edu-stack-api

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Path | Request type | Body params | Headers | Response |
| Course |  |  |  |  |
| ES/api/courses | GET | NONE | NONE | Course objects without lessons field sorted by rating |
| ES/api/courses/free | GET | NONE | NONE | free Course objects without lessons field sorted by rating |
| ES/api/courses/category/:category | GET | NONE | NONE | Course objects in that category without lessons field sorted by rating |
| ES/api/courses/:id | GET | NONE | x-token | Single course with lessons if user has paid it or the course is free |
| ES/api/courses | POST | title, detail, previewImage, cost, categories  (\*not any lessons here) | x-token | Newly created course object |
| ES/api/courses/:courseID | PATCH | title, detail, previewImage, cost, categories,  creator: { \_id, …}  (\*not any lessons here) | x-token | Updated course object |
| ES/api/courses/:courseID | DELETE | creator: {\_id,…} | x-token | Success/error message |
| ES/api/courses/pushlesson/:courseID | POST | index (starting from 0), lesson: { whatever you want } , creator: {\_id, …} | x-token | Updated course object |
| ES/api/courses/deletelesson/:courseID | POST | index (starting from 0), creator object | x-token | Updated course object |
| ES/api/courses/updatelesson/:courseID | PATCH | index, lesson (new one to update the old one), creator object | x-token | Updated course object |
| Disclaimer\*\*\* | Some routes | May change in near future | Because there is no file deletion | In s3 in current version!!! |
| Login and verification |  |  |  |  |
| \*\*\* es/api/login | POST | email, password | NONE | User obj with x-token if verified else without x-token  (x-token doesn’t come when registered) |
| \*\*\* es/api/login/verify | POST | \_id (current user \_id), vToken, password | NONE | User obj if vToken is correct otherwise error message (if verification fails 3 times, the account gets deleted) |
| News |  |  |  |  |
| \*\*\*es/api/news | POST | title, description, organization, category, imageUrl, isImportant, detail | x-token | Created news object  (first send request to /upload route) |
| \*\*\*es/api/news/:newsID | DELETE | News object or just imageUrl | x-token | Success/error message |
| Upload |  |  |  |  |
| \*\*\*Es/api/upload | POST | route(avatar, news or resource as a string),  type (pdf, docs..),  contentType (image/png …) | x-token | url (to upload the file),  key (actual file name) |
| \*\*\* amazon url | PUT | Videoni ko’r |  |  |
| User profile | | | | |
| ES/api/me | Get | NONE | x-token | user object |
| \*\*\*ES/api/me/changePhoto | Patch | newAvatar,  oldAvatar | x-token  content-type: application/json | Header:  new x-token  Body: new user obj  (first send req to /upload route) |
| ES/api/me/changePassword | Patch | oldPassword  newPassword | x-token | user obj |
| ES/api/me/changeInfo | Patch | firstName  lastName  address  occupation  organization | x-token | New User obj  x-token |
| ES/api/me/myquestions | GET | NONE | x-token | Question array |
| ES/api/me/myresources | GET | NONE | x-token | Resource array |
| ES/api/me/myblogs | GET | NONE | x-token | Blog array |
| ES/api/me/mysurveys | GET | NONE | x-token | Survey array |
| ES/api/me/wishlist | POST | wishList object | x-token | New user object |
| ES/api/me/wishlist | PATCH | wishList object (must be the same with posted wishlist) | x-token | New user object  (\*this deletes wishlist item) |
| ES/api/me/cartlist | POST | cartList object | x-token | New user object |
| ES/api/me/cartlist | PATCH | cartList object  (must be the same as posted) | x-token | New user object  (\*this deletes cartList item) |
| ES/api/me/follow | POST | follower: {  \_id,  fullName,  avatar  }  following: {  \_id,  fullName,  avatar  },  action  (action can either be ‘follow’ or ‘unfollow’) | x-token | Success/err |
| Getting user |  |  |  |  |
| ES/api/users/all | GET | NONE | x-token | Array of all user objects |
| ES/api/users/:id | GET | NONE | x-token | Single user object |
| ES/api/users/notification | POST | receiver (\_id only, not object),  sender: {  \_id, fullName, avatar  },  message ( as you wish) | x-token | Success/Err |
| Resources |  |  |  |  |
| ES/api/resources | GET | NONE | x-token | Array of all available resources |
| \*\*\*ES/api/resources | POST | title  description  resourceType  costType  cost  file: { fileName, fileType, fileSize } | x-token  content-type: application/json | Newly created resource object  (first send request to /upload route) |
| ES/api/resources/:id | PATCH | title  description  resourceType  costType  cost | x-token  content-type: application/json | Updated resource object |
| ES/api/resources/file/:id | PATCH | file: { fileName, fileSize, fileType },  oldFileName (just name itself) | x-token  content-type: application/json | Updated resource object |
| ES/api/resources/:id | DELETE | file ( as an object with fileName and other fields, don’t forget this) | x-token | Success/error message |
| ES/api/resources/rate/:id | PATCH | rating | x-token | Rated resource object |
| ES/api/resources/rating/:userId/:resourceId | GET | NONE | x-token | Json {rating: user rating} |
| ES/api/resources/download/:file  \*(file = file name only, without uploads/resources/ ) | GET | fileType  title | x-token | File downloading begins |
| Blogs |  |  |  |  |
| ES/api/blogs | GET | NONE | NONE | Top 50 blogs |
| ES/api/blogs/:id | GET | NONE | NONE | Single blog object |
| ES/api/blogs/category/:category | GET | NONE | NONE | All blog objects in that category sorted by like |
| ES/api/blogs | POST | title,  description,  content,  categories | x-token | success / message(error) |
| ES/api/blogs/:id | PATCH | title,  description,  content,  categories | x-token | success / message(error) |
| ES/api/blogs/:id | DELETE | NONE | x-token | success / message(error) |
| ES/api/blogs/like/:id | POST | creator object | x-token | success / message(error) |
| Surveys |  |  |  |  |
| ES/api/surveys/ | GET | NONE | NONE | All surveys sorted by responseCount |
| ES/api/surveys/:id | GET | NONE | NONE | Single survey |
| ES/api/surveys/ | POST | title,  description,  inputFields,  isPrivate | x-token | Success/Error |
| ES/api/surveys/answer/:id | POST | answer ( all of them as an object array or whatever you like) | x-token | Success/error |
| ES/api/surveys/answer/:id | GET | creator | x-token | Single survey with answers |
| ES/api/surveys/toggle/:id | PATCH | creator | x-token | Success/error |
| ES/api/surveys/:id | DELETE | creator | x-token | Success/error |
| Exercises |  |  |  |  |
| ES/api/exercises | GET | NONE | x-token | All exercises without inputFields (sorted by date) |
| ES/api/exercises/:id | PATCH | title, description, cost, inputFields, time, isPrivate, category, type | x-token | Success/err |
| ES/api/exercises/:id | DELETE | creator: {  \_id  } | x-token | Success/Err |
| ES/api/exercises/answer/:userId | POST | \_id (exercise id),  title,  score | x-token | Success/Err |
| ES/api/exercises/free | GET | NONE | x-token | All free exercises without inputFields (sorted by date) |
| ES/api/exercises/:id | GET | NONE | x-token | Single exercise with inputFields |
| ES/api/exercises/category/:category | GET | NONE | x-token | All exercises in that category without inputFields (sorted by date) |
| ES/api/exercises | POST | title, description, cost, inputFields, time, isPrivate, category, type  ( \*category – [ielts, dtm …], type – [aniq, ijtimoiy, reading …] | x-token | Success/err |