

System requirements

September 15, 2026

System requirements

What you need to run Conduit

Conduit is a Vue 3 application. Pinia stores application state, Vue Router handles navigation, and Vite serves and builds the application. Bun installs packages and runs project commands. Playwright runs the end-to-end browser tests. `marked` renders Markdown, DOMPurify sanitises rendered content, and `date-fns` handles dates. The application uses a thin `fetch` wrapper for API requests.

Prepare the project locally before opening Conduit.

If you need to...	Go to
Prepare the repository	Before you start
Install packages and run the application	Install and run the application
Select a backend URL	Point Conduit to a backend
Run browser tests or code checks	Test and check the project
Understand what Conduit does	What is conduit; What can you do with conduit
Get help with a problem	Getting help

Before you start

Before you run a project command, have a clone that includes the `realworld` git submodule. This submodule supplies the specification and test suite, and the application imports its theme from it.

Prerequisites

- Access to the project repository.
- The `realworld` git submodule available in the clone.

Steps

1. Prepare the repository using the command that matches its current state:

If you need to...	Run
Clone the repository with its submodules	<code>git clone --recurse-submodules https://github.com/realworld-apps/vue-realworld-example-app</code>
Initialise the submodule in an existing clone	<code>git submodule update --init</code>

Result: The repository and its `realworld` submodule are available for installing dependencies.

Install and run the application

Use this procedure to install the project, start its development server, or create a production build for preview.

Prerequisites

- The repository and `realworld` submodule are available.

Steps

1. Install the project dependencies by running:

```
bun install
```

2. Choose the way you want to run the project:

If you need to...	Run
Develop with hot reload	<code>bun run serve</code>
Create a production build	<code>bun run build</code>
Preview the production build	<code>bun run preview</code>

3. Open `http://localhost:8080` after starting the development server.

Result: The development server is available at `http://localhost:8080` , or the production build is ready for preview and backend configuration.

Point Conduit to a backend

Conduit sends its API requests to a RealWorld backend. By default, it uses `https://api.realworld.show/api` ; use `VITE_API_URL` when you need to point the application to another spec-compliant backend.

Prerequisites

- A backend URL to use when you are not using the default.

Steps

1. Choose where to set the backend URL:

If you need to...	Set
Use the default backend	Leave <code>VITE_API_URL</code> unset
Use another backend for a shell command	<code>VITE_API_URL=http://localhost:8000/api</code> before <code>bun run serve</code>
Use another backend for local development	Set <code>VITE_API_URL</code> in <code>.env.local</code>

2. Start the application with `bun run serve` .

Result: The application starts with the selected backend URL and is ready for project checks.

Test and check the project

Run the browser suite after setup, then use linting and formatting checks before sharing changes.

Prerequisites

- The project dependencies are installed.
- The development server can be started by the Playwright configuration.

Steps

1. Install the Chromium browser and its dependencies by running:

```
bunx playwright install --with-deps chromium
```

2. Run the end-to-end test suite by running:

```
bun run test
```

3. Check the source with ESLint by running:

```
bun run lint
```

4. Format the source with Prettier by running:

```
bun run format
```

Result: The browser, end-to-end, lint, and formatting commands have been run for the project.