Деление на модули   
  
**Верхний управляющий модуль**:  
У нас экшены, то есть совершение заказа состоит из нескольких модулей и нужно будет управлять параметрами программы, нужен какой-то высший и верхний модуль, который будет это делать.   
Он должен выполнять некоторые утилиты и в результате этих утилит он должен выбрать аккаунт для заказа или адрес, промокод и прочие вещи.   
  
**Примеры утилит**:  
**Утилита по товарам**  
Утилита по выбору аккаунтов в зависимости от кол-во товаров в корзине.   
Эта утилита проверяет товары в корзине, а также базу данных заказанных товаров в аккаунтах.   
И выбирает аккаунт, где эти товары встречаются наименьшее кол-во или вовсе не совпадают.  
Эта утилита уже написана на JS его нужно внедрить в проект.   
Цель высшего модуля — это смотреть на корзину, и если в корзине уникальные товары соответствует минимальному, то вызвать функцию по выбору аккаунтов и найти нужный аккаунт для заказа и записать этот аккаунт в инпут конфиг или куда-либо еще и запустить заказ через этот аккаунт.   
  
Это укороченный пример для работы с модулем выбора аккаунтов, также модуль выбора аккаунтов можно выключить.   
  
В целом этот высший модуль станет интерфейсом и управляющим, который связывает администратора программы и программу.   
  
  
**Утилита для написания отзывов:**  
например есть утилита который пишет заранее заготовленные отзывы от специалистов на нужные товары, но открывать новый профиль и писать для каждого аккаунта JS код в консоли неудобно, поэтому высший модуль должен запустить этот модуль на нужном аккаунте используя auth.js на нужной странице и записать результат действия.

**Также другие динамические параметры,** с которым должен работать высший модуль:  
**Глобальный промокод.**   
Промокоды бывают многоразовые или одноразовые.   
Этот параметр указывается в настройках.   
Также есть срок годности некоторых промокодов а некоторые вечные.   
Например если промокод одноразовый то высший модуль должен сменить аккаунт на другой в зависимости от на сколько заказов действует промокод (если на 2 заказа то через 2 заказа он помечает этот аккаунт неактивным и пока заказы требуют одноразовый код на то время он не будет заказывать с этого аккаунта и высший модуль не будет его использовать его для заказов ( то есть будет менять в конфиге аккаунт для заказа).   
  
**Адрес заказа**  
Адрес заказа должен меняться через 2-3 заказа, и высший модуль должен смотреть на какой адрес был сделан последний заказ и если он повторяется последние 30 дней больше 3 раз он берет следующий адрес (устанавливает в конфиги).

**Группа аккаунтов.**Например, в настройках можно выделить какую то группу аккаунтов что бы использовать только их.  
  
**Контроль суммы заказов:**  
Мы можем добавить в корзину товаров на сумму 190 долларов и пока мы вводим промокоды и заполняем данные таможни, некоторые товары могут пропасть из наличия или закончится промокод, высший модуль должен контролировать исходную версию корзину и сравнить с финальной версией и если сумма изменилась он должен сообщить об этом или спросить продолжить оформление или прекратить.

В целом он должен изменять исходящие параметры для заказа а также запускать нужные утилиты и на основе полученных данных изменять инпут параметры, а также контролировать что все происходит идеально и сообщить если что то пойдет не так.   
Также иногда сайт может зависнуть и его нужно будет просто перезагрузить.

1) **Страница** **добавление** **товаров**, простая страница, просто нужно нажать кнопку добавить в корзину.   
Эта страница открывается, когда мы открываем ссылку на корзину.   
ссылка на корзину примерно такая:  
<https://www.iherb.com/pal/referral/AFN8378/cart?pcodes=CGN-01251qty1_NAP-03770qty1_CAR-01524qty1_KIM-02454qty2_VTP-00509qty1_DRB-00250qty6>  
Здесь мы должны убедится что корзина пустая если там есть товары то нужно очистить корзину.   
(тут только коротко, как все это проверить и какие селекторы или апи запросы есть я напишу позже для всех модулей)

2,1) **страница адресов (adressBookPage)** после добавление товаров, у нас открывается страница корзины: <https://checkout12.iherb.com/cart>  
В айхербе такая архитектура, что бы сайт корректно работал, нужно выбрать адрес заранее в списке адресов, если нету нужно адреса то нужно добавить адрес.   
Поэтому перед операциями со страницей корзины мы заходим в адресную книгу айхерб.  
Который находится по адресу: <https://checkout12.iherb.com/users/address-book>  
К нам приходит исходящий параметр в интерфейсе который ты должен реализовать адрес который нам нужно выбрать.   
В списке адресов мы должны выбрать адрес.  
Например Эндрю Айхерб и далее код должен сделать проверки.   
2,2) **страница добавление адресов (addAdressPage)**. Это страница открывается если в адресной книге не оказалось нужного адреса.   
у него такой адрес: <https://checkout12.iherb.com/users/address-book/add-new-address>  
и там такие селекторы и рядом покажу пример заполнения:  
**3) страница корзины (cartPage)**после добавление товаров, у нас открывается страница корзины: <https://checkout12.iherb.com/cart>  
Здесь находятся наши товары и здесь мы будем применять промокод, устанавливать тип доставки, регион, валюту и прочее

5) **страница PreOrderPage**  
Тут предварительная информация и проверки (на этапе оформление все ли товары на месте, стоит тот же адрес что сохранили, запрашивает ли сайт ввести номер карты заново и так далее)

6) Инвойс   
invoice page/   
Здесь результаты нашего заказа, его номер, сумма, товары и прочее   
  
  
Тут только примерные данные, укороченные для чего этот модуль, более подробно напишем когда будем останавливаться на каждом из них