

SolarlyAU Railway Deployment - Fixes Summary

Overview

The SolarlyAU application has been debugged and fixed for successful Railway deployment. All critical errors have been resolved.

Critical Issues Identified and Fixed

1. Production Build Crash - `import.meta.dirname` Undefined

Severity: CRITICAL 

Error in Logs:

Plain Text

```
TypeError [ERR_INVALID_ARG_TYPE]: The "paths[0]" argument must be of type string. Received undefined
at Object.resolve (node:path:1097:7)
at file:///app/dist/index.js:9483:18
```

Root Cause:

- The `import.meta.dirname` property is only available in ESM modules at development time
- When esbuild bundles the server code for production, `import.meta.dirname` becomes `undefined`
- This causes `path.resolve()` to crash with an invalid argument type error

File Affected:

- `server/_core/vite.ts` (lines 29, 53-54)

Solution Applied:

TypeScript

```
// Added at top of file:
import { fileURLToPath } from "url";
const __dirname = path.dirname(fileURLToPath(import.meta.url));
```

```
// Replaced all instances of import.meta.dirname with __dirname
```

Why This Works:

- `fileURLToPath()` converts the module URL to a file path
- `path.dirname()` extracts the directory
- This pattern works reliably in both development and production builds
- The bundler preserves this code correctly during esbuild compilation

2. OAuth Configuration Error

Severity: MEDIUM 

Error in Logs:

Plain Text

```
[OAuth]: # "ERROR: OAUTHSERVERURL is not configured! Set OAUTHSERVERURL environment variable."
```

Impact:

- OAuth functionality will not work
- Application continues to run but users cannot authenticate via OAuth

Solution: Add to Railway environment variables:

Plain Text

```
OAUTHSERVERURL=https://api.manus.im
```

Code Location:

- `server/_core/sdk.ts` (lines 33-38)
- `server/_core/env.ts` (line 5)

3. Static File Serving Path Issue

Severity: HIGH 

Issue:

- The `serveStatic()` function had inconsistent path logic

- Production mode was attempting to serve from wrong directory

Fix Applied:

TypeScript

```
// BEFORE (incorrect):
const distPath = process.env.NODE_ENV === "development"
  ? path.resolve(import.meta.dirname, "../", "dist", "public")
  : path.resolve(import.meta.dirname, "public");

// AFTER (correct):
const distPath = process.env.NODE_ENV === "development"
  ? path.resolve(__dirname, "../", "dist", "public")
  : path.resolve(__dirname, "../", "dist", "public");
```

Files Modified

server/_core/vite.ts

- Added `import { fileURLToPath } from "url"`
- Created `__dirname` variable using `path.dirname(fileURLToPath(import.meta.url))`
- Replaced 3 instances of `import.meta.dirname` with `__dirname`
- Fixed static file serving path resolution

Deployment Readiness

Code Quality

- All critical errors fixed
- Build process validated
- Production bundle verified
- Static file serving corrected

Configuration

- `railway.json` configured correctly
- `Procfile` set to start command

- Build command includes migrations
- Environment variables documented

Documentation

- Error fixes documented in [DEPLOYMENT_ERROR_FIXES.md](#)
- Quick start guide created: [RAILWAY_DEPLOYMENT_QUICK_START.md](#)
- Environment variables listed
- Troubleshooting guide provided

Build Process Verification

The application uses a multi-step build process:

Plain Text

1. `pnpm install`
 - └ Installs all dependencies
2. `pnpm run build`
 - ├ `vite build`
 - └ Builds React client to dist/public/
 - └ `esbuild`
 - └ Bundles server to dist/index.js
3. `NODE_ENV=production node dist/index.js`
 - ├ Loads environment variables
 - ├ Creates Express server
 - ├ Serves static files from dist/public/
 - └ Handles API routes

Environment Variables Required

Plain Text

```
# Core
DATABASE_URL=mysql://...
JWT_SECRET=<32-char-hex>
NODE_ENV=production

# OAuth
```

```
OAUTH_SERVER_URL=https://api.manus.im
VITE_APP_ID=<your-app-id>
OWNER_OPEN_ID=<your-open-id>

# Stripe
STRIPE_SECRET_KEY=sk_live_...
STRIPE_WEBHOOK_SECRET=whsec_...
VITE_STRIPE_PUBLISHABLE_KEY=pk_live_...

# Email
SENDGRID_API_KEY=SG.xxx
FROM_EMAIL=noreply@yourdomain.com
FROM_NAME=SolarlyAU

# SMS (optional )
CLICKSEND_USERNAME=xxx
CLICKSEND_API_KEY=xxx

# Manus APIs
BUILT_IN_FORGE_API_URL=https://api.manus.im
BUILT_IN_FORGE_API_KEY=<your-key>
VITE_FRONTEND_FORGE_API_KEY=<your-key>
VITE_FRONTEND_FORGE_API_URL=https://api.manus.im

# App
VITE_APP_TITLE=SolarlyAU
VITE_APP_LOGO=/logo.svg
BASE_URL=https://your-app.up.railway.app
```

Testing the Fix

Local Verification:

Bash

```
cd solar-lead-ai
pnpm install
pnpm run build
NODE_ENV=production node dist/index.js
```

Expected Output:

Plain Text

```
[OAuth]: https://api.manus.im "Initialized with baseURL:"  
Server running on http://localhost:3000/  
[Autonomous]: # "All autonomous systems initialized and running"  
[RatingScheduler]: # "Rating email scheduler initialized"
```

Railway Verification:

1. Deploy to Railway
2. Run: railway logs
3. Verify no ERR_INVALID_ARG_TYPE errors
4. Visit Railway URL to confirm app loads

Deployment Steps

1. Push to GitHub

Bash

```
git add .  
git commit -m "Fix Railway deployment errors"  
git push origin main
```

2. Create Railway Project

- Go to railway.app
- Select "Deploy from GitHub repo"
- Choose solar-lead-ai repository

3. Add MySQL Database

- Click "+ New"
- Select "Database" → "Add MySQL"

4. Configure Environment Variables

- Add all variables from list above
- Ensure DATABASE_URL is set by Railway

5. Run Migrations

- railway run pnpm db:push

6. Configure Stripe Webhook

- Endpoint: `https://your-app.up.railway.app/api/stripe/webhook`
 - Events: `checkout.session.completed`, `payment_intent.succeeded`
-

Troubleshooting

Build Fails

- Check Railway logs for specific errors
- Verify pnpm is being used
- Run `pnpm run check` locally to verify TypeScript

Runtime Errors

- Verify DATABASE_URL is set
- Check MySQL service is running
- Ensure all environment variables are configured

Static Files Not Loading

- Verify `dist/public/` exists after build
 - Check build command completed successfully
 - Verify `serveStatic()` path resolution
-

Success Criteria

✓ Application builds without errors ✓ No `ERR_INVALID_ARG_TYPE` errors on startup ✓
Server starts successfully ✓ OAuth initializes with correct baseURL ✓ Static files serve
correctly ✓ Database connection works ✓ API routes respond ✓ Stripe webhook
configured

Documentation Files Provided

1. **DEPLOYMENT_ERROR_FIXES.md** - Detailed explanation of all fixes
 2. **RAILWAY_DEPLOYMENT_QUICK_START.md** - Step-by-step deployment guide
 3. **DEPLOYMENT_FIXES_SUMMARY.md** - This file
-

Next Steps

1. Review the fixed code in `server/_core/vite.ts`
 2. Push corrected code to GitHub
 3. Deploy to Railway following the quick start guide
 4. Verify deployment using the testing steps
 5. Configure Stripe webhook
 6. Launch the application to clients
-

Status:  READY FOR PRODUCTION DEPLOYMENT

Last Updated: 2026-01-01 **Version:** 1.0.0 **Fixed By:** Manus Deployment Assistant