

Javascript

[/js/script.js](#)

Функция `user_registration_check`

Входящие параметры — объект с полями:

user_email — электронный адрес

success_callback_func — функция, на которую вешаем отслеживание успешного аякс-ответа

failure_callback_func — функцию, на которую вешаем отслеживание неудачного аякс-ответа

Функция формирует аякс-post-запрос к скрипту

/ajax/user_registration_check.php

..., в котором передаёт электронный адрес на проверку. Отслеживание успешного и неудачного аякс-ответов заряжается на колбэк-функции.

Функция `user_authorization_proceed`

Входящие параметры — объект с полями:

user_email — электронный адрес,

user_pass — пароль для авторизации,

success_callback_func — функция, на которую вешаем отслеживание успешного аякс-ответа

failure_callback_func — функцию, на которую вешаем отслеживание неудачного аякс-ответа

Функция формирует аякс-post-запрос к скрипту

/ajax/user_authorization_proceed.php

..., в котором передаёт электронный адрес и пароль. Отслеживание успешного и неудачного аякс-ответов заряжается на колбэк-функции.

PHP, Битрикс

[/ajax/user_registration_check.php](#)

Скрипт принимает post-запрос с параметром `user_email` и проверяет, зарегистрирован ли Битрикс-пользователь с таким электронным адресом.

Возвращает ответ в формате JSON, который содержит поля:

user_email — параметр, полученный при запросе

user_id — код пользователя либо ноль, если пользователя с указанным электронным адресом нет в системе. Возможно несколько пользователей с одним электронным адресом, потому сортируем по ID => DESC (старшие — вперёд) и возвращаем первый найденный.

Дополнительно:

1. Перед выполнением действий скрипт проверяет, с какого домена был запрос. Доменное имя указывается в начале файла, в переменной-массиве. Формат:

```
allowed_domains= [ 'site.com', 'subdomain.site.com' ]
```

Если проверка закончилась неудачей, обрываем скрипт.

2. Скрипт ведёт лог запросов, которые складывает в папку

```
/ajax/user_registration_check/Текущая_дата.log
```

Лог-файл представляет собой текстовые блоки с информацией:

request_ip — ip-адрес, с которого получен аякс-запрос

user_email — полученный параметр с электронным адресом

user_id — результат проверки

access_denied — доменное имя, с которого пришёл запрос, но который не прошёл проверку

Формат лог-блока, пример №1:

```
request_ip:192.168.0.1
```

```
user_email:admin@site.com
```

```
user_id:1540
```

Формат лог-блока, пример №2:

```
request_ip:192.168.5.5
```

```
access_denied:nosite.com
```

[/ajax/user_authorization_proceed.php](#)

Скрипт принимает post-запрос с параметрами: user_email и user_pass и производит попытку авторизовать Битрикс-пользователя.

Возвращает ответ в формате JSON, который содержит поля:

authorization_status — авторизация прошла успешно или нет (true|false)

error_message — сообщение об ошибке, которая произошла в случае неуспешной авторизации.

Дополнительно:

1. Перед выполнением действий скрипт проверяет, с какого домена был запрос.

Доменное имя указывается в начале файла, в переменной-массиве. Формат:

```
allowed_domains= [ 'site.com', 'subdomain.site.com' ]
```

Если проверка закончилась неудачей, обрываем скрипт.

2. Скрипт ведёт лог запросов, которые складывает в папку

```
/ajax/user_authorization_proceed/Текущая_дата.log
```

Лог-файл представляет собой текстовые блоки с информацией:

request_ip — ip-адрес, с которого получен аякс-запрос

user_email — полученный параметр с электронным адресом

user_pass — полученный параметр с паролём

authorization_status — авторизация прошла успешно или нет (true|false)

error_message — сообщение об ошибке, которая произошла в случае неуспешной авторизации.

access_denied — доменное имя, с которого пришёл запрос, но который не прошёл проверку