

Axios Auth Debug Checklist

1 Confirm Local Storage

- Check that the user object is actually **stored and parsed correctly**.
- Log **both raw and parsed** versions:

```
console.log(localStorage.getItem("user"));
console.log(JSON.parse(localStorage.getItem("user")));
```

- Ensure the `accessToken` exists **exactly where you expect** (`user.accessToken`, not `user.token`).

2 Axios Instance Headers

- Ensure the interceptor adds the **Authorization header**:

```
if (user && user.accessToken) {
  config.headers.Authorization = "Bearer " + user.accessToken;
}
```

- Log **before sending**:

```
console.log("Headers before request:", config.headers);
```

3 Correct API Call

- GET/POST doesn't automatically inherit JSON/multipart headers unless specified.
- If you use `multipart/form-data` for POST, **don't use it for GET**. Keep GET simple:

```
const res = await api.get("/auth/avatar-signed-url");
```

4 Token Expiry / Validity

- Log decoded token expiry:

```
const token = user.accessToken;
const payload = JSON.parse(atob(token.split('.')[1]));
console.log("Token expires at:", new Date(payload.exp * 1000));
```

- If token expired → 401 occurs immediately.

5 Response Interceptor

- Handle 401 properly:

```
if (error.response?.status === 401) {
  toast.error("Session expired!");
  localStorage.removeItem("user");
  // optional: redirect
}
```

- Always log the **actual response**:

```
console.log(error.response.data, error.response.status);
```

6 Test Manually

- Copy the token from local storage.
- Use Postman/cURL to hit the endpoint with `Authorization: Bearer <token>`.
- If it fails there → backend issue, not Axios.

7 React Rendering Timing

- If API is called inside `useEffect` or `setTimeout`, make sure **user is already loaded**:

```
useEffect(function fetchAvatar() {
  if (!user) return;
  api.get("/auth/avatar-signed-url")...
}, [user]);
```

8 Small Tricks

- Clear local storage when debugging (`localStorage.clear()`), restart React dev server.
 - Log every step: `localStorage` → `parsed user` → `headers` → `request` → `response`.
-

💡 **Pro tip:** Always log headers in both Axios interceptor and the actual API call — that alone usually reveals 90% of the problems.