Atlassian Bug Dashboard

Development Workflow Guide

Project Structure

Component	Technology	Hosting	Local Port
Frontend	Next.js / React	Vercel	3000
Backend	FastAPI / Python	Railway	8000
Database	PostgreSQL	Railway	-

Development Workflow

1 Start Local Development Servers

Terminal 1 - Backend:

```
cd ~/path/to/atlassian-bug-dashboard/backend
poetry run uvicorn app.main:app --reload --port 8000
```

Terminal 2 - Frontend:

```
cd ~/path/to/atlassian-bug-dashboard
npm run dev
```

2 Point Frontend to Local Backend (Optional)

Edit `src/lib/api.ts`:

```
// Change this for local testing:
const API_BASE_URL = 'http://localhost:8000/api';

// Production URL (restore before committing):
const API_BASE_URL = 'https://atlassian-bug-dashboard-production.up.railway.app/api';
```

Remember: Always restore the production API URL before committing!

3 Test Your Changes Locally

What	URL
Frontend Dashboard	`http://localhost:3000`
Backend API Docs	`http://localhost:8000/docs`

4 Create Feature Branch & Commit

```
# Create a new branch
git checkout -b feature/my-new-feature

# Stage your changes
git add src/app/page.tsx src/lib/api.ts

# Commit with descriptive message
git commit -m "Add new feature description"

# Push to remote
git push -u origin feature/my-new-feature
```

5 Create Pull Request

```
# Interactive PR creation
gh pr create

# Or with title/body inline
gh pr create --title "Add feature X" --body "Description of changes"
```

6 Merge & Deploy

```
# Merge the PR (triggers auto-deploy)
gh pr merge --merge
```

Vercel and Railway will automatically redeploy when main is updated.

Quick Reference Commands

Development

Task	Command
Start frontend	`npm run dev`
Start backend	`poetry run uvicorn app.main:app --reload`
Build frontend	`npm run build`
Run frontend tests	`npm test`
Run backend tests	`poetry run pytest`
Install frontend deps	`npm install`
Install backend deps	`poetry install`

Git & GitHub

Task	Command
Check status	`git status`
Create branch	`git checkout -b feature/name`
Switch to main	`git checkout main`
Pull latest	`git pull origin main`
View PRs	`gh pr list`
View PR status	`gh pr status`
Merge PR	`gh pr merge`

Deployment

Task	Command
Force Vercel redeploy	`vercel --prod`
Force Railway redeploy	`railway up`
Check Vercel status	`vercel ls`

Pre-Commit Checklist

- □ API URL is set to production (not localhost)
- □ No console.log() statements left in code
- □ Frontend builds without errors: `npm run build`
- □ Changes tested locally
- □ Commit message is descriptive

Troubleshooting

Dashboard shows "Application error"

> Check browser console for errors. Usually means frontend/backend are out of sync. Make sure both are deployed with latest changes.

Changes not appearing after deploy

> - Clear browser cache (Cmd+Shift+R)
> - Check Vercel/Railway dashboards for deploy status
> - Force redeploy: `vercel --prod` or `railway up`

gh CLI not working

> Run `gh auth login` and authenticate via browser. Make sure token has `repo` and `read:org` scopes.

Key URLs

Resource	URL
Production Dashboard	atlassian-bug-dashboard.vercel.app
Production API	atlassian-bug-dashboard-production.up.railway.app/api
GitHub Repo	github.com/jamesbringetto/atlassian-bug-dashboard
Vercel Dashboard	vercel.com/dashboard
Railway Dashboard	railway.app/dashboard

Generated for Atlassian Bug Dashboard project