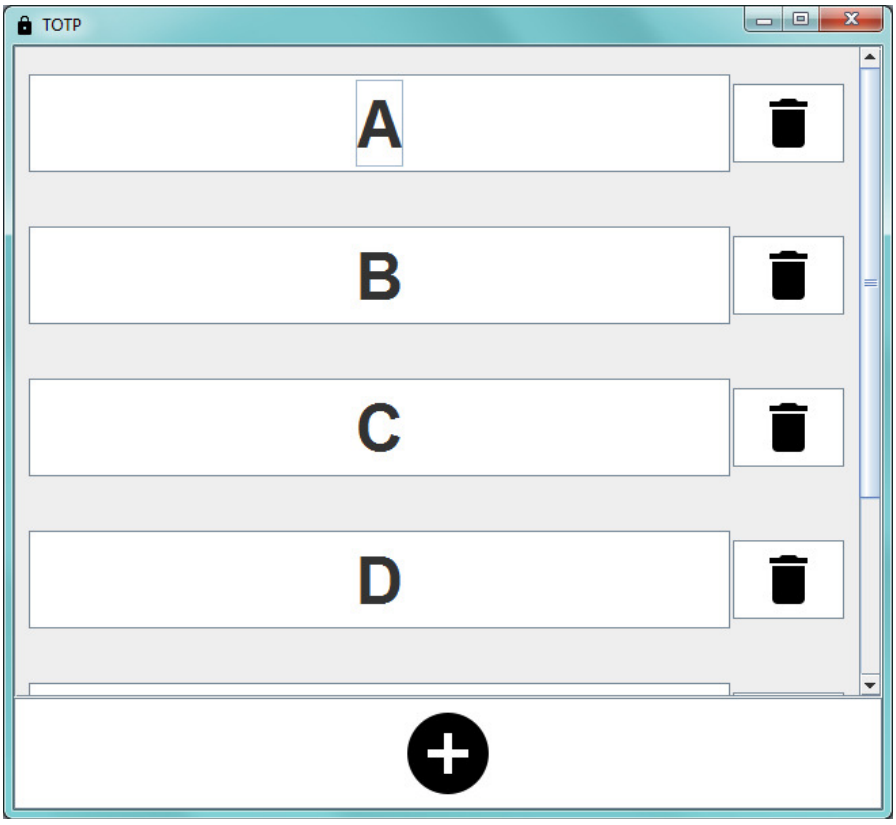
Tento maturitní projekt implementuje algoritmus pro generování jednorázových hesel (TOTP) podle standardu RFC 6238 v programovacím jazyce Java. Součástí projektu je uživatelská aplikace, která na základě sdíleného hesla generuje jednorázová hesla, a server implementující REST API pro ověřování jednorázových hesel a správu účtů. Poslední součástí je ukázkový server v PHP, který využívá REST serveru pro správu uživatelů a demonstruje použití dvoufázového ověřování v praxi.

Stažení

Aplikaci lze stáhnout na GitHubu (https://github.com/PetrM97/totp) v záložce Releases. Ke spuštění programu je potřeba mít nainstalován Java JRE verze 7 a vyšší. *Druhou **pokročilejší** možností je zkompilovat zdrojové kódy a vytvořit si vlastní sestavení (build).*

Aplikaci není potřeba nijak instalovat, stačí ji pouze spustit jako běžný program.

Klient



Vzhled programu se může lišit v závislosti na použitém operačním systému.

Přidání nového záznamu

- Klikněte na tlačítko `+` v dolní části okna programu
- V dialogu vyplňte název a vložte sdílené heslo. Záznam uložte stisknutím klávesy `Enter` nebo zavřením dialogu.
- V programu by se nyní měl objevit nově vytvořený záznam

Vygenerování jednorázového hesla

- Jednorázové heslo vygenerujete kliknutím na požadovaný záznam
- Heslo se zkopíruje do systémové schránky
- Pro vložení hesla stiskněte `Ctrl+V`

Server

Pro spuštění serveru je potřeba otevřít soubor `TOTP_Server.jar` v příkazové řádce. Pokud jste v kořenové složce repozitáře, stačí zadat

```
java -jar build/jar/TOTP_Server.jar
```

Server se automaticky spouští na adrese localhost:8080 a využívá REST rozhraní pro správu záznamů.

Struktura

- /users
  - GET = vypíše počet uživatelů (záznamů)
  - POST = přidá nového uživatele, data v POSTU je uživatelské jméno a server vrací sdílené heslo pro uživatele
  - DELETE = vymaže všechny záznamy
- /users/[username]
  - GET = vypíše informace o uživateli
  - POST = ověří poslané jednorázové heslo, odpovědí je správnost hesla
  - PUT = vygeneruje nové sdílené heslo, které server vrátí
  - DELETE = vymaže uživatele

Příklady

Jelikož tento server odpovídá na běžné HTTP požadavky, lze použít např. cURL pro komunikaci.

Vytvoření nového uživatele 'uzivatel':

```
curl localhost:8080/users --data uzivatel
```

Ověření platnosti jednorázového hesla '123456' pro uživatele 'uzivatel':

```
curl localhost:8080/users/uzivatel --data 123456
```

Smazání uživatele 'uzivatel':

```
curl localhost:8080/users/uzivatel -X DELETE
```

Demo

Demonstrační aplikace využívá PHP a php-curl modulu pro komunikaci s REST serverem. Dodatečně je pak využívána knihovna clipboard.js a Google Chart API pro vykreslení QR kódů pro Google Authenticator. Také doporučuji použít Apache místo integrovaného PHP serveru.

Instalace

Před instalací demo serveru je potřeba mít v lokálním adresáři kopii celého repozitáře (podle této stránky). Také je potřeba mít spuštěný REST server kvůli funkčnosti ukázkové aplikace.

Pro spuštění serveru je potřeba mít nainstalován PHP a php-curl. Instalaci si můžete vyzkoušet na stránce Termbox, kde lze na 3 hodiny využít zdarma virtuální linuxový terminál s jedním veřejným HTTP portem.

V Ubuntu lze vše nainstalovat zadáním

```
sudo apt-get install php php-curl apache2 libapache2-mod-php
```

Tím se nainstaluje nejnovější PHP, php-curl i Apache. Ve Windows je potřeba konfigurační soubor `php.ini` (řádek 878) nastavit tak, aby PHP používalo `php_curl.dll` knihovnu.

Poté případně upravte `/etc/apache2/ports.conf` a `/etc/apache2/sites-available/000-default.conf` a restartujte Apache

pomocí příkazu `apache2ctl restart`. Pokud používáte Termbox, jako port zadejte číslo portu uvedené v proměnné $PORT (pravděpodobně 2000) v obou konfiguračních souborech.

Ujistěte se také, že ve složce `demo` lze číst, vytvářet i spouštět soubory a případně upravte pomocí `chmod`.

Poté stačí v prohlížeči otevřít danou adresu (u Termbox uvedená dole), kde by se měl zobrazit přihlašovací dialog.

Kompilace

Pro kompilaci je potřeba mít nainstalován Java JDK a Apache Ant. Při stahování balíků se automaticky spustí Apache Ivy, který je případně stažen pomocí Antu.

V Ubuntu pro instalaci všech potřebných programů zadejte:

```
sudo apt-get install default-jdk ant
```

Tím si stáhnete všechny potřebné programy.

Nejdříve naklonujte repozitář do svého lokálního adresáře pomocí

```
git clone https://github.com/petrm97/totp
```

nebo celý repozitář stáhněte z hlavního stránky jako `.zip` soubor. Repozitář obsahuje submodul Material Design Icons, který je volitelný a je využíván klientskou aplikací. Pro stažení submodulu použijte

```
git submodule update --init
```

Pro vytvoření obou JAR souborů zadejte

```
ant jar
```

Poté se ve složce `/build/jar` objeví spustitelné JAR soubory.

Pro vytvoření `.class` souborů stačí zadat

```
ant compile
```

Pro zobrazení všech možností zadejte

```
ant -p
```

Zdroje a knihovny

- RFC 6238 (TOTP standard), `https://tools.ietf.org/html/rfc6238`
- RFC 4226 (HOTP standard), `https://tools.ietf.org/html/rfc4226`
- jUnit, `http://junit.org/`
- Hamcrest, `http://hamcrest.org/`
- Apache Commons Codec, `https://commons.apache.org/proper/commons-codec/`
- Restlet, `https://restlet.com/open-source/`
- JSON-simple, `https://code.google.com/archive/p/json-simple/`