Backend Case Study - Storyly

STORYLY

What is Storyly

Since Snapchat, Instagram, and Facebook introduced stories a few years ago, there has been an enormous hype both on mobile web, and finally, Storyly makes this a reality for iOS or Android apps.

User engagement on mobile apps is a challenge. Keeping your hardly acquired users in your app is the toughest problem. Storyly is a lightweight SDK giving the power of a popular story format to your mobile app, where you can showcase your content in the most fashionable way. You can make use of stories' bite-sized structure for content consumption only, or you can set up direct deals and monetize uniquely and stylishly. Using Storyly's dashboard, you can import stories from your social media accounts, create stories from images and videos and track all the crucial metrics for your story performance, helping you pick the best performers on the go.

You can see Storyly in action in below apps;

Kahve Dunyasi
https://play.google.com/store/apps/details?id=com.kahvedunyasi.app
https://apps.apple.com/us/app/kahve-d%C3%BCnyas%C4%B1-%C3%A7ekirdek-kazan/id1364478029

Tarif Kupu
https://apps.apple.com/tr/app/tarif-k%C3%BCp%C3%BC-yemek-tarifleri/id497267505
https://play.google.com/store/apps/details?id=air.com.ikikirpi.tarifkupu

Backend Case Study - Storyly

Glossary

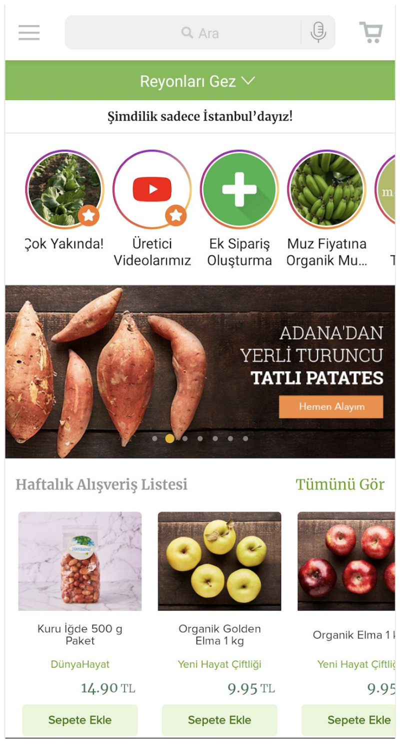
Mobile application (App): Customer's mobile application (Tarif Kupu, Kahve Dunyasi, GlowRoad)
Mobile user: Users of apps (Tarif Kupu's users)
App token: It is an access token specific to the app (you can see an example in the data fixtures)
Story metadata: Metadata that is necessary for the SDK to show stories to app users. It has an ID and data parts (you can see an example in the data fixtures). Every app has different stories.
Impression event: It is an event sent from SDK whenever a mobile user opens a story (How Storyly Works gives more information)
Close event: It is an event sent from SDK whenever a mobile user closes a story (How Storyly Works gives more information)
Event: Impression/Close are not the only events that our SDK sends. Event is a common name for all types of events.
Daily active users (DAU): Total number of mobile users of an app on a specific date. We need to get an event from SDK, impression/close, or any other one, to count that mobile user as an active user.

How Storyly Works

In a nutshell, this is how Storyly works;

1) Mobile user opens an app that has Storyly SDK in it
2) SDK sends a GET request to our servers to get story metadata (first/second assignment)
3) After getting a successful response from the servers, SDK shows story circles to the mobile user (you can see an example on the next page)
4) Whenever a mobile user taps/clicks on a story our SDK do the following
    a) Opens the story on full-screen mode
    b) Sends an impression event to our servers (third assignment)
5) Whenever a mobile user taps/clicks on the close button our SDK do the following
    a) Closes the story and goes back to the app screen
    b) Sends a close event to our servers (third assignment)

Backend Case Study - Storyly



Data Fixtures

app.csv:

story_metadata.csv:

General Requirements

- You can use any programming language you want (Python or Go is preferred)
- Please explain all necessary steps to run projects in a README file

Backend Case Study - Storyly

First Assignment - Story Metadata API

In this assignment we want you to write an HTTP API that takes app token as a parameter and returns stories in JSON format. You can find the necessary data fixtures attached, please store them in a relational database and serve from there.

Example Request: http:www.example.com/stories/{app_token}
Example Response:

```
{
  "app_id": 1, // Table ID of the app that makes the request
  "ts": 1601510400,  // Current timestamp
  "metadata": [
    {
      "id": 1,
      "metadata": "image1.png"
    },
    {
      "id": 2,
      "metadata": "image2.png"
    },
    {
      "id": 3,
      "metadata": "image3.png"
    }
  ]
}
```

Second Assignment - Enhanced Story Metadata API

Storyly is doing really well as a business and each day new customers are coming onboard. Our customer success team sends a meeting request to us and they want to talk about the complaints they receive from customers. Apparently many customers started to complain about the story load times recently and they want us to improve it.

We want you to improve your code/implementation/architecture you did in the first assignment to have better response times. You can use any technology you want.

Also, we want you to use Apache ab benchmarking tool to compare your first implementation and the one after improvements in terms of load and stress test (some important metrics; average response time, maximum concurrent limit). Please send requests from your local computer using ab tool and use maximum concurrency that your local environment allows.

Please share your ab command you used with the performance test results with us.

Third Assignment - Story Metric API

In this assignment we want you to write another HTTP API that takes app token and event details as parameters. We want you to aggregate those coming events in a relational database based on app, story, date, event type. An example table could be

app_id	story_id	date	type	count
1	1	2020-10-10	impression	100

You can use the same application you developed for the first and second assignments.

Example Request: http:www.example.com/event/{app_token}
Request Post Data

```
{
  "story_id": 1,
  “event_type”: “impression”
  “user_id”: “some-random-id” // Not necessary for this assignment, will be used for the bonus one
}
```

Bonus Assignment - Calculating DAU

Note: You DO NOT have to do this assignment.

Our users want to see their daily active users for a specific day so we want an API that takes app token and date as input and will return daily active users of the given app on the given date.

Hint: A single user can send multiple events per day and there can be millions of events for an app on a single day. Performance is a big concern here so even an algorithm/design that gives a close estimation will do the job.