

Common issues and solutions

September 15, 2026

Common issues and solutions

Common issues and solutions

Conduit troubleshooting falls into four reader jobs: starting the local application, connecting it to the API, checking routes, and separating signed-in behavior from signed-out behavior. Choose the check that matches the problem before changing the application.

If you need to...	Go to
Prepare the local installation and connection details	Before you start
Start the application or change its API endpoint	Start the application and connect it to the API
Check which page each route opens	Check the declared routes
Compare signed-out and signed-in behavior	Handle signed-out and signed-in behavior
Interpret a returned API error	When a check fails
Continue with an account, navigation, or support task	What to do next

Before you start

Have the local project available and keep the expected connection details in view. The local web server uses port `4100` . The default API root is `https://conduit.productionready.io/api` .

Check	Value or condition
Dependencies	Dependencies are installed.
Local server	Run <code>npm start</code> .
Local port	<code>4100</code>

Check	Value or condition
API root	<code>https://conduit.productionready.io/api</code>
Authentication branch	A JWT, the browser's authentication token, loads the current user when present; without one, the application loads without a current-user token.

If you are checking authenticated behavior, know whether the browser has a stored JWT before you compare the result with the signed-out path.

With these details ready, start the application and connect it to the API.

Start the application and connect it to the API

Start here when the local application is not running or when requests must go to a local API instead of the default API root.

Prerequisites

- The project is available locally.
- Dependencies are installed.
- You know the API URL that the application must use.

Steps

1. Open a terminal in the project folder.
2. Enter `npm install` in the terminal.
3. Run `npm start`.
4. Check that the local server uses port `4100`.
5. If requests must use a local API, change the endpoint setting to the local server URL.

For example, use `http://localhost:3000/api` as the local API URL.

6. If the browser has a JWT, check requests that require authentication against the signed-in path. Token-bearing requests receive an `authorization` header using that token.

Result: The local application is started on port `4100`, and its requests use the selected API root.

With the application running, check the routes that lead to the surface you need.

Check the declared routes

Open the fixed routes before checking a route that needs an article slug, article identifier, or username.

Surface	How to open it
Home	Open <code>/</code> .
Login	Open <code>/login</code> .
Register	Open <code>/register</code> .
New article editor	Open <code>/editor</code> .
Existing article editor	Open the editor route with an article slug.
Article	Open the article route with an article identifier.
Settings	Open <code>/settings</code> .
Profile favorites	Open the profile favorites route with a username.
Profile	Open the profile route with a username.

Prerequisites

- The local application is running.
- You have an article slug for the existing article editor.
- You have an article identifier for the article page.
- You have a username for the profile and profile favorites pages.

If you want to check...	Value to enter
An existing article editor	The article slug.
An article page	The article identifier.
Profile favorites	The profile username.
A profile	The profile username.

Steps

1. Open the surface you want to check using the matching route or value in the table.
2. Compare the opened surface with its intended surface.

Result: Each checked path opens its intended application surface, or the route that needs attention is identified.

If the surface changes with authentication, compare the signed-out and signed-in branches next.

Handle signed-out and signed-in behavior

The application chooses its current-user loading path from the presence of a JWT in browser local storage. Use the branch that matches the browser state you are checking.

If the browser...	The application...
Has a JWT in local storage	Sets the token and loads the current user.
Does not have a JWT in local storage	Loads without a current-user token.

When a token is present, requests use it in the authorization header. When no token is present, the token plugin does not set that header. Compare like with like: check the signed-in behavior with a stored JWT and the signed-out behavior without one.

If either branch produces an unexpected result, use the checks in the next section.

When a check fails

Compare the application rule with the API text you see.

What you see	What it means	What to check
An API error appears as a returned key followed by a returned value	The application displays the returned error key and value.	Read both parts of the displayed message.
A route does not match one of the listed surfaces	The declared route table has no explicit fallback route.	Check the path spelling and any required <code>slug</code> , <code>id</code> , or <code>username</code> .
A signed-in request differs from a signed-out request	The browser's JWT state changes current-user loading and request authorization.	Check whether the browser has a JWT, then repeat the comparison using the matching branch.

The exact API error values come from the returned data rather than from a fixed application

message. Use the displayed key and value when diagnosing the request.

After identifying the issue, continue with the task that matches it.

What to do next

Continue with the task that matches the problem:

- For local prerequisites, see [System requirements](#).
- For route and page movement, see [Navigation basics](#).
- For signing in, see [Sign in](#).
- For creating an account, see [Sign up](#).
- For account settings, see [Manage your settings](#).
- For application error text, see [General error messages](#).
- For broader support references, see [Getting help](#).