

# Services and Management Engineering

---

MIXIT



# ABOUT THE TEAM

---

As avid mobile users, we recognize that an overabundance of social networks and apps may have negative effects on our lives.  
To address this, we want to develop an all-in-one application.



Pedro Rocha - 98256



João Amaral - 98373



João Viegas - 98372



André Clérigo - 98485



# Stakeholders

01

## Market Perspective (One Stop App)

Only one app that concatenates services like Youtube and Twitter.

03

## Mental Health (stress and anxiety)

Multiple social media accounts and switching between different social networks may lead to risk of addiction and burnout.

02

## Advertising

In order to fund and support MIXIT, advertisement views and clicks may be sold on our app.

04

## FOMO

With all this services not gathered in one place, FOMO becomes a real problem.





# Provided Services

## Social Network



Uploading content, commenting, rating, searching and management.

## Advertisements



Selling views and clicks for money to fund MIXIT.

## Profile Management



Will work as the compositor. Manage the users linked accounts, full timeline

## Authentication



Authenticate (link) the users on the main apps (Google, Github, Twitter...) and MIXIT





# Cutting Corners

## Social Network



As time went on the Social API developer realised that there wasn't enough time to do every Social App that we had in mind so we focused on using Twitter and building the Social API to be scalable.

## Authentication



The Authentication Service developer wasn't able to get a Twitter Developer Account therefore we couldn't do Login/Register through it and consequently unable to do POSTs, PUTs and DELETEs on Twitter API.



# Advertisements API Service

---

MIXIT

# Advertisements Service

## Main idea

- Give the advertisers a **platform** to upload and manage their ads
- Give the publishers an **API** to consume and integrate ads on their apps
- Ensure all **metrics** are tracked and available for the advertisers



# Advertisements Service Architecture



# Advertisements API Overview

## Users

**users** Register, authenticate and profile of an user on the platform

**POST** `/v1/users` Registers a new user

**POST** `/v1/login` Authenticate an user

**GET** `/v1/profile` Get the information about an user

## Ads

**ads** Register / information about advertisements on the platform

**GET** `/v1/ads` Get ads html snippet based on filtering

**POST** `/v1/ads` Upload a new ad to the platform

**DELETE** `/v1/ads/{ad_id}` Delete an ad from the platform

**GET** `/v1/ads/{ad_id}` Update the ad's clicks

## Analytics

**analytics** Tracking of impressions and clicks of an advertisement

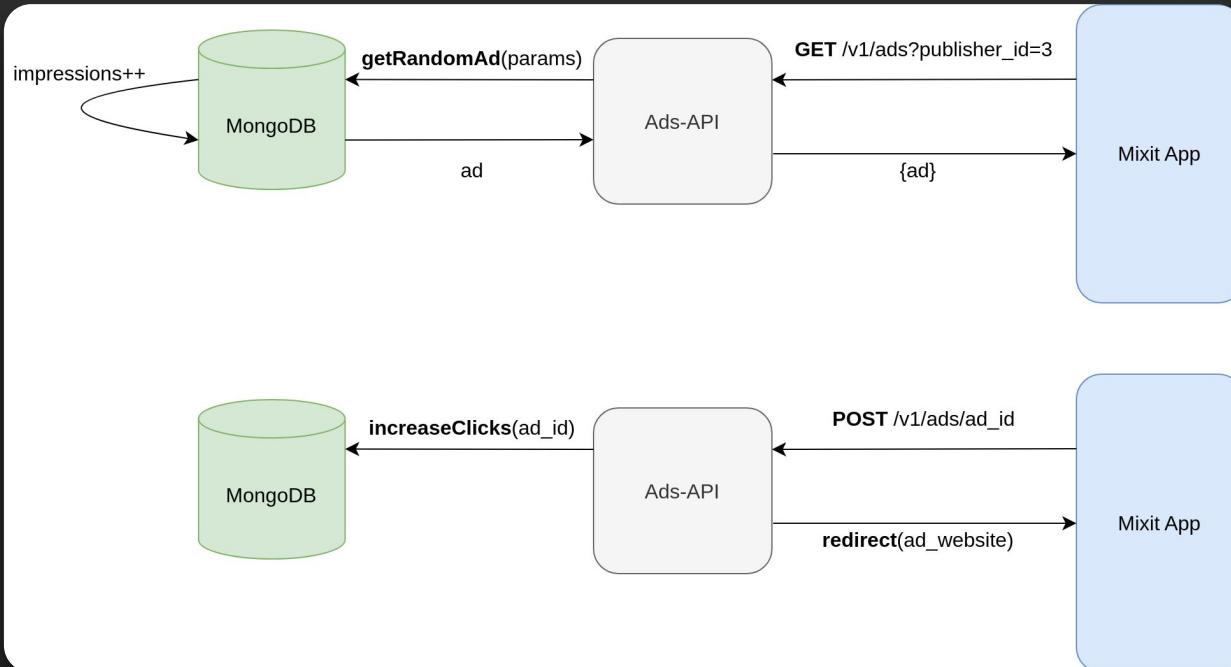
**GET** `/v1/analytics/ad/{ad_id}` Get the analytics of an advertisement

**GET** `/v1/analytics/advertiser/{adv_id}` Get the analytics of an advertiser of the platform



# Advertisements API Analytics

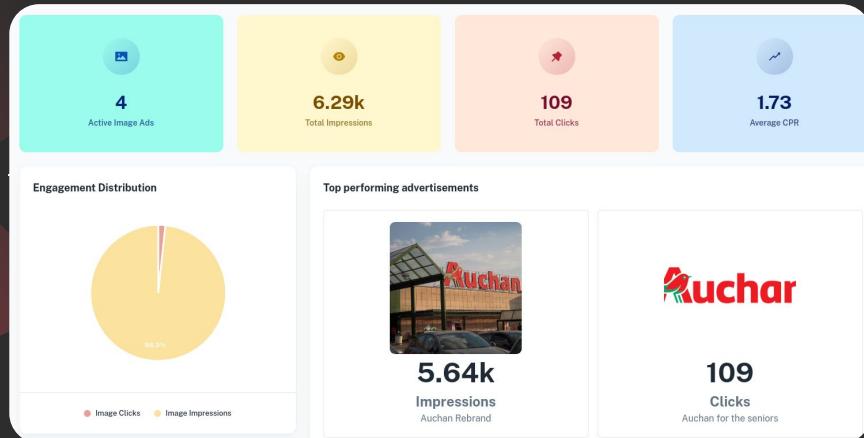
CPM



CPC



# Advertisements Platform



ID	Description	Pricing model	Impressions	Clicks	Goal	Actions
4	Auchan for the seniors	CPC	522	109	20	Delete
7	Auchan Rebrand	CPM	5640	0	8000	
10	Auchan Retail	CPM	27	0	200	
11	Cabaz Poupança	CPM	100	0	100	

### My Advertisements

Advertisement Type	Image Preview	Description	Views	Clicks
Image Ad		Auchan for the seniors	522 views	109 clicks
Image Ad		Auchan Rebrand	5640 views	0 clicks
Image Ad		Auchan Retail	27 views	0 clicks
Image Ad		Cabaz Poupança	100 views	0 clicks



# Social API Service

---

MIXIT

+

# Social API Overview

Our Social API is simple yet robust.

**Users** User information ^

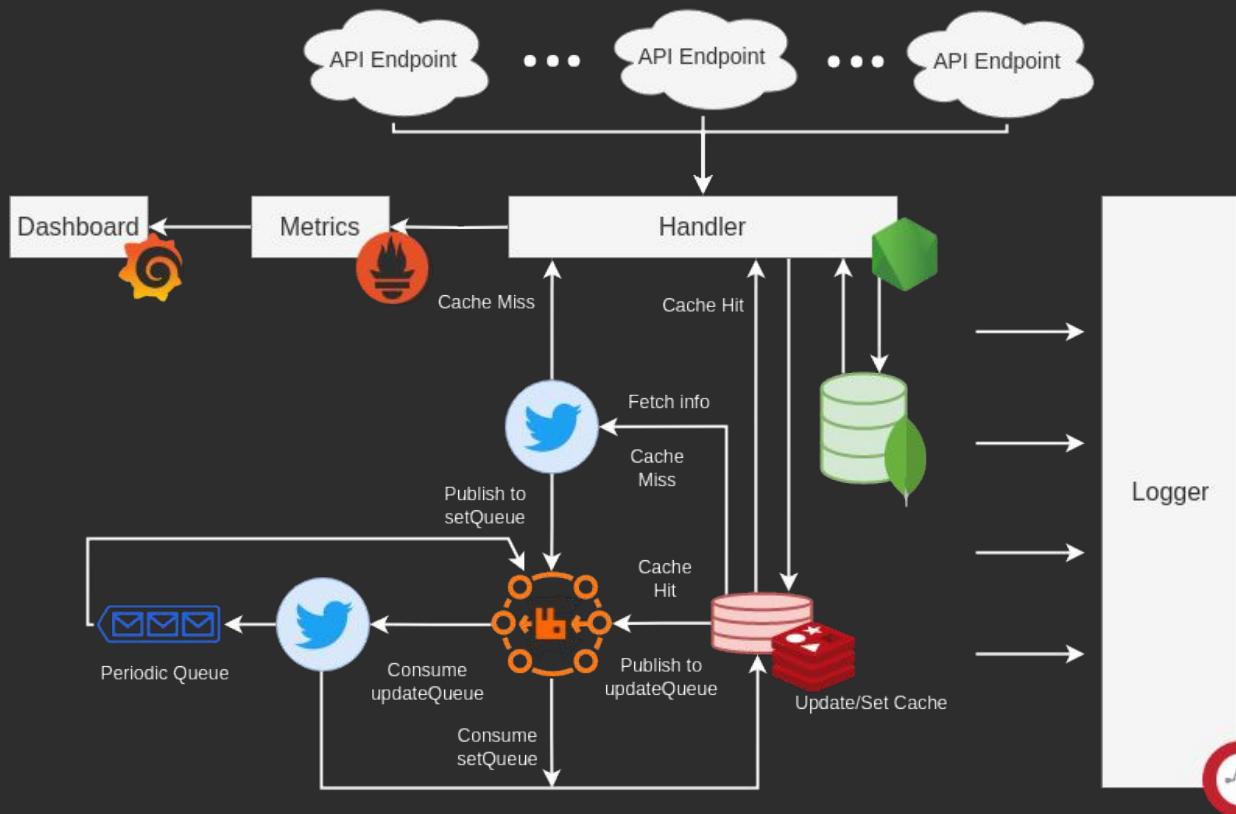
- GET /v1/users/{uuid} ▾
- POST /v1/users/{uuid} ▾
- PUT /v1/users/{uuid} ▾
- DELETE /v1/users/{uuid} ▾
- GET /v1/users/by/username/{username} ▾
- GET /v1/users/{uuid}/followers ▾
- GET /v1/users/{uuid}/following ▾

**Posts** Post ^

- GET /v1/posts/users/{uuid} ▾
- GET /v1/posts/{uuid}/timeline ▾
- GET /v1/posts/{id} ▾
- GET /v1/posts/{id}/replies ▾
- GET /v1/posts/{id}/liking\_users ▾



# Social API Architecture



+

# API Documentation

## MIXIT User Object

```
{  
    "uuid": "123456789123456780",  
    "name": "André Clérigo",  
    "username": "mrmaster",  
    "created_at": "2019-05-14T16:41:48.000Z",  
    "location": "Portugal",  
    "public_metrics": {  
        "followers_count": 144,  
        "following_count": 125  
    },  
    "external_information": {  
        "id": "1128339639209812000",  
        "twitter_name": "Andre",  
        "twitter_username": "mrmaster__"  
    }  
}
```

## Twitter User Object

```
{  
    "data": [  
        {  
            "id": "1128339639209812000",  
            "name": "André Clérigo",  
            "username": "mrmaster__",  
            "profile_image_url":  
                "https://pbs.twimg.com/profile_images/158670125447404339  
                4/1M2FGOL6_normal.jpg"  
        }  
    ],  
    "next_token": "49IA8JT6H4NHEZZZ"  
}
```



# API Documentation

## Tweet and Reply Object

```
{  
    "data": [  
        {  
            "id": "1598729618432807000",  
            "text": "Este é o meu primeiro tweet",  
            "conversation_id": "1598729618432807000",  
            "created_at": "2021-08-27T15:00:00.000Z",  
            "public_metrics": {  
                "retweet_count": 0, "reply_count": 0, "like_count": 0, "quote_count": 0  
            },  
            "author_info": {  
                "name": "André Clérigo",  
                "username": "mrmaster__",  
                "profile_image_url":  
                    "https://pbs.twimg.com/profile_images/1586701254474043394/lM2FGOL6_normal.jpg"  
            },  
            "photo_urls": [  
                "https://pbs.twimg.com/media/EtZ2Z7tXMAA5Z5_.jpg"  
            ],  
            "referenced_tweet": {  
                "data": [ <ORIGINAL_TWEET_OBJECT> ]  
            }  
        }  
    ],  
    "next_token": "49IA8JT6H4NHEZZZ"  
}
```



# API Documentation

## Error Handling

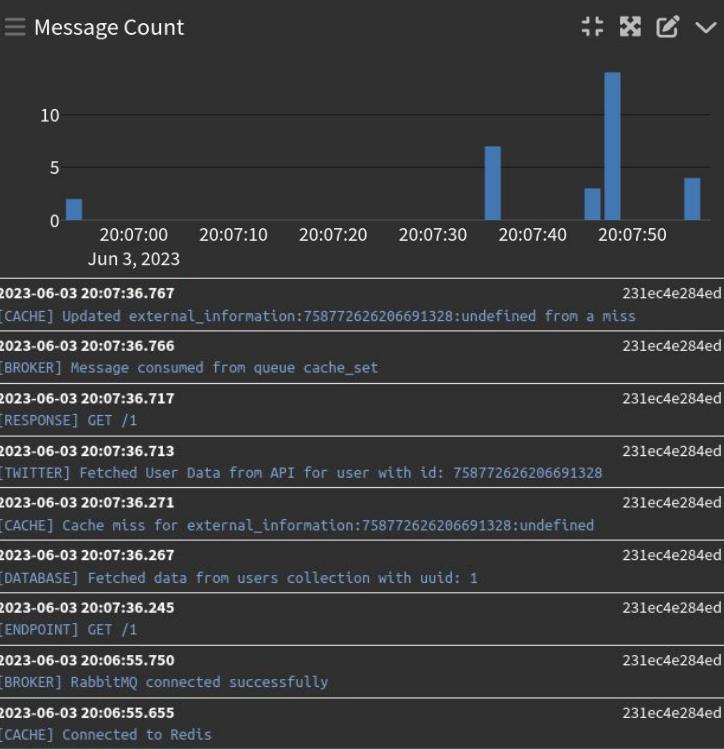
Error types:

- Invalid Parameter Error (**Code: 400**)
- Not Found Error (**Code: 404**)
- Duplicate Resource Error (**Code: 409**)

```
{  
  "errors": [  
    {  
      "parameters": {  
        "value": [  
          "<VALUE>"  
        ]  
      }  
    }  
  ],  
  "detail": "Could not find <RESOURCE_TYPE> with <PARAMETER>: [<VALUE>].",  
  "title": "<ERROR_TYPE>",  
  "parameter": "uuid",  
  "resource_type": "user"  
}
```



# API Logging



### App Response

2023-06-03 22:06:21.021	caa0fa2a13
[RESPONSE] GET /1665064427870093314/replies	
df36b241-025a-11ee-a275-02420a8b3b09	Permalink Show surrounding messages ▾ Test against
<b>Timestamp</b> 2023-06-03 22:06:21.021	<b>env</b> development
<b>Received by</b> MixITI-API on <a href="#">P a6fd8dcf / server</a>	<b>level</b> 6
<b>Stored in index</b> graylog_0	<b>message</b> [RESPONSE] GET /1665064427870093314/replies
<b>Routed into streams</b> <ul style="list-style-type: none"><li>Default Stream</li></ul>	<b>source</b> caa0fa2a1393
	<b>status_code</b> 200
	<b>time_elapsed</b> 505.384 ms
	<b>timestamp</b> 2023-06-03 22:06:21.021

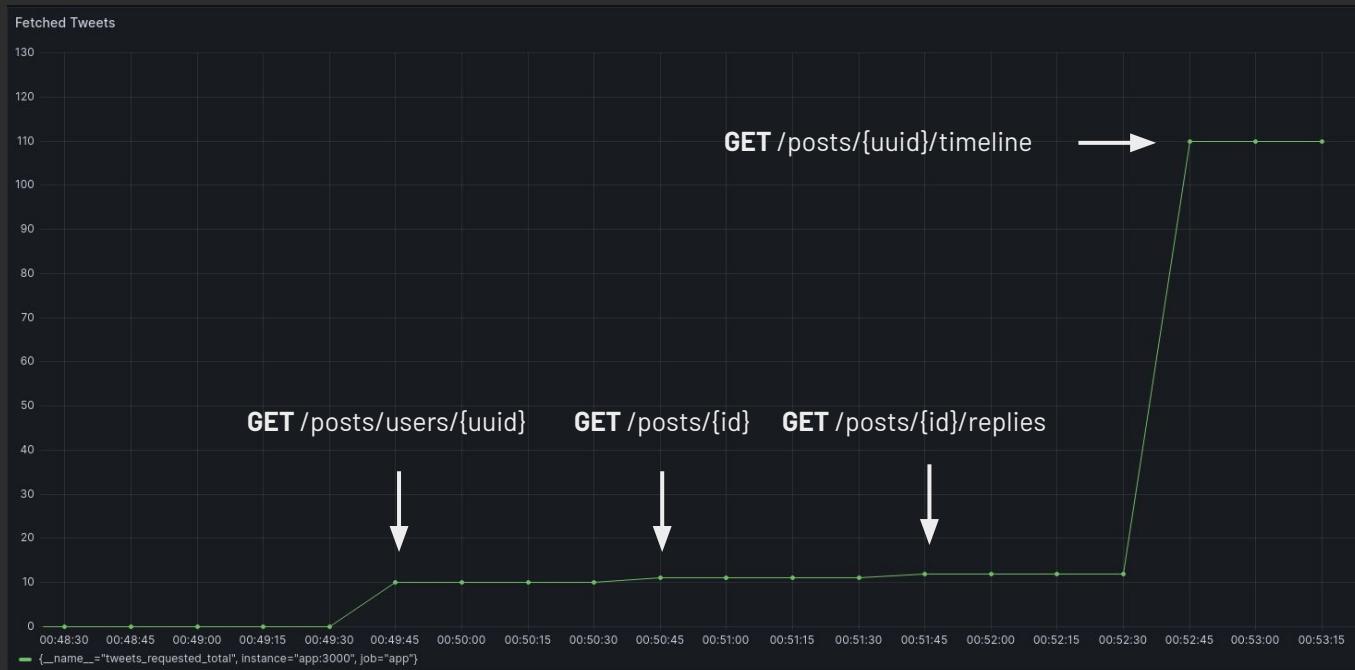
### Handler Error

2023-06-03 22:06:40.171	caa0fa2a1393
[HANDLER] No replies found for conversation_id: [166506442787009331]	
ea13650-025a-11ee-a275-02420a8b3b09	Permalink Show surrounding messages ▾ Test against
<b>Timestamp</b> 2023-06-03 22:06:40.171	<b>env</b> development
<b>Received by</b> MixITI-API on <a href="#">P a6fd8dcf / server</a>	<b>ip</b> ::ffff:10.139.59.1
<b>Stored in index</b> graylog_0	<b>level</b> 3
<b>Routed into streams</b> <ul style="list-style-type: none"><li>Default Stream</li></ul>	<b>message</b> [HANDLER] No replies found for conversation_id: [166506442787009331]
	<b>params_id</b> 166506442787009331
	<b>source</b> caa0fa2a1393
	<b>timestamp</b> 2023-06-03 22:06:40.171

# API Metrics



The only metric that this API tracks is the number of Tweets fetched from Twitter API.



# API Performance

## Apache Benchmark 10000 requests

Time taken: 70.244 sec

Requests per sec: 14.24 #/sec

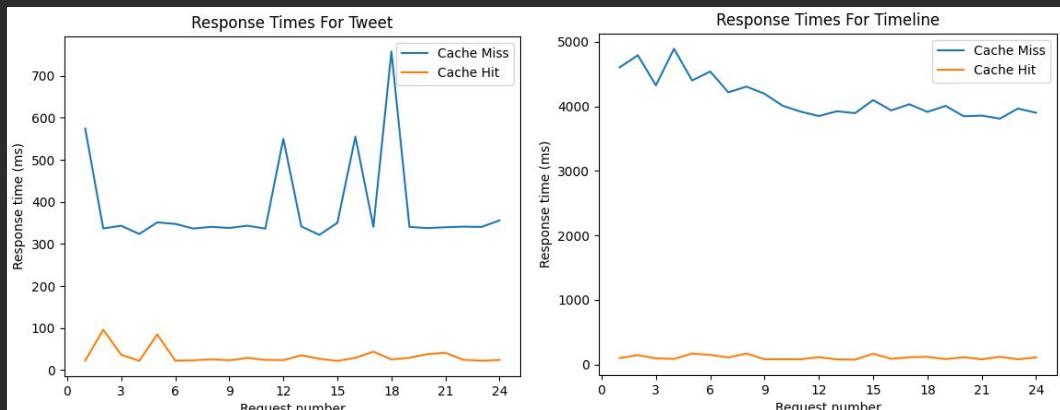
Time per request: 702.442 ms

### Connection Times (ms)

	min	mean	median	max
Connect:	2	8	4	130
Processing:	200	650	458	4744
Waiting:	187	627	435	4714
Total:	206	658	464	4746

### % of requests within a certain time (ms)

50%	464
66%	526
75%	704
80%	877
90%	1451
95%	1627
99%	1887
100%	4746 (longest request)



Cache Miss - ± 450ms  
Cache Hit - ± 40ms

Cache Miss - ± 4100ms  
Cache Hit - ± 200ms



# Authentication Service

---

MIXIT

+

# Main objectives



**Compliance with regulations**

**Ease of Integration**

**Performance**



# Authentication API

**Self authentication**



**Google authentication**



**Github authentication**





# Authentication API Overview

## External authentication



## Register

**register**

POST

/registersubmit

## Login

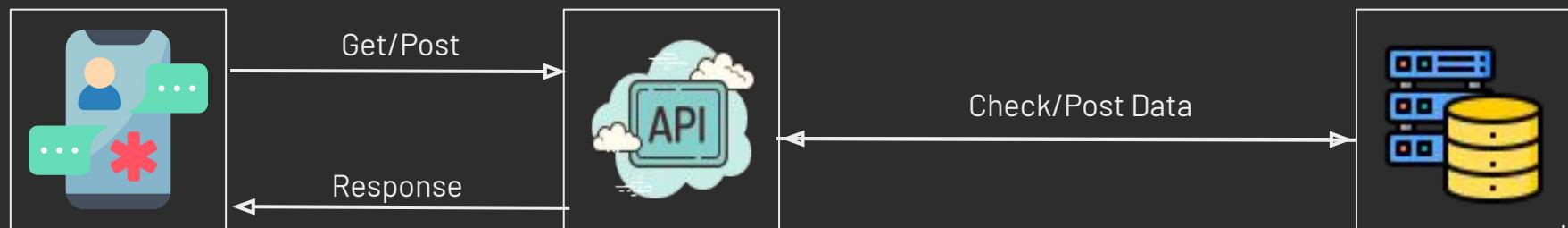
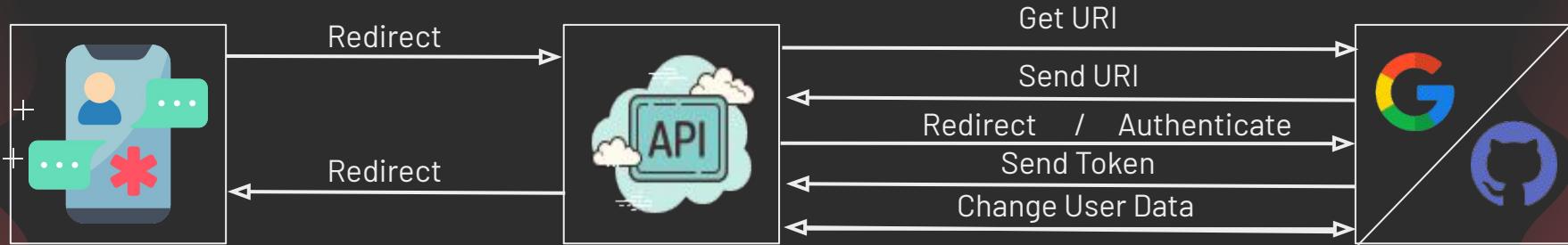
**login**

GET

/loginsubmit



# API Flow



# Composer

---

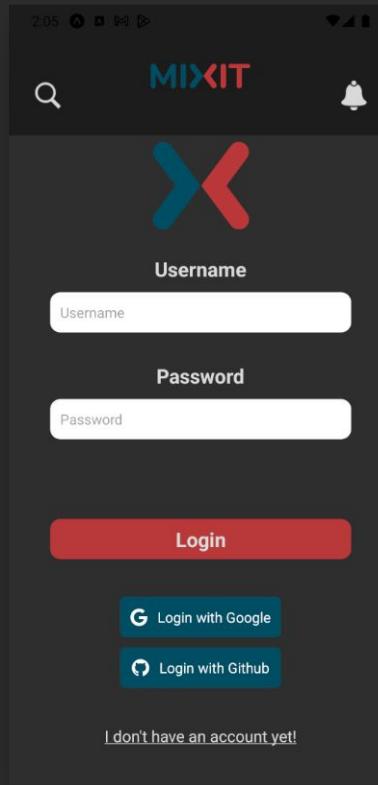
MIXIT

+

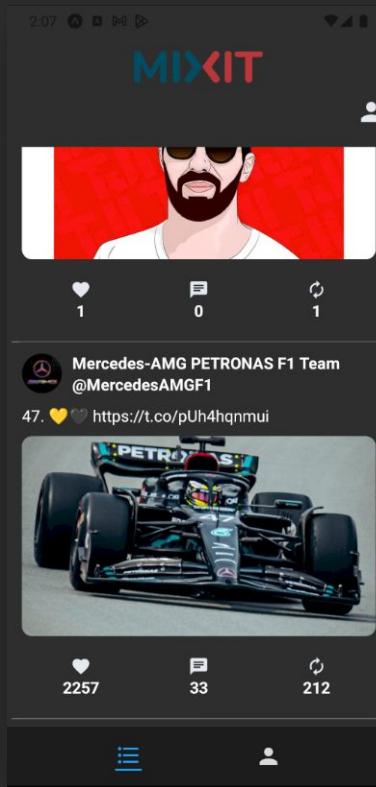
# Advertisements Service Integration



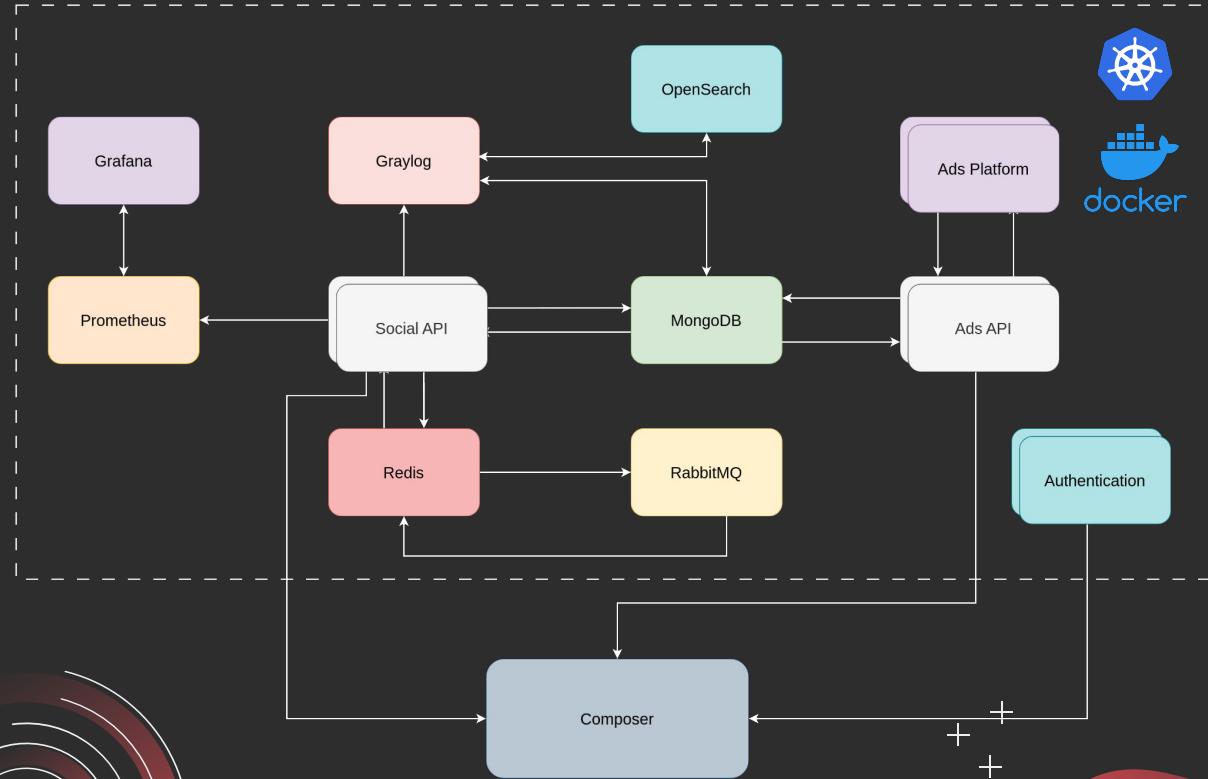
# Authentication Service Integration



# Social API Service Integration



# DEPLOYMENT



# MIXIT DEMO

Preview



# Future Work

01

## Twitter Authentication

Use Twitter API to do App authentication. With that done we are able to modify content from a Twitter user (POST, PUT, DELETE).

02

## Advertisement Videos

Add the functionality to display advertisements as videos

03

## YouTube Integration

Integrate youtube videos/shorts on the timeline



04

## Messaging Service

Simple and functional messaging services between our users

# Conclusions

- Successful integration of three **independent services** into one cohesive application.
- **Composer**, built with React Native, harmoniously unifies the services.
- Utilized **Service Composition** to successfully integrate three independent services.

# Github

<https://github.com/PedroRocha9/EGS>