**Авторизация от Дмитрия Полоса**

wp\_create\_user() **-** Регистрирует нового пользователя. Указываются логин (имя), пароль и email.

$user\_id = wp\_create\_user( $username, $password, $email );

Возвращает **int|WP\_Error. или ID,** если удалось добавить пользователя. Если создать не удалось (имя или мыло уже существуют), вернет объект WP\_Error со следующими сообщениями об ошибке:

* empty\_user\_login — невозможно создать пользователя, потому что логин не указан (пуст).
* existing\_user\_login — такой пользователь уже зарегистрирован.
* existing\_user\_email — такой email уже существует.

Все что делает функция — это добавляет слэши в логин/email и вызывает

wp\_insert\_user() - Создает пользователя WordPress в Базе Данных.

wp\_authenticate() - Аутентифицирует пользователя (проверяет данные авторизации: логин/email и пароль) и возвращает **объект пользователя**, если проверка прошла успешно и **объект WP\_Error** в противном случае.

Функция не авторизует пользователя, а только проверяет данные. Для фактической авторизации полученные данные нужно использовать в wp\_set\_auth\_cookie(). Используйте wp\_signon(), когда нужно проверить логин, пароль и одновременно авторизовать юзера.

wp\_authenticate( $username, $password );

wp\_set\_auth\_cookie - Авторизует пользователя по ID. Устанавливает куки аутентификации на основе переданного ID пользователя.

wp\_set\_auth\_cookie( $user\_id, $remember, $secure, $token );

*$user\_id* (число) (обязательный) - ID пользователя. По умолчанию: нет

*$remembe* r(логический) - Нужно ли запоминать пользователя. true - увеличивает время жизни куков до 14 дней. По умолчанию (false) куки действительны в течении 2-х дней. Этот параметр аналогичен галочке "запомнить меня" в форме авторизации. По умолчанию: false

*$secure*(логический)- Устанавливает какое имя куки использовать, для ssl (защищенное) или нет? берется из констант: SECURE\_AUTH\_COOKIE или AUTH\_COOKIE. По умолчанию: is\_ssl()

**$token**(строка) - Токен сессии который нужно использовать для текущих куков. C WP 4.3. По умолчанию: ''

get\_userdata($user\_id) - Получает данные указанного пользователя в виде объекта WP\_User

Пример работы с пользователями - **регистрация и авторизация**:

**Фронтенд:**

<?php if (is\_user\_logged\_in()) { ?>

код страницы

<?php } else { ?>

<form class="registration" action="<?php echo esc\_url(admin\_url('admin-ajax.php')); ?>">

<input type="text" name="login" required>

<input type="password" name="pass" required>

<input type="email" name="email" required>

<input type="hidden" name="action" value="registration">

<button type="submit">Регистрация</button>

</form>

<form class="auth" action="<?php echo esc\_url(admin\_url('admin-ajax.php')); ?>">

<input type="text" name="login" required>

<input type="password" name="pass" required>

<input type="hidden" name="action" value="auth">

<button type="submit">Войти</button>

</form>

<style>

.auth {

display: none;

}

</style>

<script>

**window.onload** = () => {

const reg = document.querySelector('.registration');

const auth = document.querySelector('.auth');

**reg.addEventListener( 'submit**', async e => {

e.preventDefault();

const data = new FormData( e.target );

const result = **await fetch**( e.target.getAttribute('action'), {

method: 'POST',

body: data

});

if( result.ok ) {

e.target.style.display = 'none';

auth.style.display = 'block';

let id = await result.text();

console.log('id :', id);

}

});

**auth.addEventListener( 'submit'**, async e => {

e.preventDefault();

const data = new FormData( e.target );

const result = **await fetch**( e.target.getAttribute('action'), {

method: 'POST',

body: data

});

if( result.ok ) {

e.target.style.display = 'none';

let login = await result.text();

document.body.insertAdjacentHTML('afterbegin', `<h1>Привет, ${login}!</h1></h1>` );

}

});

}

</script>

<?php } ?>

**Бэкенд ()functions.php:**

// регистрация пользователей

add\_action( 'init', 'rl\_registration');

function rl\_registration() {

}

add\_action( '**wp\_ajax\_nopriv\_registration'**, 'rl\_reg');

function rl\_reg() {

$log = $\_POST['login'];

$pass = $\_POST['pass'];

$email = $\_POST['email'];

$res = **wp\_create\_user**($log, $pass, $email);

if( !is\_wp\_error($res) ) {

wp\_die($res, 200);

} else {

wp\_die('не получилось зарегистрироваться', 400);

}

}

add\_action(**'wp\_ajax\_nopriv\_auth'**, 'rl\_auth');

function rl\_auth() {

$log = $\_POST['login'];

$pass = $\_POST['pass'];

$res = **wp\_authenticate**( $log, $pass );

if (!is\_wp\_error($res)) {

**wp\_set\_auth\_cookie**( $res->ID );

wp\_die(**get\_userdata**($res->ID)->user\_login, 200);

} else {

wp\_die('не получилось авторизоваться', 400);

}

}