

Тестовое задание

на позицию «Backend-разработчик / PHP-программист»

Задача: Реализация авторизации и регистрации в приложении (Реализации API серверного приложения)

Описание задачи: Вашей задачей является разработка серверной части приложения, которая позволит пользователям авторизовываться и регистрироваться в нем.

1. Создать проект на языке программирования PHP 7.4 (Возможно использование фреймворка – Codeigniter4, Yii, Laravel, Symfony)
2. Реализовать механизм регистрации, который позволит пользователям зарегистрироваться в вашем приложении
3. Реализовать механизм авторизации, который будет проверять подлинность пользователей

Требования к задаче:

1. Используйте механизм хранения данных пользователей в базе данных (используйте СУБД MYSQL)
2. Обеспечьте безопасность передачи и хранения пользовательских данных
3. Документируйте ваш код и создайте простую инструкцию по развертыванию и запуску вашего сервера
4. Разместите решение задачи на GitHub

Имя запроса авторизации и регистрации: user_auth

Метод запроса: GET

Входящие данные: access_token, id, first_name, last_name, country, city, sig

Таблица для хранения данных: users, поля id (int) – идентификатор пользователя, first_name (string) – Имя пользователя, last_name (string) – фамилия, city (string) – Город, country (string) – Страна. Таблица для хранения сессий – users_sessions (user_id (int) – идентификатор пользователя, access_token (string))

Формат ответа: JSON. В Ответе вернуть данные: id, first_name, last_name, city, country, access_token в случае успешной авторизации/регистрации, в противном случае ошибку.

Примечание: *access_token* носит в задаче формальный характер и будет использоваться в последующих запросах, не указанных в данном задании.

Данные для теста:

Секретный ключ приложения: 977fea8deca4c2c2330544cf2e388284 – Необходимо задать на сервере (например, в «конфиге»)

Входящие данные: access_token=07b38cd0e778340eb40b25e005476ce8, id=1, first_name=Иван, last_name=Иванов, city=Москва, country=Россия, sig=5427b31460cd807aab7e184364960958

Алгоритм проверки подписи:

1. Сортировка входящих параметров по ключу (используйте ksort)
2. Сформируйте строку \$str из полученных параметров исключив из нее sig
3. К концу полученной строки добавьте секретный ключ
4. С полученной строкой выполните mb_strtolower(md5(\$str), 'UTF-8') и сравните результат с пришедшем sig
5. Если mb_strtolower(md5(\$str), 'UTF-8') === sig, зарегистрируйте пользователя, в противном случае выдайте ошибку

Пример полученной строки относительно входящих параметров:

access_token=07b38cd0e778340eb40b25e005476ce8city=Москvacountry=Россияfirst_name=Иванid=1last_name=Иванов977fea8deca4c2c2330544cf2e388284

mb_strtolower(md5('access_token=07b38cd0e778340eb40b25e005476ce8, id=1, first_name=Иван, last_name=Иванов, city=Москва, country=Россия, sig=5427b31460cd807aab7e184364960958'), 'UTF-8') === 5427b31460cd807aab7e184364960958

Если подпись верна, проверьте наличие пользователя в таблице users по id. Если нет пользователя, то добавьте его в таблицу users. Если пользователь уже существует, сделайте update присланных полей: first_name, last_name, city и country. Также примените запрос replace к таблице users_sessions (user_id (уникальный ключ) = id, access_token)

Пример успешного ответа:

```
{
  "access_token": "07b38cd0e778340eb40b25e005476ce8",
  "user_info": {
    "id": 1,
    "first_name": "Иван",
    "last_name": "Иванов",
    "city": "Москва",
    "country": "Россия",
  },
  "error": "",
  "error_key": "",
}
```

Входящие данные: access_token=07b38cd0e778340eb40b25e005476ce8, id=1, first_name=Иван, last_name=Иванов, city=Москва, country=Россия, sig=4427b31460cd807aab7e184364960958

Пример неуспешного ответа:

```
{  
  "error": "Ошибка авторизации в приложении",  
  "error_key": "signature error",  
}
```

Примечания:

1) ДЕДЛАЙН выполнения задания – 3 дня с момента получения.

Выполнение раньше поставленного срока будет являться плюсом.

2) Необходимо зафиксировать общее время, которое было потрачено на выполнение данного тестового задания, и указать его в файле README.txt (файл положить в корне проекта)

3) Код для выполнения задания необходимо писать на PHP 7.4 (Возможно использование фреймворка – Codeigniter4, Yii, Laravel, Symfony)

4) Готовое тестовое задание необходимо отправить вашему HR-менеджеру (https://t.me/Natali_HRit) , к письму прикрепить Ваше резюме (или ссылку на него).

Файлы готового проекта необходимо разместить на Google Drive, Яндекс.Диске или Github и прислать ссылку на скачивание.