PreWork (once a day):

* Make API Calls to get initial list of products to display
  + Currency conversion to USD
* Inactivate Old Products from previous day
* Add new list of products to product Table

Landing Page:

* Create session\_ID
  + Collect data that we can collect
* Validate that products are still available before showing to user
  + - Query DB for products in price point
    - Make API call for each product using Listing ID and determine availability
* Render HTML page with products to choose from (for their specified box size. Column delimited)

MysteryBox Logic

* Select only set amount (4)
* Validate products are still available before showing to user
  + Query DB
    - We will have a set list of parameters
      * Ex. For $50 box we will always auto select a 20, 10 4 @$5 or whatever algorithm we come up with
  + Make API calls to check for availability
* Add to Boxes table
* Update product table with new quantity
* Add to Cart
* Update User table with box info
* If we have pet details update Pet table

Build custom box

* Create logic to get items from DB to bring back for given price point/parameters
  + Validate they are still valid
  + Query DB and return
    - Listing\_id
    - Title
    - Image
    - Price
    - Size
    - description
* Select from set amount, column delimited
* Query DB to bring a list of items
  + Listing\_id
    - To check for
* Create logic to allow users to select one item from each price point column
  + Once price point is reached, bring Purchase Now to focus

Create New Account

* Capture First Name, Last Name, Address, City, State, Zip
  + Nice to have Pet details
* Finish populating session table

Login

* Finish populating session table
* Bo

Cart

* If not mystery box, show build a box items
* If mystery box, show mystery box quantity
* Purchase total
  + Shipping and handling
  + Taxes
* Update product table with new quantity

Confirmation

* Reveal mystery box content
* Show total charges

How are we using Express? mySQL? Handlebars? Travis? Bootcamp? API?

Express