TABSERA - REST API DOCUMENTATION v1.0

**DEPENDENCIES**

* "body-parser": "^1.18.2",
* "cookie-parser": "^1.4.3",
* "cors": "^2.8.4",
* "debug": "^2.2.0",
* "express": "^4.14.1",
* "express-fileupload": "^0.3.0",
* "express-session": "^1.15.6",
* "iconv-lite": "^0.4.19",
* "jade": "^1.11.0",
* "jsonwebtoken": "^8.1.0",
* "md5": "^2.2.1",
* "method-override": "^2.3.10",
* "morgan": "^1.7.0",
* "mysql": "^2.15.0",
* "node-session-tokens": "^0.1.6",
* "nodemailer": "^4.4.2",
* "pug": "^2.0.0-rc.4",
* "serve-favicon": "^2.3.2",
* "token-session": "^1.0.0"

**CONFIG.JS**

* **Description**

Here we declare a secret keyword ‘tabsera’ that is used for token generation.

**DbConnection.js**

* **Description**

Here we define the database host, username, password, and database in order to connect to database.

**API’s / Endpoints to be Consumed :**

Here is the API documentation of TABSERA , below api have been tested and documented according to our company standard document .

1. **Language Types**

* **Description**:

This will Return the type of languages which user wants to select , this will circulate in whole application

* **Url**:

/languages

* **Method**:

GET,GET/:id

* **Url Params**:

Normal : none , specifically language : id

* **Data Params**:

none

* **Required Fields**:

N/A

* **Success Response**:

[{"languageid":1,"name":"English","nativename":"English","flagurl":"https://www.countries-ofthe-world.com/flags-normal/flag-of-United-States-of-America.png"},{"languageid":2,"name":"Arabic","nativename":"عربي","flagurl":"https://www.countries-ofthe-world.com/flags-normal/flag-of-Saudi-Arabia.png"}]

* **Error Responses**:  
  N/A

1. **User Types**

* **Description**:

This will Return the type of user , for registration page , which type of user is needed .

* **Url**:

/usertypes

* **Method**:

GET

* **Url Params**:

NONE

* **Data Params**:
* **Required Fields**:

N/A

* **Success Response**:

[{"usertypeid":1,"description":"Student"},{"usertypeid":2,"description":"Parent"},{"usertypeid":3,"description":"Teacher"},{"usertypeid":4,"description":"Reseller"},{"usertypeid":5,"description":"Developer"},{"usertypeid":6,"description":"Admin"}]

* **Error Responses**:  
  N/A

1. **Get Schools**

* **Description**:

Return list of schools needed for registration page.

* **Url**:

/schools

* **Method**:

GET

* **Url Params**:

none

* **Data Params**:

none

* **Required Fields**:

none

* **Success Response**:

[{"schoolid":1,"fullname":"Maple Tree School","cityid":2,"address":null,"lat":null,"lng":null},{"schoolid":2,"fullname":"Beaconhouse","cityid":1,"address":"E11","lat":null,"lng":null},{"schoolid":3,"fullname":"F G School","cityid":2,"address":"F-11","lat":null,"lng":null},{"schoolid":4,"fullname":"Frobels International","cityid":2,"address":"E-11","lat":null,"lng":null}]

* **Error Responses**:

{"err”: {

"success": false,

"message": "User not found with the userId: 205"

}}

1. **Get Cities**

* **Description**:

Return list of cities.

* **Url**:

/cities

* **Method**:

GET

* **Url Params**:

none

* **Data Params**:

none

* **Required Fields**:

none

* **Success Response**:

[{"cityid":1,"fullname":"Rawalpindi","nativename":"","countryid":1},{"cityid":2,"fullname":"Islamabad","nativename":null,"countryid":1}]

* **Error Responses**:

{"err”: {

"success": false,

"message": "User not found with the userId: 205"

}}

1. **Register Student User**

* **Description**:

Register a user with required parameters & gaves succuess in return

* **Url**:

/students

* **Method**:

POST

* **Headers**

Access-Control-Allow-Origin:\*

enctype:multipart/form-data

Accept:application/json

x-access-token:

* **Url Params**:
* **Data Params**:

{"mobileno":"2155843", "email":"osamaabid03@gmail.com" , "username":"shami1231", "usertypeid":"1" , "password":"123456", "languageid":"1" , "deviceid":"1", "firstname":"Osama", "lastname":"Abid" ,"cityid" :"1" }

**Required**

* **Success Response**:

{

"success": true,

"message": "User Registered successfully"

}

* **Error Response**:

{"err”: {

"success": false,

"message": "Not able to update user status"

}

1. **Login User**

* **Description**:

Login User with Token in return of success response!

* **Url**:

/authenticate

* **Method**:

POST

* **Url Params**:
* **Data Params**:

{"username":"2155843", "password":"osamaabid03@gmail.com" , "deviceid ":"shami1231", "loginip":"1" }

**Required**

username,password,

**Success Response**:

{

"success": true,

"userid": 20,

"message": "Enjoy your token!",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIyMzA2Nn0.-buUC7Tohh-DHnwv7zoYYnHrAuim\_j6R\_d5MXGuuWGI"

}

* **Error Response**:

{"err”: {

"success": false,

"message": "Not able to update user status"

}

1. **Send email verification**

This module is working, with the registration module (Integrated)

1. **Change Password**

* **Description**:

End point for changing password

* **Url**:

/users/editPassword/:userid

* **Method**:

PUT

* **Url Params**:

/:userid

* **Data Params**:

{ "password": value , "token": value }}

**Required**

“password” is required

“token” is required

* **Success Response**:

{

"success": true,

"message": "Password updated Successfuly"

}

* **Error Response**:

1. **Change Mobile No**

* **Description**:

API for changing mobile number in the settings screen

* **Url**:

/users/editMobileno/:userid

* **Method**:

PUT

* **Url Params**:

/:userid

* **Data Params**:

{ "mobileno": value , "token": value }

**Required**

“mobile” is required

“token” is required

* **Success Response**:

{

"success": true,

"message": "Password updated Successfuly"

}

* **Error Response**:

{

"success": false,

"message": "Mobile No already Exists"

}

**Design Screen No** :10.3 -

1. **Change Student Name**

* **Description**:

Student can change the name from settings screen

* **Url**:

/students/editName/:stdid

* **Method**:

PUT

* **Url Params**:

/:stdid

* **Data Params**:

{ "fname": value , "lname" : value , "token": value }

**Required**

“fname” is required

“lname” is required

“token” is required

**Design Screen No** :10.3 -

* **Success Response**:

{"success": true,"message": "Name updated Successfuly"}

* **Error Response**:

1. **Change Parent Name**

* **Description**:

Parent can change the name from settings screen

* **Url**:

/parents/editName/:parentid,

* **Method**:

PUT

* **Url Params**:

/:parentid

* **Data Params**:

{ "fname": value , "lname" : value , "token": value }

**Required**

“fname” is required

“lname” is required

“token” is required

**Design Screen No** :10.3 -

* **Success Response**:

{"success": true,"message": "Name updated Successfuly"}

* **Error Response**:

1. **Register Parent**

* **Description**:

API endpoint for parents registration

* **Url**:

/parents

* **Headers**:

Access-Control-Allow-Origin:\*

enctype:multipart/form-data

Accept:application/json

x-access-token:

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"mobileno":"2155843", "email":"osamaabid03@gmail.com" , "username":"abid", "usertypeid":"1" , "password":"123456", "languageid":"1" , "deviceid":"1", "firstname":"Osama", "lastname":"Abid" , "mode" : "email" ,"cityid" :"1" }

**Required**

mobileno, email,username, usertypeid, password, languageid, deviceid,firstname, lastname, dob, resellerid, cityid

* **Success Response**:

{"success": true,"message": "Parent Registered Successfuly!"}

* **Error Response**:

{

"success": false,

"message": "user already exists. "

}

1. **User Login**

* **Description**:

API endpoint for user login

* **Url**:

/authenticate

* **Method**:

POST

* **Url Params**:

NONE

* **Data Params**:

{"username":"dexter", "password":"123456"}

**Required**

“username” is required and its value is username of the user trying to login

“password” is required and its value is password of the user trying to login

* **Success Response**:

{

"success": true,

"userid": 20,

"message": "Enjoy your token!",

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIzMTg4M30.GCO3b7rDi\_s56cxN5A3HcS4b0z0ogv-Thf1ICZehOz4"

}

* **Error Response**:

{

"success": false,

"message": "Authentication failed. "

}

1. **Get Subjects/Grades**

* **Description**:

API endpoint for getting Subjects of units , via subject id

* **Url**:

/subjects,/subjects/grade/:gradeid

**Method**:

GET,GET/:id

* **Url Params**:

NONE

* **Data Params**:
* **Success Response**:

[{"subjectid":1,"gradeid":1,"languageid":1,"name":"Mathematics","description":null,"imageurl":null,"isenabled":{"type":"Buffer","data":[1]}},{"subjectid":2,"gradeid":1,"languageid":2,"name":"English","description":null,"imageurl":null,"isenabled":{"type":"Buffer","data":[1]}}]

]

* **Error Response**:

1. **Get Subjects/Grades**

* **Description**:

API endpoint for getting Subjects of units , via subject id

* **Url**:

/subjects,/subjects/grade/:gradeid

**Method**:

GET,GET/:id

* **Url Params**:

NONE

* **Data Params**:
* **Success Response**:

[{"subjectid":1,"gradeid":1,"languageid":1,"name":"Mathematics","description":null,"imageurl":null,"isenabled":{"type":"Buffer","data":[1]}},{"subjectid":2,"gradeid":1,"languageid":2,"name":"English","description":null,"imageurl":null,"isenabled":{"type":"Buffer","data":[1]}}]

]

* **Error Response**:

1. **Get Units**

* **Description**:

API endpoint for getting Units , also single unit can be accessed!

* **Url**:

/units

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"unitid":2,"subjectid":1,"developerid":2,"name":"Aljebra","description":null,"unitprice":5,"isdemo":{"type":"Buffer","data":[0]},"isenabled":{"type":"Buffer","data":[1]}}]

* **Error Response**:
* **Design Screen No** : 7

1. **Get Transactions Types :**

* **Description**:

API endpoint for getting Type Of transactions for Payment

* **Url**:

/units

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"trantypeid":0,"description":"reversal"},{"trantypeid":1,"description":"credit purchase"},{"trantypeid":2,"description":"unit enrollment"},{"trantypeid":3,"description":"reseller credit purchase"},{"trantypeid":4,"description":"tution enrollment"},{"trantypeid":5,"description":"teacher tution credit"},{"trantypeid":6,"description":"teacher tution payment"},{"trantypeid":7,"description":"subscription enrollment"}]

* **Error Response**:
* **Design Screen No** :

1. **Enroll Units**

* **Description**:

API endpoint for getting Units enrolled for the user , !

* **Url**:

/unitenrollments

**Method**:

POST

* **Url Params**:

none

* **Data Params**:

{userid": 8 , "unitid" : 2 , "tranid": 4 , "unitcost" : 200 , "enrollmethodid" : 3 , "token" : "" }

**Success Response**:

{"success": true,"message": "Unit Enrolled Successfuly!"}

* **Error Response**:
* **Design Screen No** : 7

1. **Enroll Units Of user**

* **Description**:

API endpoint for getting Units enrolled for the user , !

* **Url**:

/unitenrollments/userid

**Method**:

POST

* **Url Params**:

Token , id

* **Data Params**:
* **Success Response**:

ERORR

* **Error Response**:
* **Design Screen No** : 7

1. **Get Lessons:**

* **Description**:

API endpoint for getting lessons , with specific excersice , which provide us the lessons

* **Url**:

/exercises/lesson/:lessonid,

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"lessionid":2,"unitid":3,"description":"","lessonminutes":15,"lessonurl":"https://www.travel.com/","isenabled":{"type":"Buffer","data":[1]}}]

* **Error Response**:
* **Design Screen No** : 8.1.6 -

1. **Get Enrolled Lessons of User :**

* **Description**:

API endpoint for getting enrolled lessons for specific user , which he have enrolled

* **Url**:

/lessonenrolls/:userid

**Method**:

GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"userid":1,"lessonid":2,"secondsviewed":5,"iscompleted":{"type":"Buffer","data":[1]},"isdownloaded":{"type":"Buffer","data":[1]},"filepassword":"dummy","fileurl":"https://www.travel.com","enrolldate":"2018-01-02T00:27:03.000Z"}]

* **Error Response**:
* **Design Screen No** :

1. **Get Excersices :**

* **Description**:

API endpoint for getting Excersices of lessons

* **Url**:

/exercises/lesson/:lessonid,

/exercises/:exerciseid

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"exerciseid":1,"unitid":2,"lessionid":1,"description":"PHP Exercise","exerciseminutes":15,"exerciseurl":"https://medlineplus.gov/","isenabled":{"type":"Buffer","data":[1]}}]

* **Error Response**:
* **Design Screen No** : 8.1.5 - / 9

1. **Get Enrolled Excersices :**

* **Description**:

API endpoint for getting Excersices of lessons which are enrolled by the user

* **Url**:

/exerciseenrolls/:userid

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"tranid":7,"reversedtranid":null,"trantypeid":1,"trandate":"2017-12-29T02:01:15.000Z","transtatusid":1,"paymentmethodid":3,"description":"Adding Amount","userid":4,"credit":0,"debit":50},{"tranid":8,"reversedtranid":null,"trantypeid":4,"trandate":"2017-12-29T02:02:41.000Z","transtatusid":1,"paymentmethodid":3,"description":"Purchase Lesson","userid":4,"credit":15,"debit":0}]

* **Error Response**:
* **Design Screen No** : 8.1.5 - / 9

1. **Submit Exercises results :**

* **Description**:

API endpoint for saving result of exercise which is been done

* **Url**:

/exerciseresults

**Method**:

POST

* **Url Params**:

token

* **Data Params**:
* {"userid": 2 , "exerciseid" : 1 , "questionid" : 3 , "iscorrect" : 1 , "token" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjI5MTI2MH0.IGdKCSLfO5SNoqvZ2XSvMawcUFlVK-2Lq8ikoKGE-fA"}
* **Success Response**:

{"success": true,"message": "Exercise Result added Successfuly!"}

1. **Get Excersices result :**

* **Description**:

API endpoint for saved result getting, to preview

* **Url**:

/exerciseresults/userid/:userid/exerciseid/:exerciseid

**Method**:

GET : GET:id

* **Url Params**:

Token , userid , exerciseid

* **Data Params**:

none

* **Success Response**:

[{"userid":2,"exerciseid":1,"questionid":1,"iscorrect":{"type":"Buffer","data":[1]}},{"userid":2,"exerciseid":1,"questionid":2,"iscorrect":{"type":"Buffer","data":[0]}}]

1. **Get Payment Methods :**

* **Description**:

API endpoint for getting Payment Methods for purchasing Units

* **Url**:

/paymentmethods,

/paymentmethods/:paymntid

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"paymentmethodid":1,"description":"paypal"},{"paymentmethodid":2,"description":"airtime"},{"paymentmethodid":3,"description":"card"},{"paymentmethodid":4,"description":"voucher"},{"paymentmethodid":5,"description":"waafipay"}]

* **Error Response**:
* **Design Screen No** :8.1.2 -

1. **E-wallet Balance Page :**

* **Description**:

API endpoint for getting Details for user balance , to see how much balance is in wallet of user

* **Url**:

/transactions/userid/:userid

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"tranid":7,"reversedtranid":null,"trantypeid":1,"trandate":"2017-12-29T02:01:15.000Z","transtatusid":1,"paymentmethodid":3,"description":"Adding Amount","userid":4,"credit":0,"debit":50},{"tranid":8,"reversedtranid":null,"trantypeid":4,"trandate":"2017-12-29T02:02:41.000Z","transtatusid":1,"paymentmethodid":3,"description":"Purchase Lesson","userid":4,"credit":15,"debit":0}]

* **Error Response**:
* **Design Screen No** :11.1.1 -

1. **Search Reseller :**

* **Description**:

API endpoint for searching reseller in list to purchase credit

* **Url**:

/resellerusers

**Method**:

GET,GET/:id

* **Url Params**:

token

* **Data Params**:

none

* **Success Response**:

[{"tranid":7,"reversedtranid":null,"trantypeid":1,"trandate":"2017-12-29T02:01:15.000Z","transtatusid":1,"paymentmethodid":3,"description":"Adding Amount","userid":4,"credit":0,"debit":50},{"tranid":8,"reversedtranid":null,"trantypeid":4,"trandate":"2017-12-29T02:02:41.000Z","transtatusid":1,"paymentmethodid":3,"description":"Purchase Lesson","userid":4,"credit":15,"debit":0}]

* **Error Response**:
* **Design Screen No** :11.1.1 -

1. **Add new Reseller :**

* **Description**:

API endpoint for searching reseller in list to purchase credit

* **Url**:

/resellerusers

**Method**:

Post

* **Url Params**:

token

* **Data Params**:
* **Success Response**:

[{"tranid":7,"reversedtranid":null,"trantypeid":1,"trandate":"2017-12-29T02:01:15.000Z","transtatusid":1,"paymentmethodid":3,"description":"Adding Amount","userid":4,"credit":0,"debit":50},{"tranid":8,"reversedtranid":null,"trantypeid":4,"trandate":"2017-12-29T02:02:41.000Z","transtatusid":1,"paymentmethodid":3,"description":"Purchase Lesson","userid":4,"credit":15,"debit":0}]

* **Error Response**:
* **ERROR FOUND**

1. **Check Amount / Balance from Reseller :**

* **Description**:

API endpoint for checking in e wallet of reseller that entered amount is available or not , for buying credit.

* **Url**:

/resellers/check

**Method**:

Post

* **Url Params**:

token

* **Data Params**:

{"resellerid": 1 , "amount" : 200 }

* **Success Response**:

{"success": true,"message": "Credit Exist!"}

{"success": false,"message": "Credit is low!"}

* **Error Response**:

**Invalid route**

1. **Parents Purchace unit/course :**

* **Description**:

API endpoint for parents to purchase units / courses for their kids

* **Url**:

/parnt/purchaseTrans

**Method**:

Post

* **Url Params**:

token

* **Data Params**:

{"reversedtranid": 1 , "trantypeid" : 1 , "paymentmethodid" : 4 , "description" : "abcd" , "userid": 1 , "amount" : 30 , "token" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIzMTg4M30.GCO3b7rDi\_s56cxN5A3HcS4b0z0ogv-Thf1ICZehOz4"}

* **Success Response**:

{

"success": true,

"message": "Transaction done Successfuly"

}

1. **Parents Purchace Balance / Amount :**

* **Description**:

API endpoint for parents to purchase units / courses for their kids

* **Url**:

/parnt/purchaseTrans

**Method**:

Post

* **Url Params**:

token

* **Data Params**:

{"reversedtranid": 1 , "trantypeid" : 1 , "paymentmethodid" : 4 , "description" : "abcd" , "userid": 1 , "amount" : 30 , "token" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIzMTg4M30.GCO3b7rDi\_s56cxN5A3HcS4b0z0ogv-Thf1ICZehOz4"}

* **Success Response**:

{

"success": true,

"message": "Transaction done Successfuly"

}

1. **Student Purchace unit/ course:**

* **Description**:

API endpoint for student to purchase units / courses for their kids

* **Url**:

/std/purchaseTrans

**Method**:

Post

* **Url Params**:

token

* **Data Params**:

{"reversedtranid": 1 , "trantypeid" : 1 , "paymentmethodid" : 4 , "description" : "abcd" , "userid": 1 , "amount" : 30 , "token" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIzMTg4M30.GCO3b7rDi\_s56cxN5A3HcS4b0z0ogv-Thf1ICZehOz4"}

* **Success Response**:

{

"success": true,

"message": "Transaction done Successfuly"

}

1. **Student Purchace Balance / Amount :**

* **Description**:

API endpoint for student to purchase units / courses for their kids

* **Url**:

/std/purchaseTrans

**Method**:

Post

* **Url Params**:

token

* **Data Params**:

{"reversedtranid": 1 , "trantypeid" : 1 , "paymentmethodid" : 4 , "description" : "abcd" , "userid": 1 , "amount" : 30 , "token" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIzMTg4M30.GCO3b7rDi\_s56cxN5A3HcS4b0z0ogv-Thf1ICZehOz4"}

* **Success Response**:

{

"success": true,

"message": "Transaction done Successfuly"

}

36. **Get Transaction been done:**

* **Description**:

API endpoint for user to get single transaction

* **Url**:

/transactions/tranid/:tranid

**Method**:

Post

* **Url Params**:

Token , id

* **Data Params**:

[{"tranid":8,"reversedtranid":null,"trantypeid":4,"trandate":"2017-12-29T02:02:41.000Z","transtatusid":1,"paymentmethodid":3,"description":"Purchase Lesson","userid":4,"credit":15,"debit":0}]

* **Success Response**:

{

"success": true,

"message": "Transaction done Successfuly"

}

**37. Student Purchace Balance / Amount :**

* **Description**:

API endpoint for student to purchase units / courses for their kids

* **Url**:

/std/purchaseTrans

**Method**:

Post

* **Url Params**:

token

* **Data Params**:

{"reversedtranid": 1 , "trantypeid" : 1 , "paymentmethodid" : 4 , "description" : "abcd" , "userid": 1 , "amount" : 30 , "token" : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbiI6ImRleHRlciIsImlhdCI6MTUyMjIzMTg4M30.GCO3b7rDi\_s56cxN5A3HcS4b0z0ogv-Thf1ICZehOz4"}

* **Success Response**:

{

"success": true,

"message": "Transaction done Successfuly"

}

38. **LOG OUT:**

* **Description**:

API endpoint for user to log out

* **Url**:

/users/logout/:userid

**Method**:

PUT

* **Url Params**:

Token , id

* **Data Params**:

none

* **Success Response**:

{"success": true,"message": "User logged out Successfuly"}

39. PayVia PayPal etc (Payment Gateway)

* **Description**:

API endpoint for payment method i.e. PayPal, end point will get the id of payment Method and trigger the system which is been called via id.

* **Url**:

/stdewallet/pay/:id

**Method**:

POST

* **Url Params**:

Token

* **Data Params**:

none

* **Success Response**:

Required Payment Method Preview

40. List of Students for search (Associating Child Search)

* **Description**:

API endpoint is for searching child, this endpoint lists us all child , so we can search them while associating to the parents.

* **Url**:

/students/childs

**Method**:

GET

* **Url Params**:

Token

* **Data Params**:

none

* **Success Response**:

List of child with id , full name along.

41. List of Parents for search (Associating Child Search)

* **Description**:

API endpoint is for searching parent, this endpoint lists us all parents, so we can search them while associating to the parents.

* **Url**:

/parents/list

**Method**:

GET

* **Url Params**:

Token

* **Data Params**:

none

* **Success Response**:

List of parents with id , full name along.

42. Associate Parent to the child

* **Description**:

API endpoint is for associating parent to the student/ child ,

* **Url**:

/students/assignParent/:id

**Method**:

PATCH

* **Url Params**:

Token , student id ,

* **Data Params**:

{"parentuserid": value} – must be a number

* **Success Response**:

{"message": "Parrent Associated Successfuly" }

* **Error**

43. Associate Child to the Parent

* **Description**:

API endpoint is for associating child to the parent ,

* **Url**:

/parents/assignChild/:id

**Method**:

PATCH

* **Url Params**:

Token , parent id ,

* **Data Params**:

{"childID":value } – must be a number

* **Success Response**:

{"message": "Child Associated Successfully”}

**Error**

44. Add New School

* **Description**:

API endpoint is for adding school to the list if not available ,

* **Url**:

/schools/addSchool

**Method**:

POST

* **Url Params**:

Token

* **Data Params**:

{"fullname":"TMUC" , "cityid":2 , "address":"Islamabad" , “lat” : 63.00 , “lng”: 72.00}  
fullname – string/char

Address -- string

cityId - number

lat, lng – double/float/number

* **Success Response**:

{"message": "Child Associated Successfully”}

45. Get Parent Profile :

* **Description**:

API endpoint is for getting profile data of the parent user.

* **Url**:

/parents/profile/:id

**Method**:

GET

* **Url Params**:

Token , id

* **Data Params**:

none

* **Success Response**:

[

{

"userid": 1,

"firstname": "Ahmad",

"middlename": null,

"lastname": "Khan",

"gender": "M",

"dob": "2018-03-28T13:38:54.000Z",

"resellerid": 1,

"credit": 0,

"points": 0,

"cityid": 1

}

]

45. Get Student Profile :

* **Description**:

API endpoint is for getting profile data of the Student user.

* **Url**:

/students/profile/:id

**Method**:

GET

* **Url Params**:

Token , id

* **Data Params**:

none

* **Success Response**:

[

{

"userid": 2,

"firstname": "Ali",

"middlename": null,

"lastname": "Ahmad",

"gender": "M",

"dob": "2018-04-02T10:10:14.000Z",

"schoolid": 1,

"resellerid": null,

"parentuserid": 5,

"teacheruserid": null,

"credit": 85,

"points": 0,

"cityid": 1

}

]

s

46. Change Student Profile Image :

* **Description**:

API endpoint is for updating profile photo while signing up.

* **Url**:

/students/ changeDp/:id

**Method**:

PATCH

**HEADERS :**

//Content-Type:multipart/form-data

x-access-token:

enctype:multipart/formdata

Access-Control-Allow-Origin:\*

Accept:application/json

* **Url Params**:

Token , id

* **Data Params**:

Image / form data

* **Success Response**:

{

"success": true,

"message": "Profile photo updated Successfuly"

}

47. Change Parent Profile Image :

* **Description**:

API endpoint is for updating profile photo while signing up.

* **Url**:

/parents/changeDp/:id

* **Method**:

PATCH

* **HEADERS :**

//Content-Type:multipart/form-data

x-access-token:

enctype:multipart/formdata

Access-Control-Allow-Origin:\*

Accept:application/json

* **Url Params**:

Token , id

* **Data Params**:

Image / form data

* **Success Response**:

{

"success": true,

"message": "Profile photo updated Successfuly"

}

48. Get Total Answered Stats :

* **Description**:

API endpoint is for profile to show the stats of answers which user have done right or wrong!

* **Url**:

/answeredStat/:userid

* **Method**:

GET

* **HEADERS :**
* **Url Params**:

Token , id

* **Data Params**:

none

* **Success Response**:

{ "Correct Answers": 2, "Wrong Answers": 1, "Total Questions ": 3 }

49. Get Total Answered Stats Subject wise (Total Question Completed count) :

* **Description**:

API endpoint is for profile to show the stats of answers which user have done right or wrong but subject wise !

* **Url**:

/answeredStat/:userid?subjectid=1

* **Method**:

GET

* **HEADERS :**

x-access-token

* **Url Params**:

Userid , ?subjectid

* **Data Params**:

none

* **Success Response**:

{

"Res": [

{

"SubjectID": 1,

"UnitID": 2,

"LessonID ": 1,

"QuestionID ": 1,

"ExerciseID ": 1,

"Correct": true

}

],

"TotalResult": {

"Correct Answers": 2,

"Wrong Answers": 1,

"Total Questions ": 3

}

}

50 . Get Total Lessons Completed , Uncompleted , Total of User:

* **Description**:

API endpoint is for statistics , to see how many overall lessons are completed , not complted , remaining , and what is the total count of the subjects , of the user , in all subjects

* **Url**:

http://localhost:3000/subjects/lessonsStats/:userid

* **Method**:

GET

* **HEADERS :**

x-access-token

* **Url Params**:

user id

* **Data Params**:

none

* **Success Response**:

{

"Result": [

{

"SubjectID": 3,

"UnitID": 6,

"LessonId": 5,

"LessonCompleted": false

},

{

"SubjectID": 3,

"UnitID": 5,

"LessonId": 4,

"LessonCompleted": false

},

{

"SubjectID": 3,

"UnitID": 5,

"LessonId": 3,

"LessonCompleted": false

}

],

"TotalResult": {

"Complted": 0,

"UnComplete": 3

}

}

51 . Get Total Lessons Completed , Uncompleted , Total of User But Each Subject wise :

* **Description**:

API endpoint is for statistics , to see how many overall lessons are completed , not completed , remaining , and what is the total count of the subjects , of the user , but each subject

* **Url**:
* http://localhost:3000/subjects/lessonsStats/:userid?subjectid=id
* **Method**:

GET

* **HEADERS :**

x-access-token

* **Url Params**:

user id , subject id

* **Data Params**:

none

* **Success Response**:

Same as above Response API.No. 50

52 . Get Total Lessons Completed , Uncompleted , Total of User But Each Subject wise :

* **Description**:

API endpoint is for statistics to see total completion of subjects , with their percentages , lessons total result , exercises total result , subject , unit , ids etc in details .

* **Url**:

/subjects/SubjectProgressStats/:id

* **Method**:

GET

* **HEADERS :**

x-access-token

* **Url Params**:

user id

* **Data Params**:

none

* **Success Response**:

{

"Result": [

{

"SubjectID": 2,

"UnitID": 3,

"LessonId": 5,

"LessonCompleted": false,

"ExerciseID": 4,

"ExerciseCompleted": false

}

],

"LessonsTotalResult": {

"Complted": 0,

"UnComplete": 1,

"Completion Percentage ": 0

},

"ExerciseTotalResult": {

"Complted": 0,

"UnComplete": 1,

"Completion Percentage ": 0

}

}

53. SET Seconds/Minutes of Enrolled Lesson :

* **Description**:

API endpoint is for adding time been spent on the lesson enrolled by user.

* **Url**:

/lessonenrolls/addSeconds

* **Method**:

PUT

* **HEADERS :**

x-access-token

* **Url Params**:

user id , lessoned , secondsViewd (time)

* **Data Params**:

{"userid":20 , "lessonid": 5 , "secondsviewed" : 20}

* **Success Response**:

{ "success": true, "message": "Time Added!" }

54. Redeem Voucher

Api end point is for redeeming voucher for the reseller and consumed by the user which he receives from reseller

DEPRECIATED

54. Check Device Id & return the users been logged on it before

* **Description**:

API endpoint is for checking the device Unique id base search in user list in database ,that is been saved during user registration of the user , so if device id exist , api return the data like username and image to set recent login slider

* **Url**:

/deviceIdCheck/:id?

* **Method**:

GET

* **HEADERS :**

none

* **Url Params**:

user

* **Data Params**:

none

* **Success Response**:

[ { "userid": 29,"firstname": "Osama", "lastname": "Abid" , "username": "aqil222","email": "muhammad.aaqil@carinotech.com","image": null} ]

55. GET LIST OF ENROLLED UNITS OF USER

* **Description**

API endpoint is for checking the the units which are enrolled , so that we can able to show them status of unit on selecting

* **Url**:

/units/enrolledUnits/:id?

* **Method**:

GET

* **HEADERS :**

none

* **Url Params**:

userid

* **Data Params**:

none

* **Success Response**:

[ { "unitid": 29," "unitid": 28," "unitid": 30," "unitid": 31," } ]

56. GET LIST OF ENROLLED LESSONS OF UNIT OF USER

* **Description**

API endpoint is for checking the the lesson which are not enrolled , so that they will be locked on first sight.

* **Url**:

lessonenrolls/lessonsEnrollCheck/userid/:userid/unitid/:unitid

* **Method**:

GET

* **HEADERS :**

token

* **Url Params**:

Userid , unit id

* **Data Params**:

none

* **Success Response**:

[ { "lesson id": 29 } ]

**TABSERA API’s – Missing / Required that are not available in the Developed End Points**

* Adding New School in the database, list.
* Populating List of Parents for searching Parents Scenario
* Associating parents to the child profile (Registering Student Needs Parent ID, but that is only 1 times while registration only)
* Populating List of Students List for searching Students Scenario
* Associating child/Student to the parent.
* General rating / Analytics & Score point for profile.
* Get Detail Object of User (Child, Parent).
* Uploading Profile photo of student/parent/teacher.
* Get History of Transactions been done previously.
* Change Parent Profile Photo
* Change Student Profile Photo
* Get Count of Total Correct & Wrong answers with their relevant ids
* Get Count of Total Correct & Wrong answers subject wise