**Введение и общее описание модуля.**   
Мы почти закончили с модулем синхронизаций для собирания и синхронизаций информаций про аккаунт, теперь нужно написать модуль, который будет заказывать корзину клиента, то есть открывать ссылку корзину, проверять корзину, применять промокод, записывать данные, заполнять поля и тд.   
Также у нас будут некоторые параметры, например смена адреса через х количество заказов, потому что по закону таможни нельзя на одного человека много корзин заказывать и тд или параметр для задания максимальной суммы корзины.  
Этот модуль гораздо обширен и сложен по сравнению синхронизацией, потому что здесь много действий, и наша задача не просто получать информацию с сайта а заполнять поля, нажимать на кнопки и тд.  
Как обычно скрипт работает на папитир и это проект является продолжением и дополнением другого проекта.

**Входящие данные и параметры для заказа.**  
Эти данные будут использоваться для заказа, среди них есть входящие данные, которые в будущем могут быть получены от телеграм бота по веб сокету, а также некоторые параметры – настройки.   
Пока что админ сам вручную получает эти данные из ТГ бота и вручную записывает в конфиг файл.  
(это общее введение)  
1) ссылки на корзину   
2) аккаунт для заказа   
3) карта   
4) адрес и настройка  
5) промокод и настройка   
6) клиент.   
  
  
Заказ и запись данных.

Начало заказа: нужные данные и параметры   
 1) **ссылки на корзину**   
обычная ссылка на корзину, в ссылке на корзину содержаться все нужные товары   
 2) **аккаунт для заказа**   
выбираем аккаунт для заказа  
 3) **карта**   
при заказе на новый адрес нужно ввести номер карты заново, по этому нужный параметр, вводится только номер карты и все.   
 4) **адрес и настройка**  
выбор адреса для заказа, обычно нам нужно только имя и фамилия.  
 5) **промокод и настройка**   
промокод для заказа и реферальный код.  
 6) **клиент**.  
имя клиента, айди и тд.  
  
  
  
  
  
  
**1)** **первый шаг, получение настроек и их установка и перенастройка окружения.**все эти данные, а также скрипты будут храниться по такому пути:  
C:\Users\Didar1520\Docks\CRM\scripts\actions\ordersInputConfig.js

Примерный вид данного файла (этот вид примерный в текстовом формате, нужно переделать его в JS :   
ordersInputConfig.js (это входящие параметры, в финальной версий проекта он будет поступать через интерфейс в браузере через инпуты, пока что задаем в файле).  
вот данные и параметры, если они заданы будут браться именно они, но если они не заданы будут браться параметры по умолчанию из файла **ordersConfig.js**  
  
   
**ordersInputConfig.js (данные для примера):**  
**аккаунт**: Didar2520 или default  
**промокод**: Gold120 или default  
**реферальный код**: DUA9670 или default  
**ссылка на корзину**: <https://www.iherb.com/pal/referral/ARN0872/cart?pcodes=HEE-10035qty5_SNS-00803qty1_NOW-00369qty1_CGN-01597qty1_SOR-12109qty1_GIO-18361qty1_TUK-00730qty1&rcode=ARN0872&utm_medium=appshare>

**клиен**т: ID54 (Иван Владимирович) или default .   
  
Это данные входящей корзины и его параметров, таких корзин может быть много.

**Параметры по умолчанию (**в той же директорий нужно создать файл **ordersConfig.js**):  
**Клиент**:   
**Аккаунт**:  
**Промокод**:  
**Реферальный** код:  
**Валюта**:  
**Карты**: 5457 4567 5474 5647, 5474 5474 4571 5477 и тд  
**Адрес для заказа**: N X (в файле адреса для заказа нужно взять адрес с индексом {4} для примера)

**Настройки:**  
**Настройка частоты смены адресов**: данные числовые, программа должна менять адрес каждые X заказов .  
**Настройка смены аккаунтов**: данные числовые, программа должна менять адрес каждые X заказов.  
(какой адрес и аккаунт использовался для заказа хранится в C:\Users\Didar1520\Docks\CRM\data\OrdersData\ordersData.json  
orderAccount и deliveryAddress соответственно

Когда настройки получены начинаем модуль заказа.   
**Проверка аккаунта и адреса**   
1) Проверка авторизаций**, считываем аккаунт для заказа** из параметров и ищем директорию аккаунта, используем если нашли, если нет создаем и авторизуемся, а если нашли то проверяем состояние авторизаций и проверяем нужный ли аккаунт, в целом все это есть в модуле авторизаций auth.js по этому просто используем его.   
  
Проверка нужного адреса.   
Этот код уже написан. Нужно будет его переделать что бы он работал в нашем проекте, он раньше работал в расширений для хром Automa который имеет графический интерфейс и может выполнять JS код или внутренние блоки.   
  
Давай сначало настроим среду, параметры и тд все что я написал и откроем просто ссылку на корзину из параметров и введем в консоль все данные из параметров и укажем получили мы данные из входящих параметров или по умполчанию и тд.