Home Mixer

==========

Home Mixer is the main service used to construct and serve Twitter's Home Timelines. It currently

powers:

- For you - best Tweets from people you follow + recommended out-of-network content

- Following - reverse chronological Tweets from people you follow

- Lists - reverse chronological Tweets from List members

Home Mixer is built on Product Mixer, our custom Scala framework that facilitates building

feeds of content.

## Overview

The For You recommendation algorithm in Home Mixer involves the following stages:

- Candidate Generation - fetch Tweets from various Candidate Sources. For example:

- Earlybird Search Index

- User Tweet Entity Graph

- Cr Mixer

- Follow Recommendations Service

- Feature Hydration

- Fetch the ~6000 features needed for ranking

- Scoring and Ranking using ML model

- Filters and Heuristics. For example:

- Author Diversity

- Content Balance (In network vs Out of Network)

- Feedback fatigue

- Deduplication / previously seen Tweets removal

- Visibility Filtering (blocked, muted authors/tweets, NSFW settings)

- Mixing - integrate Tweets with non-Tweet content

- Ads

- Who-to-follow modules

- Prompts

- Product Features and Serving

- Conversation Modules for replies

- Social Context

- Timeline Navigation

- Edited Tweets

- Feedback options

- Pagination and cursoring

- Observability and logging

- Client instructions and content marshalling

## Pipeline Structure

### General

Product Mixer services like Home Mixer are structured around Pipelines that split the execution

into transparent and structured steps.

Requests first go to Product Pipelines, which are used to select which Mixer Pipeline or

Recommendation Pipeline to run for a given request. Each Mixer or Recommendation

Pipeline may run multiple Candidate Pipelines to fetch candidates to include in the response.

Mixer Pipelines combine the results of multiple heterogeneous Candidate Pipelines together

(e.g. ads, tweets, users) while Recommendation Pipelines are used to score (via Scoring Pipelines)

and rank the results of homogenous Candidate Pipelines so that the top ranked ones can be returned.

These pipelines also marshall candidates into a domain object and then into a transport object

to return to the caller.

Candidate Pipelines fetch candidates from underlying Candidate Sources and perform some basic

operations on the Candidates, such as filtering out unwanted candidates, applying decorations,

and hydrating features.

The sections below describe the high level pipeline structure (non-exhaustive) for the main Home

Timeline tabs powered by Home Mixer.

### For You

- ForYouProductPipelineConfig

- ForYouScoredTweetsMixerPipelineConfig (main orchestration layer - mixes Tweets with ads and users)

- ForYouScoredTweetsCandidatePipelineConfig (fetch Tweets)

- ScoredTweetsRecommendationPipelineConfig (main Tweet recommendation layer)

- Fetch Tweet Candidates

- ScoredTweetsInNetworkCandidatePipelineConfig

- ScoredTweetsTweetMixerCandidatePipelineConfig

- ScoredTweetsUtegCandidatePipelineConfig

- ScoredTweetsFrsCandidatePipelineConfig

- Feature Hydration and Scoring

- ScoredTweetsScoringPipelineConfig

- ForYouConversationServiceCandidatePipelineConfig (backup reverse chron pipeline in case Scored Tweets fails)

- ForYouAdsCandidatePipelineConfig (fetch ads)

- ForYouWhoToFollowCandidatePipelineConfig (fetch users to recommend)

### Following

- FollowingProductPipelineConfig

- FollowingMixerPipelineConfig

- FollowingEarlybirdCandidatePipelineConfig (fetch tweets from Search Index)

- ConversationServiceCandidatePipelineConfig (fetch ancestors for conversation modules)

- FollowingAdsCandidatePipelineConfig (fetch ads)

- FollowingWhoToFollowCandidatePipelineConfig (fetch users to recommend)

### Lists

- ListTweetsProductPipelineConfig

- ListTweetsMixerPipelineConfig

- ListTweetsTimelineServiceCandidatePipelineConfig (fetch tweets from timeline service)

- ConversationServiceCandidatePipelineConfig (fetch ancestors for conversation modules)

- ListTweetsAdsCandidatePipelineConfig (fetch ads)