Projekt niskiego poziomu

Opis klas i metod odpowiedzialnych za bezpieczeństwo w aplikacji

AuthManager

Funkcjonalne wymagania bezpieczeństwa:

- FAU_GEN
- FAU_ARP
- FAU_SAR
- FAU_STG

Klasa odpowiedzialna jest za kontrolę dostępu użytkownika, podejmuje próby logowania, rejestracji i zwraca wynik. Odpowiedzialna za pobranie danych nowego użytkownika (login, hasło) i weryfikuje je. Implementuje akcje związane z użytkownikami, łącznie z akcjami CRUD.

Metoda	Opis metody
putUser	Dostęp tylko dla niezalogowanego użytkownika. Pobiera login, hasło, potwierdzenie hasła następnie tworzy nowego użytkownika. Gdy login jest zajęty zwraca błąd 409. Gdy chcemy utworzyć admina podajemy również zmienną środowiskową ADMIN_SECRET.
postUser	Dostęp tylko dla niezalogowanego użytkownika. Pobiera login, hasło, a nastepnie loguje użytkownika.
deleteUser	Dostęp tylko dla zalogowanego użytkownika. Usuwa użytkownika
logoutUser	Dostęp tylko dla zalogowanego użytkownika. Wylogowuje użytkownika

DocumentManager

Klasa odpowiedzialna jest za zarządzanie dokumentami użytkownika. Podejmuje akcje związane z dodaniem nowego dokumentu dla użytkownika.

Funkcjonalne wymagania bezpieczeństwa:

- FDP_ACC
- FDP_ACF

Metoda	Opis metody
putDocument	Dostęp tylko dla zalogowanego użytkownika. Dodaje nowy dokument dla wskazanego użytkownika. Tworzy id danego dokumentu
getDocumentID	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu lub użytkownik któremu został przydzielony dostęp. Przesyła dokument, jeśli użytkownik jest właścicielem otrzymuje id wszystkich dostępów do dokumentu. Jeśli użytkownikowi jest tylko udostępniony dokument widzis swój dostęp oraz właściciela dokumentu.

Metoda	Opis metody
getRoleDocument	pozwala na pobranie listy użytkowników uprawnionych do korzystania z pliku.
EncodeKey	pozwala na zaszyfrowanie klucza do pliku.
DecodeKey	pozwalaa na odszyfrowanie klucza do pliku w celu skorzystania z udostępnionych zasobów.

ShareManager

Klasa odpowiedzialna jest za udostępnianie dokumentów pośród użytkowników. Podejmuje akcje związane z przypisaniem danego dokumentu dla użytkownika.

Funkcjonalne wymagania bezpieczeństwa:

- FIA_UAU
- FIA_UID
- FIA_AFL

Metoda	Opis metody
putShare	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Tworzy obiekt udostępnienia dokumentu
getShare	Dostęp tylko dla zalogowanego użytkownika, zwraca listę obiektów udostępnionych dla danego użytkownika
deleteShare	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu lub użytkownika któremu został przyznany dostęp. Jeśli wykonuje ją właściciel - usuwa obiekt udostępniania. Jeśli wykonuje ją użytkownik, który ma do niej dostęp - udostępnienie dokumentu zostaje odrzucone (state ustawione na -1), właściciel widzi że partner go odrzucił.
getShareID	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu lub użytkownika któremu został przyznany dostęp. Pobiera udostępniony obiekt.
postShareID	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu lub użytkownika któremu został przyznany dostęp.Przekierowuje request pod odpowiedni post dla state share'a. Requesty zależne od state moga byc ustawione tylko raz dla udostepnienia. Nie mozna zmeniac ustawionych im wartosci w pozniejszym procesie (ale mozna zaczac udostepniac dokument od nowa). Uzytkownik moze udostepnic drugiemu użytkownikowi dokument parę razy - aby miec do niego wiele hasel.
postID0	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Przyjmuje klucz publiczny dla udostepnienia od uzytkownika ktory udostepnia dokument. Zmienia state na 1
postID1	Dostęp tylko dla zalogowanego użytkownika, użytkownika któremu został przyznany dostęp. Przyjmuje klucz publiczny dla udostepnienia od uzytkownika ktoremu udostepnia dokument. Zmienia state na 2

Metoda	Opis metody
postID2	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Sprawdza poprawnosc ponownie wyslanego klucza publiczegnego od uzytkownika udostepniajcego dokument. Jesli zgadza sie z poprzednio wyslanym ustaia crypted jako zakodowane haslo do dokumentu. Zmienia state na 3.
postID3	Dostęp tylko dla zalogowanego użytkownika, użytkownika któremu został przyznany dostęp. Sprawdza poprawnosc klucza uzytkownika ktoremu udostepniamy dokument. jesli sie zgadza wysyla crypted oraz dostep do dokumentu.

PermissionManager

Klasa odpowiedzialna jest za ustawianie praw dostępu dla użytkowników. Podejmuje akcje związane z przypisaniem roli dostępu danego użytkownika do dokumentu.

Funkcjonalne wymagania bezpieczeństwa:

- FMT_MSA
- FMT_SMR
- FPT_STM

Metoda	Opis metody
deletePerm	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Usuwa dostep po id oraz obiekt share jesli premission dotyczy udostepnienia. Jesli dostep jest dostepem wlasciciela usunie on wszystkie dostepy do dokumentu.
getPerm	Dostęp tylko dla zalogowanego użytkownika, Zwraca wszystkie dostępy dla użytkownika
getPermID	Dostęp tylko dla zalogowanego użytkownika, Zwraca obiekt dostępu dla użytkownika

TimerManager

Klasa odpowiedzialna jest za ustawianie czasu przechowywania dokumentu w serwisie.

Funkcjonalne wymagania bezpieczeństwa:

- FPT_STM

Metoda	Opis metody
putTimer	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Ustawia timer który po aktywacji usunie dostęp dla użytkownika. Zwraca 409 jeśli timer jest już ustawiony dla tego dostępu
deleteTimer	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Ustawia timer.
getTimerID	Dostęp tylko dla zalogowanego użytkownika, właściciela dokumentu. Zwraca obiekt timera.

UserManager

Klasa odpowiedzialna jest za obsługę użytkowników.

Funkcjonalne wymagania bezpieczeństwa:

- FIA_AFL
- FIA_UID
- FDP_IFF

Metoda	Opis metody
getUser	Dostęp tylko dla zalogowanego użytkownika. Zwraca informacje o użytkowniku.
postUser	Dostęp tylko dla zalogowanego użytkownika. Użytkownik musi być adminem. Zmienia danu użytkownika login lub hasło na własne. Może zmienić login i hasło jednocześnie. Zwraca 409 jeśli nowy login jest zajęty.
deleteUser	Brak dostępu, usuwa użytkownika.
getUserID	Dostęp tylko dla zalogowanego użytkownika. Zwraca informacje o użytkowniku o podanym ID.

DatabaseManager

Klasa odpowiedzialna jest za zarządzanie bazą danych

Funkcjonalne wymagania bezpieczeństwa:

- FCS_CKM
- FCS_COP

Metoda	Opis metody
getDB	Brak dostępu, wysyła aplikacje do zarządzania bazą danych

LogsManager

Klasa odpowiedzialna jest za zarządzanie logami.

Funkcjonalne wymagania bezpieczeństwa:

- FMT_MSA
- FMT_SMR

Metoda	Opis metody
getLogs	Brak dostępu, wysyla logi serwera w postaci pliku html, zmienna srodowiskowa USERS_CAN_READ_LOGS moze zapewnic serwowanie dla wszystkich

Architektura bazodanowa

Documents

Typ pola	Nazwa	Dodatkowe informacje
String	Name	wymagane
Base-64	Content	wymagane
Reference	Permissions	PERMISSIONS_DATABASE

Permissions

Typ pola	Nazwa	Dodatkowe informacje
Reference	UserID	USERS_DATABASE
Reference	DocumentID	DOCUMENTS_DATABASE
Reference	ShareID	SHARE_DATABASE
String	Type	enum(o,r), wymagane
Reference	Timer	TIMERS_DATABASE

Shares

Typ pola	Nazwa	Dodatkowe informacje
Reference	DocumentID	DOCUMENTS_DATABASE
Reference	PermissionID	PERMISSIONS_DATABASE
String	Prime	wymagane
String	Generator	wymagane
String	Crypted	
Number	State	Default: 0

Origin User

Typ pola	Nazwa	Dodatkowe informacje
String	PublicKey	
Reference	Id	USERS_DATABASE

Destination User

Typ pola	Nazwa	Dodatkowe informacje
String	PublicKey	
Reference	Id	USERS_DATABASE

Timers

Typ pola	Nazwa	Dodatkowe informacje
String	Type	wymagane
Object	Params	
Date	When	wymagane
String	ObjectModelName	wymagane

Users

Typ pola	Nazwa	Dodatkowe informacje
String	Login	wymagane, ilość znaków (3-20)
HASH(bcrypt)	Password	wymagane
Boolean	IsAdmin	Default: false
Reference	Permissions	PERMISSIONS_DATABASE
Reference	Shares	SHARES_DATABASE