

| | | | | |
|-----------------------------|--------|---|--|--|
| Airbnb | | | | |
| | | | | |
| Link | Method | Input | Output | Working |
| /auth/register | POST | name, email, password, profilePhoto | Save the user details in the database and navigate to "/login" | |
| /auth/login | POST | email, password | Check the user in the database, and navigate user to "homepage" | |
| /users/:userId/trips | GET | params: userId | It will show all the booked properties | The frontend requests a get method to the backend with userId. The backend searches if the userId matches any customerId in the Booking Model and then sends the data to the frontend. The frontend receives the data and dispatches it to the redux store. The components to show the listing takes the data from the redux store and show it in the UI. |
| /users/:userId/listingId | PATCH | params: userId, listingId | It will add and remove the properties from wishlist | The frontend requests a get method to the backend with userId and listingId. The backend searches in the wishlist array in the User Model that listingId is present in it or not. If it is present then it removes from it otherwise it adds to it. After that it saves the updated user in the database. In the frontend it receives the response and dispatches the wishlist with the new one. |
| /users/:userId/properties | GET | params: userId | It will show all the properties hosted by the user | The frontend requests a get method with sending userId in the params to the Backend. In the backend it searches the userId with the creatorId in the Listing Model and sends the data back to the frontend. The frontend then dispatches the data to the property list which updates the store. |
| /users/:userId/reservations | GET | params: userId | It will show list of the properties that you have hosted and others have booked. | The frontend requests a get method with sending userId in the params to the Backend. In the backend it searches the userId with the hostId in the Booking Model and sends the data back to the frontend. The frontend then dispatches the data to the reservation list which updates the store. |
| /properties/create | POST | creator, category, type, streetAddress, aptSuite, city, province, country, guestCount, bedroomCount, bedCount, bathroomCount, amenities, title, description, highlight, highlightDesc, price, listingPhotos | It will create a new property and will be saved in database. If response is ok then navigate to "/" page | All the information is sent from the frontend as a post request. In the backend all the data is saved to the database. If the response is ok then the user will be navigated to the "/" route |
| /properties | GET | query: "All" or any category like "windmill" or "barns" | It will show properties according to the Categories for eg. windmill, beachfront etc | Frontend sends get request to backend with query in its link. The query is got at the backend in query parameters. Now we will search all the listings in the listing model whose category is same as the query category. If no category is matched then it will show all the properties. It will send all those to frontend. In frontend all the data is dispatched to the listing in redux store. |
| /properties/search/:search | GET | params: category or title | It will show us all the properties related to what you have searched | The search params are sent from the frontend. The params data is searched in the Listing model. If the params data is "all" then all the data from the listing model is sent, else the category and title are searched in the Listing Model with regex and case insensitivity. All the data is then sent to frontend. The data from frontend is dispatched to listing field in redux store. |
| /properties/:listingId | GET | params: listingId | It will show all the properties listed or are there in the database. | Frontend sends a get request to backend with params in it. The data for that listingId is searched in the Listing model. All the data is then sent to frontend. The data from frontend is dispatched to listing field in redux store. |
| /bookings/create | POST | customerId, listingId, hostId, startDate, endDate, totalPrice | It will create a new booking and will save that in database. If response is ok then redirect to "/customerId/trips" | All the data is sent from frontend to backend in post request. In backend the newBooking is saved in database and data is sent to frontend. If response is ok then navigate user to "/customerId/trips" to see all his trips. |