

API Documentation

Change Log

Date	Changed By	Comments
13 June 2018	Sidharth Shah	Proposed First Draft

Table of Contents

Table of Contents	3
Introduction	4
API Products	4
Recommendation Engine	4
Features	4
How it Works?	4
Search Engine custom Plugins	4
Features	4
How it Works?	4
Crawling APIs	5
Features	5
Why FineAI?	5
Focus on your core	5
Continuous Improvements	5
Auto-tune your Algorithms	5
Recommendation APIs	6

Introduction

FineAI is suite of products that help our customers achieve last-mile in terms of user experience. Following are some of our customer segments

1. Enterprises looking to leverage un-structured data collected in form of:
 1. User interaction logs
 2. Content getting created {E.g. Reviews, Complaints, Feedbacks}
2. Startups looking to iterate faster without investing heavily in specialized AI and NLP tools

API Products

Recommendation Engine

Features

1. Easy API Integration
2. Pre-built models to choose from
3. Auto-tuned recommendation algorithms
4. Our experts help you evaluate results

How it Works?

1. Log key data using our API
2. Choose parameters of your models
3. Call our API to get Recommendations

Search Engine custom Plugins

Features

1. Pre-build Apache Solr Plugins
2. Auto-suggestions/Correction data generated
3. Ability to support queries with custom grammar

How it Works?

1. Install our Request Handler
2. Index your Data
3. Call our API to get Results from our Custom Handler

Crawling APIs

Features

1. Leverage high performance crawling infrastructure
2. Support of rotating proxies
3. Easy setup and tear down
 - How it Works?
 1. Configure your job
 2. Monitor vital stats
 3. Get compressed dumps of files

Why FineAI?

Focus on your core

Don't reinvent the wheel, our team has been working on IR and NLP tools for couple of years now. Do what you do the best, leave the rest to us.

Continuous Improvements

We keep on tuning algorithms across different customer and different verticals. We have figure out the science of stuff that we do.

Auto-tune your Algorithms

Getting an algorithm to behave right can be daunting, we've build a suite of tools that allow us to select what is best for you.

Recommendation APIs

There are 4 key APIs

1. **"/log"**: API used to collect key events which are fed to recommendation algorithm

Log Creation

Method: POST

Parameters: **ALL** parameters are mandatory

1. **app_id**: You app id to identify your events
2. **user_id**: User id representation {integer}
3. **item_id**: Item id representation {integer}
4. **action**: Identifier of what action user performed on the item (E.g. view, search, add, upvote, downvote) {string}
5. **rating**: Rating user has given {integer}

2. **"/user"**: API used to manage collection of users

User Creation

Method: POST

Parameters: Parameters marked in **Bold** are mandatory

1. **app_id**: You app id to identify your events
2. **user_id**: User id representation {integer}
3. **key:value**: This API accepts multiple Key Value pair

3. **"/item"**: API used to manage collection of items. Items is an abstraction that allow you to generate recommendations. E.g. Item could be list of Movies or Song Albums. At same time item can be News Items or Image items

Item Creation

Method: POST

Parameters: Parameters marked in **Bold** are mandatory

4. **app_id**: You app id to identify your events
5. **item_id**: User id representation {integer}
6. key:value: This API accepts multiple Key Value pair

4. **"recommend"** : API used to actually to generate recommendations

Get Recommendation

Method: GET

Parameters: Parameters marked in **Bold** are mandatory

7. **app_id**: You app id to identify your events
8. **user_id**: User id representation {integer}
9. action: What action to consider for generate recommendations, default action is "view"