



MeetMyRoute

By Theresa, Kim and Sofia



What does MeetMyRoute do?

MeetMyRoute is an application designed to bring together travellers with shared interests and similar itineraries to enhance their experience through companionship and local insights and encourage meaningful connections.

Key Features:

- Registration and Login
- Profile and Itinerary Management
- Search and Filter
- Social Connections



Example Code

Conditional Statements

If the logged in user ID does not match with the user ID from URL params, return error message of “User not authorised”

If user is not found, return error message of “User not found”

Database Operations

```
const updateProfile = async (req, res) => {
  try {
    // Get the user ID from the URL parameters
    const {id} = req.params;

    // Get data to update from the request body
    const {name, location, status, profilePic, travelPreferencesAndGoals, socialMediaLink} = req.body;

    // Get logged in user ID from the request
    const loggedInUserId = req.user.id;

    // Check if logged in user ID matches user ID from the URL parameters
    if (loggedInUserId !== id) {
      return res.status(403).json({
        message: "User not authorised"
      });
    }

    // Find user profile that belong to the user ID
    const user = await UserModel.findById(id);

    // Check if user ID exists
    if (!user) {
      return res.status(404).json({
        message: "User not found"
      });
    }

    // Update fields if new data is provided
    user.name = name || user.name;
    user.location = location || user.location;
    user.status = status || user.status;
    user.profilePic = profilePic || user.profilePic;
    user.travelPreferencesAndGoals = travelPreferencesAndGoals || user.travelPreferencesAndGoals;
    user.socialMediaLink = socialMediaLink || user.socialMediaLink;

    // Save updated user profile
    const updatedUser = await user.save();

    // Respond with the updated user profile
    return res.status(200).json({
      message: "User profile updated successfully",
      data: updatedUser
    });
  } catch (error) {
    res.status(500).json({
      message: "Error updated user profile:",
      error
    });
  }
}
```



Example Code

```
async function fetchAndSetProfileData(userId) {  
  try {  
    // Get the JWT token from local storage  
    const jwt = localStorage.getItem("jwt");  
  
    // Check if the JWT token exists  
    if (!jwt) {  
      throw new Error("Not authenticated. Please login");  
    }  
  
    // Get the profile data using the provided user ID  
    const response = await axios.get(`${API}/profile/${userId}`, {  
      headers: {  
        Authorization: `Bearer ${jwt}`  
      }  
    });  
  
    // Set the fetched profile data into the state  
    setProfileData(response.data.data);  
  } catch(error) {  
    setError(error.response?.data?.message || "Failed to fetch user data");  
  } finally {  
    setLoading(false);  
  }  
}
```

Network Request

Send a GET request with JWT token passed into the Authorization header to authenticate

```
if (loading) {  
  return <p>Loading...</p>  
}
```



Code Challenge

```
export function ProfileDataProvider({children}) {  
  const [profileData, setProfileData] = useState(initialiseProfileData);  
  // const user = useAuth();  
  const { userId } = useParams();  
  if (!user) {  
    return console.log("User not authenticated");  
  }  
  // Fetch the profile data from the API  
  async function fetchAndSetProfileData(userId) {  
    const API = import.meta.env.API_URL;  
    const url = `${API}/profile/${userId}`;  
    console.log("Requesting URL:", url);  
    try {  
      // const result = await axios.get(`${API}/profile/${userId}`);  
      const result = await axios.get(`${API}/profile/${userId}`, {  
        headers: {  
          // Authorization: `Bearer ${user.token}`  
        }  
      });  
      // Set the result profile data into the state  
      setProfileData(result);  
      console.log(result);  
    } catch (error) {  
      console.log("Error fetching user profile:", error);  
    }  
  };  
  // Fetch the profile data and set the result into the state when the user ID changes  
  useEffect(() => {  
    fetchAndSetProfileData(userId)  
  }, [userId])  
  return (  
    <ProfileDataContext.Provider value={profileData}>  
      <ProfileDataSetterContext.Provider value={setProfileData}>  
        {children}  
      </ProfileDataSetterContext.Provider>  
    </ProfileDataContext.Provider>  
  )  
}
```

Returning:

“Requesting URL: undefined/profile/undefined”



Code Solution

```
export function ProfileDataProvider({children}) {
  const [profileData, setProfileData] = useState(initialiseProfileData);
  // const user = useAuth();
  const { userId } = useParams();
  if (!user) {
    return console.log("User not authenticated");
  }
  // Fetch the profile data from the API
  async function fetchAndSetProfileData(userId) {
    const API = import.meta.env.API_URL;
    const url = `${API}/profile/${userId}`;
    console.log("Requesting URL:", url);
    try {
      // const result = await axios.get(`${API}/profile/${userId}`);
      const result = await axios.get(`${API}/profile/${userId}`, {
        headers: {
          // Authorization: `Bearer ${user.token}`
        }
      });
      // Set the result profile data into the state
      setProfileData(result);
      console.log(result);
    } catch (error) {
      console.log("Error fetching user profile:", error);
    }
  };
  // Fetch the profile data and set the result into the state when the user ID changes
  useEffect(() => {
    fetchAndSetProfileData(userId)
  }, [userId])
  return (
    <ProfileDataContext.Provider value={profileData}>
      <ProfileDataSetterContext.Provider value={setProfileData}>
        {children}
      </ProfileDataSetterContext.Provider>
    </ProfileDataContext.Provider>
  )
}
```

Returning:

"Requesting URL: undefined/profile/undefined"

stackoverflow About Products OverflowAI Search...

Home Questions Tags Users Companies

Enviroment variables in Vite + React coming back undefined

Asked 2 years, 5 months ago Modified 11 months ago Viewed 11k times

Sorted by: Highest score (default)

1 Answer

You need to prefix the name of your environment variables with `VITE_` if you want to expose the variables clientside.

8 Like this: `VITE_HOST` both in the `.env` -file and in the code. See the example below.

```
const API = import.meta.env.VITE_API_URL;
```

For API = undefined:
Prefix of VITE



Code Solution

```
export function ProfileDataProvider({children}) {  
  const [profileData, setProfileData] = useState(initialiseProfileData);  
  // const user = useAuth();  
  const { userId } = useParams();  
  if (!user) {  
    return console.log("User not authenticated");  
  }  
  // Fetch the profile data from the API  
  async function fetchAndSetProfileData(userId) {  
    const API = import.meta.env.API_URL;  
    const url = `${API}/profile/${userId}`;  
    console.log("Requesting URL:", url);  
    try {
```

Returning:

"Requesting URL: undefined/profile/undefined"

```
function App() {  
  return (  
    <div>  
      <Router>  
        <NavBar />  
        <Routes>  
          <Route path="/" element={<HomePage />} />  
          <Route path="/login" element={<LoginPage />} />  
          <Route path="/signup" element={<SignUpPage />} />  
          <Route path="/profile/:userId" element={<ProfilePage />} />  
          <Route path="/itinerary" element={<ItineraryPage />} />  
          <Route path="/search" element={<SearchPage />} />  
          <Route path="/connects" element={<ConnectsPage />} />  
          <Route path="/logout" element={<HomePage />} />  
          <Route path="/forgot-password" element={<ForgetPassword />} />  
          <Route path="/reset-password" element={<ForgetPassword />} />  
        </Routes>  
      </Router>  
      <Footer />  
    </div>  
  )  
}
```

```
createRoot(document.getElementById('root')).render(  
  <StrictMode>  
    <AuthProvider>  
      <ProfileDataProvider>  
        <App />  
      </ProfileDataProvider>  
    </AuthProvider>  
  </StrictMode>  
)
```

Why doesn't react-router-dom useParams hook work within Context?

Asked 2 years, 11 months ago Modified 2 years, 2 months ago Viewed 2k times

Your ProfileContextProvider is outside the route that defines the "id" param.... this cant work.

Share Improve this answer Follow

answered Oct 9, 2022 at 1:42



Thomas Decaux
22.6k • 2 • 125 • 136

```
};  
} catch (err) {  
  console.log(err);  
}  
// Fetch the profile data from the API  
useEffect(() => {  
  fetchAndSetProfileData(userId)  
}, [userId])  
return (  
  <ProfileDataContext.Provider value={profileData}>  
    <ProfileDataSetterContext.Provider value={setProfileData}>  
      {children}  
    </ProfileDataSetterContext.Provider>  
  </ProfileDataContext.Provider>  
)  
}
```




Code Solution

```
export default function ProfileContainer() {
  const [isEditing, setIsEditing] = useState(false);
  const [profileData, setProfileData] = useState({});
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState("");
  const { userId } = useParams();

  // Get userId from local storage
  const loggedInUserId = localStorage.getItem("userId");

  // Determine if this is the logged-in user's profile
  const isOwnProfile = loggedInUserId === userId;

  // Fetch and set profile data based on user ID
  async function fetchAndSetProfileData(userId) {
    try {
      // Get the JWT token from local storage
      const jwt = localStorage.getItem("jwt");

      // Check if the JWT token exists
      if (!jwt) {
        throw new Error("Not authenticated. Please login");
      }

      // Get the profile data using the provided user ID
      const response = await axios.get(`${API}/profile/${userId}`, {
        headers: {
          Authorization: `Bearer ${jwt}`
        }
      });

      // Set the fetched profile data into
      setProfileData(response.data.data);
    } catch (error) {
      setError(error.message);
    }
  }

  // Call the fetch function when the component mounts
  useEffect(() => {
    fetchAndSetProfileData(userId);
  }, [userId]);

  return (
    <div>
      <ProfileContainer />
    </div>
  );
}
```

For userId = undefined:

Deleted the Profile Data Context Provider and moved the location of the network request so it was within the scope of the useParams hook



Code Challenge

```
<label htmlFor="travelPreferencesAndGoals">Travel Preferences & Goals:</label>
{tempProfileData.travelPreferencesAndGoals?.map((item, index) => {
  return (
    <div key={index} className="travelPrefAndGoalsItem">
      <input
        type="text"
        defaultValue={item}
        onChange={(e) => handleChangeTravelPrefAndGoals(index, e.target.value)} />
      <button
        type="button"
        className="removeTravelPrefAndGoalsButton"
        onClick={() => handleRemoveTravelPrefAndGoals(index)}>Remove</button>
      </div>
    );
  )}
</button>
type="button"
className="addTravelPrefAndGoalsButton"
onClick={handleAddTravelPrefAndGoals}>Add</button>
```

In the ProfileForm component, the Travel Preferences & Goals values were not updating correctly when an item was removed (when it was not the last item on the list)



Code Solution



```
▼ (4) ['2', '4', '5', '7'] ⓘ  
  0: "2"  
  1: "4"  
  2: "5"  
  3: "7"  
  length: 4  
  ► [[Prototype]]: Array(0)  
▼ (3) ['4', '5', '7'] ⓘ  
  0: "4"  
  1: "5"  
  2: "7"  
  length: 3  
  ► [[Prototype]]: Array(0)
```

The values on the form were not updating as expected although the values were being updated correctly on the console.log



Code Solution

2

Remove

stackoverflow

Home

Questions

Tags

Remove

7

Remove

About Products OverflowAI Search...

Why isn't my input value updating with React?

Asked 8 years, 8 months ago Modified 1 year ago Viewed 80k times

That doesn't mean that your input is being unmounted and then a new input is being rendered, though. `defaultValue` can't be updated after an initial render. – TAGraves Apr 18, 2016 at 12:12

```
▼ (4) ['2', '4', '5', '7'] ⓘ  
  0: "2"  
  1: "4"  
  2: "5"  
  3: "7"  
  length: 4  
  ► [[Prototype]]: Array(0)  
▼ (3) ['4', '5', '7'] ⓘ  
  0: "4"  
  1: "5"  
  2: "7"  
  length: 3
```

The values on the form were not updating as expected although the values were being updated correctly on the console.log



Code Solution

```
<label htmlFor="travelPreferencesAndGoals">Travel Preferences & Goals:</label>
{tempProfileData.travelPreferencesAndGoals?.map((item, index) => {
  return (
    <div key={index} className="travelPrefAndGoalsItem">
      <input
        type="text"
        value={item}
        onChange={(e) => handleChangeTravelPrefAndGoals(index, e.target.value)} />
      <button
        type="button"
        className="removeTravelPrefAndGoalsButton"
        onClick={() => handleRemoveTravelPrefAndGoals(index)}>Remove</button>
      </div>
    );
  )}
</button>
type="button"
className="addTravelPrefAndGoalsButton"
onClick={handleAddTravelPrefAndGoals}>Add</button>
```

Updated defaultValue to value

Thank You!