**LEXA API**

Lexa API is a RestAPI.

This API Provides Database for Ecommerce, ~~Video Streaming and Music Application(under Progress)~~.

This API also has GraphQL based API endpoint’s which are used for message services!!

This API has Two Sides to It:

* Dev-side
* Client-side

API is built using NodeJS and MongoDB.

API Features in Brief:

* User Authorization.
* Products Querying with No reference.
* Categories Query.
* Product with In-depth Detail.
* Messaging with customer.

TECHNOLOGY STACK

Factory Pattern is used to design the API

This whole api (except user-endpoints) makes use of single factory handler which will handle all CRUD operations.

To keep the user routes secured and user’s data encrypted Users endpoint used User Handler where CRUD related to users are handled.

Google Storage service (Google Buckets) is used for storing the images/video/files.

Backend server automatically creates a new bucket to save different files in its own bucket.

Ex: if mobiles related images has to be stored – the backend is designed in a way that it takes bucket-name by the developer at the time of consuming the route and create a new bucket, where all the current and next upload related to mobiles will be stored in one place.

Json Web Token is used to store the user Auth details

Only user\_id is used as payload here to avoid data-theft.

If Api needs to be used for keep-signed-in option, this api also provides that

option. Which will be explained in detail below.

MongoDB with mongoose driver is used to query the documents and mutate the document.

Google Cloud App Engine is used to serve the deployed code – which stores backend code in encrypted way.

Complete Api is written with ES6+ features using Babel.

Every secured are passed through Protect middleware which will check the user entire details and gives access.

CORS is attached to the API.

To avoid user Route being exposed – used routes are passed through X-Rate-limiter.

And Other best practices of securing Api.

ENDPOINT – DETAILES

USER:

SIGNUP

* [https://lexa-api.uc.r.appspot.com/api/v1/signup](https://lexa-api.uc.r.appspot.com)
* The Above Endpoint takes
  + Username
  + Email
  + Password
  + Confirm-Password
  + Phone Number
  + Here Phone Number, Email are Unique Fields.
  + And Confirm Password will not be stored in database.
  + The password is encrypted with bcrypt package.
  + Token Is Created Using Json Wen Token Package.

SIGNIN

* [https://lexa-api.uc.r.appspot.com/api/v1/signin](https://lexa-api.uc.r.appspot.com/)
* The Above Endpoint Takes
  + Email
  + Password
  + API checks for email existence, if No user is found with the provided email ID. API send an error response saying “NO User Found with Given Email”.
  + Once the Email is verified API checks for Password By comparing DB stored password. If Password is wrong API sends Error response saying “Entered Password is Wrong”.
  + Once all Checks are done User gets new Token with his Payload

JWT AUTH Settings

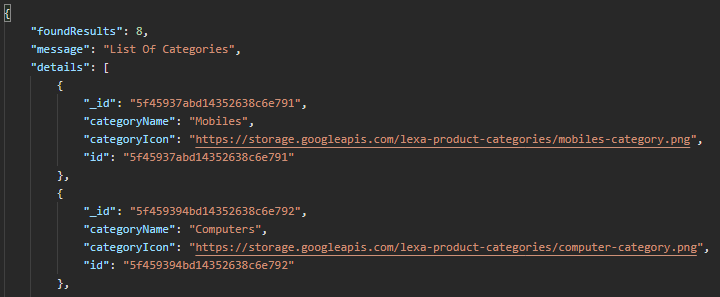
* User is by default signed for 24h
* If User chooses keep Signed in Token Will be valid forever

ACCREDITED / AUTHORIZED

* [https://lexa-api.uc.r.appspot.com/api/v1/accredit](https://lexa-api.uc.r.appspot.com)
* This Route Check User Auth Status / logged In Status through cookies
* Here This Route is Purposely designed only for React!
* React App has to get cookie which is the name of auth\_token!
* This is the most secured route as it handles auth of complete website
* There are three checks
  + JWT token is verified, checked for modifications, and expiry check
  + Once the above checks are passed it is passed to API’s protect route!
  + API’s protect route checks for bearer token and keeps the auth status for all API’s secured routes

PRODUCT ENDPOINTS:

Get All Product Categories (Method: GET)

* API Endpoint: [https://lexa-api.uc.r.appspot.com/api/v1/getAllCategories](https://lexa-api.uc.r.appspot.com)
* This endpoint fetches all products categories!
* Product categories contains
  + Category Name
  + Category Icon
  + ID
* Example: 

GET ALL PRODUCTS (Method: GET)

* [https://lexa-api.uc.r.appspot.com](https://lexa-api.uc.r.appspot.com/product/getProducts)[/product/getProducts](https://lexa-api.uc.r.appspot.com)
* Gets all the products by uploaded order.
* This route may not be that much useful that much, but this route has chained queries attached to it!!!
* Example



Chained Methods OF GET Products (Method: GET)

* API ENDPOINTS :<https://lexaapi.uc.r.appspot.com/> ---- chains?
* Get products by category type(takes: categoryId) : [/product/getProducts?categoryId=5f45937abd14352638c6e791](https://lexaapi.uc.r.appspot.com/product/getProducts?categoryId=5f45937abd14352638c6e791)
* Get product by Exact ProductName (rare use case) : [/product/getProducts?productName=Apple iPhone 11 (64GB) – Black](https://lexaapi.uc.r.appspot.com/)
* Get product by Search Term(takes : fieldname/search Term): </product/getProducts?searchin=productName&searchTerm=z>
* Any methods can be chained to one another!!!
* Additional Query methods available:
  + Pagination: ?page=1
  + Fields: fields=ProductName , Id, etc...,
  + Pagination with Results restriction: ?page=1&limit=10

Get Details About Products in cart (METHOD: POST)

* [https://lexaapi.uc.r.appspot.com/api/v1/](https://lexaapi.uc.r.appspot.com/api/v1/product/getProductsDetailsInCart)[product/getProductsDetailsInCart](https://lexaapi.uc.r.appspot.com)
* This Routes Doesn’t loop through list and reduce fetch time instead it uses mongo DB's $in method with collection to find the list of products with there unique Ids.
* This route takes exact cartsItems as req.body’s key.
* Example



GET Product by Id or Individual (METHOD: GET)

* Gets Complete detail about the product through virtual fields.
* This route gets different sets of data through reference.
* Example





* This method also fetches advertisement banner if available!