

de.htwberlin.kba.gr7.vocabduel.user_administration.export

Interface AuthService

public interface **AuthService**

Service for managing authentication, i.e. user logins/registrations, checking access rights using auth tokens and token management.

Version:

1.0, May 2021

Author:

Sebastian Kehl, Lucas Larisch

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type		Method and Description
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User		fetchUser (java.lang.String token) Verifies a given auth token, decodes it and returns the user it belongs to.
boolean		hasAccessRights (java.lang.String token) Checks whether a given auth token is valid, i.e. if the user has access rights or not.
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.LoggedInUser		loginUser (java.lang.String email, java.lang.String password) Authenticates an existing user in order to take actions in this application.
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.AuthTokens		refreshAuthTokens (java.lang.String refreshToken) Updates both refresh and auth token and returns the two tokens if the given refresh token is valid.
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.LoggedInUser		registerUser (java.lang.String username, java.lang.String email, java.lang.String firstname, java.lang.String lastname, java.lang.String password, java.lang.String confirmPassword) Registers a new user who can take actions in this application.
int		updateUserPassword (de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User user, java.lang.String currentPassword, java.lang.String newPassword, java.lang.String confirmPassword) Updates a user's password if the current password is correct and the new password is both valid and matching the confirm password.

Method Detail

registerUser	
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.LoggedInUser	registerUser(java.lang.String username, java.lang.String email, java.lang.String firstname, java.lang.String lastname,

```
        java.lang.String password,
        java.lang.String confirmPassword)
throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PasswordsDoNotMatchException,
       de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PwTooWeakException,
       de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidOrRegisteredMailException,
       de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.AlreadyRegisteredUsernameException,
       de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.IncompleteUserDataException,
       javax.naming.InvalidNameException,
       de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException
```

Registers a new user who can take actions in this application. After having successfully stored the new user data, a `LoggedInUser` object is returned which is used for authentication using tokens without revealing the user's password. (see `loginUser(java.lang.String, java.lang.String)`).

Parameters:

- username - Username to be set.
- email - Email to be set.
- firstname - First name of the new user.
- lastname - Family name of the new user.
- password - Password for the new user, will be stored in a different way than the other user data
- confirmPassword - Repeated password for confirmation purposes (must match with password).

Returns:

The object of the new user. Password will never be revealed, instead, this object contains auth tokens.

Throws:

- `de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PasswordsDoNotMatchException` - password and confirmPassword do not match.
- `de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PwTooWeakException` - The given password is not strong enough.
- `de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidOrRegisteredMailException` - The user's specified email is either already in use or invalid and, thus, cannot be used again.
- `de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.AlreadyRegisteredUsernameException` - The user's specified username is already in use and, thus, cannot be used again.
- `de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.IncompleteUserDataException` - The passed user object does not contain all required user data.
- `javax.naming.InvalidNameException` - The passed user user does have an invalid first/ last name.
- `de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException` - An `OptimisticLock` error in the user module occurred.

loginUser

```
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.LoggedInUser loginUser(java.lang.String email,
                                             java.lang.String password)
throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException
```

Authenticates an existing user in order to take actions in this application. The returned user object contains all relevant user data including auth tokens without revealing the user's password. If the login fails, `null` is returned instead of an object.

Parameters:

- email - Unique email address the user signs in with.
- password - The user's password.

Returns:

User object including authentication tokens / `null` if the login was not successful.

Throws:
de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException - An OptimisticLock error in the user module occurred.

fetchUser

```
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User fetchUser(java.lang.String token)
                                throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException
```

Verifies a given auth token, decodes it and returns the user it belongs to. If no or no valid token is given, null is returned.

Parameters:
token - Auth token the user is to be determined by.

Returns:
User the given token belongs to or null in case of no/no valid token.

Throws:
de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException - An OptimisticLock error in the user module occurred.

refreshAuthTokens

```
de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.AuthTokens refreshAuthTokens(java.lang.String refreshToken)
                                throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException
```

Updates both refresh and auth token and returns the two tokens if the given refresh token is valid. Otherwise, null is returned. Note: After having called this method successfully, the refreshToken passed in as a param is obsolete and instead, the new refresh token (AuthTokens.getRefreshToken()) is to be used.

Parameters:
refreshToken - Token to be used for refreshing auth/refresh token.

Returns:
Object containing refreshed tokens, null if the operation was not successful.

Throws:
de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException - An OptimisticLock error in the user module occurred.

hasAccessRights

```
boolean hasAccessRights(java.lang.String token)
            throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException
```

Checks whether a given auth token is valid, i.e. if the user has access rights or not.

Parameters:
token - Auth token to be checked.

Returns:
true if the token is valid, false if not/null.

Throws:
de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException - An OptimisticLock error in the user module occurred.

updateUserPassword

```
int updateUserPassword(de.htwberlin.kba.gr7.vocabduel.user_administration.export.model.User user,
    java.lang.String currentPassword,
    java.lang.String newPassword,
    java.lang.String confirmPassword)
throws de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidFirstPwdException,
    de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PasswordsDoNotMatchException,
    de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PwTooWeakException,
    de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidUserException,
    de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException
```

Updates a user's password if the current password is correct and the new password is both valid and matching the confirm password.

Parameters:

- user - User whose password is to be changed.
- currentPassword - The user's current password for authentication purposes.
- newPassword - The user's new password - will be stored in a different way than the other user data.
- confirmPassword - Repeated password for confirmation purposes (must match with password).

Returns:

int database status of the update query.

Throws:

- de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidFirstPwdException - currentPassword is wrong.
- de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PasswordsDoNotMatchException - newPassword and confirmPassword do not match.
- de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.PwTooWeakException - The given password is not strong enough.
- de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.InvalidUserException - The given user is null or does not exist.
- de.htwberlin.kba.gr7.vocabduel.user_administration.export.exceptions.UserOptimisticLockException - An OptimisticLock error in the user module occurred.