|  |  |  |
| --- | --- | --- |
| Description | Method + Path | Request / Response |
| Check if ID and Email not existing | POST  /validate | Request: {  Email: string  ID: string  }  Response: {  Error: false,  Message: "Continue to step 2"  } |
| Register user | POST  / | Request: {  first\_name,  last\_name,  ID,  city,  street,  email,  password,  },  Response: {  {  "error": false,  "registeredUser": {  "has\_access": boolean,  "role": string,  "\_id": string,  "first\_name": string,  "last\_name": string,  "ID": number,  "city": string,  "street": string,  "email": string,  "password": string  }  }  } |

-Register Route

Endpoint: http://localhost:5000/register

Login Route

Endpoint: <http://localhost:5000/login>

|  |  |  |
| --- | --- | --- |
| Description | Method/Path | Request / Response |
| Login User | POST  / | Request: {  Username: string,  Password: string  }  Response: {  {  "error": false,  "user": [  {  "has\_access": boolean,  "role": string,  "\_id": string,  "first\_name": string  "last\_name": string,  "email": string,  "ID": number,  "city": string,  "street": string  }  ],  "token": string  }  } |
| Disable or enable access to website when clicking on "Starting Shopping" | POST  /handle-access/:userId | Request: {  userId:string  }  Response: {  {  "has\_access": boolean,  "role": string,  "\_id": stirng,  "first\_name": string  "last\_name": string,  "email": string,  "ID": number,  "city": string,  "street": string  }  } |
| Decode existing token and grant access to the website while refreshing | GET  /refresh/:token | Request : {  Token:string  }  {  "error": false,  "data": {  "user": [  {  "has\_access": boolean,  "role": string,  "\_id": string ,  "first\_name": string,  "last\_name": string,  "email": string,  "ID": number,  "city": string,  "street": string  }  ],  "iat": number,  "exp": number  }  } |
| Get amount of products and orders created in the store | GET  /store-data | Request : {}  Response: {  {  "error": false,  "guitars": number,  "orders": number  }  } |

Guitars Route

Endpoint: <http://localhost:5000/guitars>

|  |  |  |
| --- | --- | --- |
| Description | Method + Path | Request / Response |
| Get all products in the store | GET  / | Request: {  }  Response: {  Error: false,  Guitars: [  {  "\_id": string,  "name": string,  "category": {  "\_id": string,  "name": string  },  "price": string,  "picture": string  },  ]  } |
| Get products by category | GET  /category/:id | Request : {  Id: string  }  Response: {  Error: false,  Guitars: [  {  "\_id": string,  "name": string,  "category": {  "\_id": string,  "name": string  },  "price": string,  "picture": string  },  ]  } |
| Get categories in store | GET  /categories | Request : {}  Response: {  [  {  “\_id”:string,  “name”:string  }[{ |
| Get products by search filter | GET  /filter | Request: {  Query: string  }  Response: {  Error: false,  Guitars: [  {  "\_id": string,  "name": string,  "category": {  "\_id": string,  "name": string  },  "price": string,  "picture": string  },  ]  } |

Cart Route

Endpoint: http://localhost:5000/cart

|  |  |  |
| --- | --- | --- |
| Description | Method + Path | Request / Response |
| Create a new cart, or return existing active cart. | POST  /newcart/:userId | Request{  userId: string  }  Response: {  {  "error": false,  "cart": {  "is\_active": true,  "\_id": string,  "buying\_user": string,  "date\_created": Date  }  }  } |
| Get existing cart | POST  /existing-cart/userId | Request {  userId: string  }  Response: {  {  "error": false,  "cart": {  "is\_active": true,  "\_id": string,  "buying\_user": string,  "date\_created": Date  }  }  } |
| Add product to cart | POST  /add/:userId/:productId | Request {  userId: string,  productid: string,  amount: string  },  Response {  {  "error": false,  "cartDetails": [  {  "\_id": string,  "guitar": {  "\_id": string,  "name": string,  "category":string,  "price": string,  "picture": string  },  "amount": number,  "price": number,  "cartID": string  }  ],  "totalPrice": number  }  } |
| Delete product from cart | DELETE  /delete/:userId/:productId | Request {  userId: string,  productid: string  }  Response {  {  "error": false,  "cartDetails": [  {  "\_id": string,  "guitar": {  "\_id": string,  "name": string,  "category": string,  "price": string,  "picture": string  },  "amount": number,  "price": number,  "cartID": string  }  ],  "totalPrice": number  }  } |
| Delete all products from cart | DELETE  /delete/cartID | Request {  cartID: string  }  Response {  {  "error": false,  "cart": [],  "totalPrice": 0  }  } |
| Get products in cart | GET  /products/:userId | Request {  userId: string  },  Response {  {  "products": [  {  "\_id": string,  "guitar": {  "\_id": string,  "name": string,  "picture": string  },  "amount": number  "price": number,  }  ]  } |

Order Route

Endpoint <http://localhost:5000/order>

|  |  |  |
| --- | --- | --- |
| Description | Method + Path | Request / Response |
| Create new order | POST  /purchase/:userId/:cartID | Request: {  userId: string,  cartID: string  },  Response: {  {  "cartProducts": [  {  "\_id": string,  "guitar": {  "\_id": string,  "name": string  "category": string,  "price": number,  "picture": string  },  "amount": number,  "price": number,  "cartID": string  }  ],  "newOrder": {  "\_id": string,  "user\_id": string,  "cart\_id": string,  "total\_price": number,  "delivery\_city": string,  "delivery\_street": string,  "delivery\_date": Date,  "card\_digits": number,  "date\_created": Date  }  }  } |
| Get latest action by user -  Active cart / Last order/ No actons at all | GET  latest-action/:buying\_user | Request {  Buying\_user: string  },  Response: {  Type: string,  message:string  } |

|  |  |  |
| --- | --- | --- |
| Description | Method + Path | Request / Response |
| Add a new product to the store | POST  /new-product | Request{  Name: string,  Category: string,  Price: number,  Picture: string  },  Response {  {  error: false,  message: "Product added successfully!"  }  } |
| Edit product by product id | PUT  /edit-product/productId | Request{  productid: string,  Name: string,  Category: string,  Price: number,  Picture: string  },  Response {  {  "error": false,  "product": {  "\_id": string,  "name": string,  "category": string,  "price": number,  "picture": string  }  }  } |

Admin route

Endpoint: http://localhost:5000/admin