Contents

[Files: 2](#_Toc436306596)

[InputRecheck.php 2](#_Toc436306597)

[Relog.php 3](#_Toc436306598)

[Login.php 4](#_Toc436306599)

[UserHandler.php 5](#_Toc436306600)

[Database.php 7](#_Toc436306601)

**API**

## Files:

1. InputRecheck.php
2. Relog.php
3. Login.php
4. UserHandler.php
5. Database.php

### InputRecheck.php

Author: Tomas Paronai

#### Includes:

* Database.php

Class: Recheck

#### Class params:

1. Private
   * handlerDB
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * CheckEmail(email,size)
   * CheckInput(input,size)
   * dumpsSpecialChars(input)

**CheckEmail(email,size)**

Checks the email if its out of bounds and if the email is already in the database.

**@author** Tomas Paronai

**@param** $email - email we want to check

**@param** $size - the max number of chars the string can have

**@return** "Input out of bounds." - long input

**@return** "Email already used." - same email found in the database

**@return** "Invalid email." - invalid email format

**@return** true - all OK

**CheckInput(input,size)**

Checks if the input string isnt out fo bounds.

**@author** Tomas Paronai

**@param** $input - the string we want to check

**@param** $size - the max number of chars the string can have

**@return** "Input out of bounds." - long input

**@return** true - all OK

**dumpSpecialChars(input)**

Deletes all special characters from the string.

**@author** Matus Kacmar

**@param** $input - the string we want to clean

**@return** $input - clean string

### Relog.php

Author: Tomas Paronai

Relog handles registration, login and user profile edit.

#### Includes:

* Database.php
* InputRecheck.php
* UserHandler.php
* Login.php

Class: -

#### Functions:

1. Private
   * -
2. Public
   * errorControl(name,surname,email,password)

**errorControl(name,surname,email,password)**

Validates user inputs for registration form.

**@author** Tomas Paronai

**@param** $name

**@param** $surname

**@param** $email

**@param** $password

**@return** - true if all OK

### Login.php

Author: Tomas Paronai

#### Includes:

* Database.php

Class: Login

#### Class params:

1. Private
   * handlerDB
   * name
   * id
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * checkLogin(email,password)
   * getName()
   * getUserId()

**checkLogin(email,password)**

Compares the email and password if they match the data in database.

**@author** Tomas Paronai

**@param** $email - login email

**@param** $password - login password

**@return** false - email or password not matching

**@return** true - login correct

**getName()**

**@author** Tomas Paronai

**@return** $this->name - the name of the logged in user.

**getUserId()**

**@author** Tomas Paronai

**@return** $this->id - the id of the logged in user.

### UserHandler.php

Author: Tomas Paronai

#### Includes:

* Database.php

Class: User

#### Class params:

1. Private
   * handlerDB
   * success
   * id
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * editUser(id)
   * newUser(name,surname,email,password)
   * saveFirstData(parameter,value)
   * saveData(parameter,value)
   * setId(id)
   * getId()
   * getValidId()
   * getData(parameter)
   * isSaved()

**editUser(id)**

Creates an UserHandler instance with only the given ID

**@author** Tomas Paronai

**@param** $id

**newUser(name,surname,email,password)**

Creates an UserHandler instance with only the given params

**@author** Tomas Paronai

**@param** $name

**@param** $surname

**@param** $email

**@param** $password

**saveFirstData(parameter,value)**

Saves the first data in the table.

**@author** Tomas Paronai

**@param** $parameter - name of the colum

**@param** $value

**saveData(parameter,value)**

Edits user info by the given id | saves the user input in the database.

**@author** Tomas Paronai

**@param** $parameter - name of the colum

**@param** $value

**@param** $id - given id

**setId(id)**

**@author** Tomas Paronai

**@param** $id

**getId()**

**@author** Tomas Paronai

**@return** $id

**getValidId()**

**@author** Tomas Paronai

**@return** - id of the user which is being handled

**getData(parameter)**

Returns the data from database of the current which is being handled.

**@author** Tomas Paronai

**@param** $parameter - the data i want

**@return** $users[$i][$parameter] - data at parameter

**isSaved()**

**@author** Tomas Paronai

**@return** "User registered." - user was successfuly saved in the database

**@return** "Registration failed." - saving failed. Probably database error.

### Database.php

Author: Matus Kacmar

Class: DBHandler

…autor doplnit