

Реализовать модуль шифрования данных клиента

```
1  const customers = [  
2    {  
3      name: 'Pitter Black',  
4      email: 'pblack@email.com',  
5      password: 'pblack_123'  
6    },  
7    {  
8      name: 'Oliver White',  
9      email: 'owhite@email.com',  
10     password: 'owhite_456'  
11   }  
12 ];  
13  
14 const ui = new Ui(customers);  
15 const guardian = new Guardian();  
16 const manager = new AccountManager();  
17  
18 ui.pipe(guardian).pipe(manager);
```

1. Реализовать класс **Ui** который будет имплементировать **Readable** интерфейс и будет служить поставщиком данных.
2. Реализовать класс **Guardian** который будет имплементировать **Transform** интерфейс и будет служить для шифрования данных. Для шифрования пароля будем использовать преобразование строки в **hex** формат. Зашировать необходимо **email** и **password**.

Пример преобразования для первого объекта.

```
1  // Было  
2  {  
3    name: 'Pitter Black',  
4    email: 'pblack@email.com',  
5    password: 'pblack_123'  
6  }  
7  
8  // Стало  
9  {  
10     meta: {  
11       source: 'ui'  
12     },  
13     payload: {  
14       name: 'Pitter Black',
```

```
15         email: '70626c61636b40656d61696c2e636f6d',  
16         password: '70626c61636b5f313233 '  
17     }  
18 }
```

1. Реализовать класс **AccountManager** который будет имплементировать **Writable** интерфейс и будет служить в качестве хранилища данных.

Обратите внимание!

1. Все поля в объекте пользователя являются обязательными. Генерировать ошибку в случае если условие не соблюдено.
2. Все поля в объекте пользователя должны быть строками. Генерировать ошибку в случае если условие не соблюдено.
3. Объект пользователя не должен содержать дополнительных полей. Генерировать ошибку в случае если условие не соблюдено.