Product Mixer

=============

## Overview

Product Mixer is a common service framework and set of libraries that make it easy to build,

iterate on, and own product surface areas. It consists of:

- \*\*Core Libraries:\*\* A set of libraries that enable you to build execution pipelines out of

reusable components. You define your logic in small, well-defined, reusable components and focus

on expressing the business logic you want to have. Then you can define easy to understand pipelines

that compose your components. Product Mixer handles the execution and monitoring of your pipelines

allowing you to focus on what really matters, your business logic.

- \*\*Service Framework:\*\* A common service skeleton for teams to host their Product Mixer products.

- \*\*Component Library:\*\* A shared library of components made by the Product Mixer Team, or

contributed by users. This enables you to both easily share the reusable components you make as well

as benefit from the work other teams have done by utilizing their shared components in the library.

## Architecture

The bulk of a Product Mixer can be broken down into Pipelines and Components. Components allow you

to break business logic into separate, standardized, reusable, testable, and easily composable

pieces, where each component has a well defined abstraction. Pipelines are essentially configuration

files specifying which Components should be used and when. This makes it easy to understand how your

code will execute while keeping it organized and structured in a maintainable way.

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.