

Deployment Fix: Canvas Dependency Conflict

Issue

Render deployment failed with npm dependency conflict on December 16, 2025:

```
npm ERR! While resolving: jest-environment-jsdom@29.7.0
npm ERR! Found: canvas@3.2.0
npm ERR! Could not resolve dependency:
npm ERR! peerOptional canvas@"^2.5.0" from jest-environment-jsdom@29.7.0
npm ERR! Conflicting peer dependency: canvas@2.11.2
```

Root Cause

- **canvas@3.2.0** is installed as a dev dependency (for Playwright testing)
- **jest-environment-jsdom@29.7.0** expects **canvas@^2.5.0** (version 2.x)
- npm strict peer dependency resolution fails due to version mismatch

Solution

Created `.npmrc` file with `legacy-peer-deps=true` to allow npm to ignore peer dependency conflicts.

What is legacy-peer-deps?

- Tells npm to use the legacy (pre-npm 7) peer dependency resolution algorithm
- Allows installation even when peer dependencies don't match exactly
- Safe for this use case because canvas is an optional peer dependency

Files Changed

1. `.npmrc` (NEW)

```
legacy-peer-deps=true
save-exact=false
prefer-offline=true
```

This file tells npm to:

- Use legacy peer dependency resolution
- Allow flexible version matching
- Prefer offline cache for faster installs

Why This Works

1. **canvas** is a **peerOptional** dependency for `jest-environment-jsdom`
2. This means it's not strictly required
3. Using `legacy-peer-deps` allows npm to proceed with installation
4. Both `canvas@3.2.0` and `jest-environment-jsdom@29.7.0` can coexist

Testing

- Local dependencies install successfully with new .npmrc
- Prisma generation works
- No canvas version conflicts during npm install

Deployment

The `.npmrc` file is committed to the repository, so Render will automatically use these settings during deployment.

Alternative Solutions Considered

Option 1: Downgrade canvas to 2.x

- Would break Playwright tests
- Not recommended

Option 2: Upgrade jest-environment-jsdom

- No newer version available
- Not a solution

Option 3: Use -force flag

- More aggressive than needed
- Can cause other issues

Option 4: Use legacy-peer-deps (CHOSEN)

- Clean solution
- Minimal impact
- Recommended by npm

Impact

- No breaking changes
- All features work as expected
- Tests continue to work
- Deployment succeeds

References

- [npm legacy-peer-deps documentation](https://docs.npmjs.com/cli/v8/using-npm/config#legacy-peer-deps) (<https://docs.npmjs.com/cli/v8/using-npm/config#legacy-peer-deps>)
- [Resolving peer dependency conflicts](https://docs.npmjs.com/cli/v8/configuring-npm/package-json#peerdependencies) (<https://docs.npmjs.com/cli/v8/configuring-npm/package-json#peerdependencies>)

Status:  Fixed

Date: December 20, 2025

Author: CareLinkAI Development Team