# Site creates new user POST /Users/email { Name:name }

## Create User & add to user-dictionary

### If User already exists, fail

# Site logs in user

## If user not in dictionary, fail

# User creates new Item POST /Users/email/ItemsSelling/ { name : n, imageUrl: u, Expiration: e, StartAmount: a }

## If user doesn’t exist, fail

## Create Item transactionally modify user, items, and unexpired items dictionaries

### If ItemId already exists, fail

### If items gets large, old item value (but not key) can move to warm storage

# User bids on other user’s item POST /Users/bidderEmail/ItemsBidding/Item

## If bidding user doesn’t exist, fail

## Update bidder user to track itemID

### NOTE: item may not exist or network could fail after this

## Tell seller’s partition we’re attempting bid

### If item not in unexpired dictionary, fail

### Attempt update of per-user-items dictionary

#### If bid fails tests, fail

# Users checks items they’re bidding GET /Users/email/ItemsBidding/[item]

## If user doesn’t exist, fail

## For each item bidding, look up item & return bids

### Note: user may not have a bid due to network failure or bid didn’t pass tests

# User checks items they’re selling GET /Users/email/ItemsSelling/[item/]

## If user doesn’t exist, fail

## For each item in –per-user dictionary, return items

# Garbage Collect expired items

## Periodically scan per-partition unexpired items and delete old ones

# Show items for auction GET /Items/[item]

## Return items in each partition’s unexpired dictionary

### Note: Same may be expired; can purge now or wait for GC, or UI can filter