

Підпрограми calls.pm

Глобальні змінні

- `$ses::ajax` - встановлений, якщо запит є ajax.
- `$ses::cookie` - поточні куки, наприклад `$ses::cookie→{debug}`.
- `$ses::http_prefix` - 'http://' або 'https://'.
- `$cfg::img_url` - '<https://nodeny.com.ua/i>'.
- `$ses::ip` - ip адміна/клієнта, який залогінився в веб-інтерфейсі.
- `$ses::server` - ім'я сервера витягнутого з адресного рядка, наприклад, `nodeny.com.ua` в <https://nodeny.com.ua/cgi-bin/stat.pl>.
- `$ ses:: t` - поточний час, unix timestamp.
- `$ses::day_now` - поточний день.
- `$ses::mon_now` - поточний місяць.
- `$ses::year_now` - поточний рік.
- `$ses::auth` - посилання на дані поточної авторизації в адмінці:
 - `$ses::auth→{auth}` встановлений, якщо адмін/клієнт авторизований.
 - `$ses::auth→{uid}` - id адміна/клієнта
 - `$ses::auth→{role}`: 'admin' - авторизований адмін, 'user' - клієнт
 - `$ses::auth→{ses}` - рядок-сесія
 - `$ses::auth→{trust}` - чи довірена сесія
- `$ses::unikey` - ключ до даних у таблиці `webses_data`, які будуть додані до поточних даних, переданих браузером.

У будь-який модуль, крім авторизації, можна потрапити лише будучи авторизованим. Авторизація буває довіреною та недовіреною. Якщо джерело авторизації надійне, наприклад, авторизація за логіном та паролем, він встановлює параметр `trust` в 1. Небезпечні функції слід дозволяти тільки при довіреному з'єднанні:

```
$ses::auth->{trust} or Error('З'єднання не довірене. Необхідно перелогінитися');
```

До клієнтської статистики може потрапити не тільки клієнт, а й адміністратор від його імені, тому в деяких випадках бажано виводити різну інформацію для адміна та клієнта, а також фіксувати, що саме адміністратор виконав дію:

```
if( 1 + 1 > 3 )
{
    Error( Adm->id? 'Паніка! Perl неправильно вважає' : 'Сервіс тимчасово недоступний');
}
# Поповнимо рахунок клієнта, при цьому Pay_to_DB сама проставит автора
Pay_to_DB(uid=>$uid, cash=>10, category=>1);
```

POST/GET параметри

Отримати параметри, надіслані через браузер, можна за допомогою `ses::input`:

<http://xxx?a=test&b=0&uid=33>

```
if( ses::input('a') eq 'test' && ses::input_exists('b') )
{
    # Умова спрацює т.к. в url a = test і параметр b присутній
    # Параметр uid про всяк випадок примусово приведемо до цілого числа
    my $uid = ses::input_int('uid');
    $uid or Error('Введіть ціле число, що не дорівнює нулю');
}
```

Хеш усіх переданих параметрів зберігається в `$ses::input_orig`:

```
# Якщо параметр yes не існує або не встановлено,
# відредагуємо самі на себе і пошлемо всі параметри + встановимо yes
ses::input('yes') or url->redirect( { %$ses::input_orig, yes=>1 } );
```

Існує альтернативний спосіб передачі параметрів - дані записуються в базу даних таблицю [webses_data](#), доступ до яких здійснюється за ключом `unikey` таблиці.

Взагалі через дану таблицю можуть передаватися будь-які дані, а передача параметрів від браузера - одна з фіч. На прикладі покажемо

навіщо потрібний такий механізм.

Припустимо, клієнт надсилає дані. Модуль їх перевіряє. Знаходить помилку. Якщо вивести помилку, клієнт буде роздратований тим, що ввів багато даних і йому знову знадобиться їх вводити. Тому передані дані записуються в `webses_data` і відбувається редирект на сторінку введення, при цьому параметр `_unikey` встановлюється значення поля `unikey` збережених даних.

Перед завантаженням модуля, якщо `calls.pm` бачить параметр `_unikey`, він автоматично витягує з БД зв'язану з ключем рядок і дешифрує її змінну `$ses::data`. Далі, якщо існує ключ `$ses::data→{-input}`, то `calls.pm` «дає вигляд», що браузер надіслав POST/GET дані, що знаходяться `$ses::data→{-input}`.

Найчастіше ці події відбуваються прозоро, тобто. не потрібно думати про ключ `-input`. Наприклад, редирект завжди здійснюється через `webses_data`:

```
$Url->redirect(a=>'main', uid=>15);
```

В даному випадку відбудеться запис `{a=>'main', uid=>15}` у `webses_data`, після чого відбудеться перехід `http://xxx/?_unikey=uuuuuu`. Тут `uuuuuu` – ключ до даних, записаних у таблицю.

Якщо ми вже заговорили про редирект, то відзначимо 2 корисні його параметри `-made` і `-error`. Перший призначений для того, щоб вивести повідомлення вгорі сторінки після редиректу, при цьому якщо буде встановлено `-error`, то червоним кольором:

```
$Url->redirect(a=>'main', -error=>1,  
-made=>'Помилкова операція. Редиректимосся на титульну сторінку');
```

Підпрограми модуля main

ToLeft, ToRight, ToTop

Виводять дані в ліву, праву і відповідно у верхню частину html-сторінки, що формується.

```
ToLeft 'Текст зліва';
```

MessageBox виводить повідомлення у рамці

```
ToLeft MessageBox( 'Повідомлення 1');  
ToRight MessageBox( 'Повідомлення 2');
```

MessageWideBox виводить повідомлення у рамці, розтягнутій по ширині

<code> Show Message

From:
<https://ndp.pp.ua/> - my NoDeny Wiki

Permanent link:
<https://ndp.pp.ua/doku.php/nodeny/docs/calls.pm?rev=1674683227>

Last update: **25/01/2023 21:47**

