



Railway Root Directory Issue - Complete Fix Guide

Problem Identified

Railway is **IGNORING** the **Root Directory setting** in the dashboard and using the **wrong Dockerfile** (frontend instead of backend).

Evidence:

- **Dashboard shows:** Root Directory = backend/ 
- **Deployment logs show:** Using nginx-based Dockerfile (frontend) 
- **Error:** COPY nginx.conf /etc/nginx/conf.d/default.conf - ERROR: "/nginx.conf": not found

Root Cause

Railway configuration file precedence issue!

Railway found a **railway.json file in the ROOT directory** (`/home/ubuntu/workigom/railway.json`) which is **OVERRIDING** your dashboard settings.

Configuration Hierarchy (Priority Order):

1. **railway.json in the repo root**  **HIGHEST PRIORITY - THIS IS THE PROBLEM!**
2. `railway.toml` in the repo root
3. Dashboard settings (Root Directory, etc.)
4. `railway.json` in the backend directory
5. `railway.toml` in the backend directory

Current File Structure

```
/home/ubuntu/workigom/
├── railway.json          ! PROBLEM FILE! (Frontend config in root)
├── Dockerfile
├── nginx.conf
└── .railwayignore
    └── backend/
        ├── railway.json      ✓ Backend config (but ignored due to root file)
        ├── railway.toml      ✓ Backend config (but ignored due to root file)
        └── Dockerfile         ✓ Backend - Node.js based (what we want!)
```

The Problem File

`/home/ubuntu/workigom/railway.json` (root directory):

```
{
  "$schema": "https://railway.app/railway.schema.json",
  "build": {
    "builder": "dockerfile",
    "dockerfilePath": "Dockerfile",           ← Points to ROOT Dockerfile (frontend)
    "watchPatterns": [
      "src/**",
      "public/**",
      "index.html",
      "vite.config.ts"                         ← Frontend patterns
    ]
  },
  "deploy": {
    "startCommand": "nginx -g 'daemon off;'",   ← Frontend Start command
    "restartPolicyType": "on_failure",
    "restartPolicyMaxRetries": 10
  }
}
```

This file tells Railway:

- Use Dockerfile builder
- Use Dockerfile (which resolves to `./Dockerfile` = frontend!)
- Start nginx (frontend server)

Even though you set Root Directory to `backend/` in the dashboard, **this file takes precedence!**

✓ Solutions (Choose ONE)

Solution 1: Remove the Root `railway.json` (RECOMMENDED ⭐)

This is the cleanest solution - let Railway use the dashboard settings.

```
cd /home/ubuntu/workigom

# Backup the file first
mv railway.json railway.json.frontend.backup

# Or delete it completely
# rm railway.json

# Commit and push
git add railway.json
git commit -m "Remove root railway.json to fix backend deployment"
git push origin master
```

After doing this:

1. Railway will respect the Dashboard “Root Directory” setting
2. It will look in `backend/` directory
3. It will find `backend/railway.json` or `backend/railway.toml`
4. It will use `backend/Dockerfile` ✓

Solution 2: Modify Root railway.json to Point to Backend

Keep the root config but make it point to the backend.

```
cd /home/ubuntu/workigom
```

Replace the contents of `railway.json` with:

```
{
  "$schema": "https://railway.app/railway.schema.json",
  "build": {
    "builder": "dockerfile",
    "dockerfilePath": "backend/Dockerfile",
    "watchPatterns": [
      "backend/**"
    ],
  },
  "deploy": {
    "startCommand": "node dist/server.js",
    "healthcheckPath": "/api/health",
    "healthcheckTimeout": 100,
    "restartPolicyType": "on_failure",
    "restartPolicyMaxRetries": 10
  }
}
```

Note: This overrides your Root Directory setting. The `dockerfilePath` becomes relative to repo root.

```
git add railway.json
git commit -m "Update root railway.json to point to backend"
git push origin master
```

Solution 3: Use `.railwayignore` to Exclude Root Config

Add the root `railway.json` to `.railwayignore`:

```
cd /home/ubuntu/workigom
echo "railway.json" >> .railwayignore

git add .railwayignore
git commit -m "Ignore root railway.json to use backend config"
git push origin master
```

Note: This might not work if Railway processes the file before checking `.railwayignore`.



Steps to Force Fresh Deployment

After applying any solution:

1. Clear Railway Cache (In Dashboard)

- Go to your `workigom` service settings
- Click “Deployments” tab
- Click “⋮” menu on the latest deployment
- Select “Redeploy” or “Restart”

2. Or Disconnect & Reconnect (Nuclear Option)

If the above doesn't work:

a. **Disconnect the service:**

- Go to Settings tab
- Scroll to “Danger” section
- Click “Disconnect Source Repo”

b. **Wait 30 seconds**

c. **Reconnect:**

- Click “Connect Repo”
- Select your GitHub repo
- **IMPORTANT:** Set Root Directory to `backend/`
- Set Branch to `master`
- Deploy

3. Verify Correct Dockerfile is Being Used

Check the deployment logs for:

```
=====
Using Detected Dockerfile
=====
```

Then look for:

- ✓ Correct: `COPY --from=builder /app/prisma ./prisma`
- ✓ Correct: `RUN npm run prisma:generate`
- ✓ Correct: `HEALTHCHECK --interval=30s`
- ✗ Wrong: `COPY nginx.conf /etc/nginx/conf.d/default.conf`
- ✗ Wrong: `FROM nginx:alpine`

Recommended Action Plan

Follow these steps in order:

Step 1: Remove the Root `railway.json`

```
cd /home/ubuntu/workigom
mv railway.json railway.json.frontend.backup
git add railway.json
git commit -m "Remove root railway.json to fix backend deployment"
git push origin master
```

Step 2: Verify Backend Configs are Correct

```
# Check backend/railway.json exists
cat backend/railway.json

# Check backend/Dockerfile exists
ls -l backend/Dockerfile
```

Step 3: Trigger Railway Redeploy

- Go to Railway dashboard
- Navigate to your `workigom` service
- Go to “Deployments” tab
- Click latest deployment’s “ : ” menu
- Click “Redeploy”

Step 4: Monitor Deployment Logs

Watch for:

```
 "Using Detected Dockerfile"
 "COPY --from=builder /app/prisma ./prisma"
 "RUN npm run prisma:generate"
 "Running Prisma migrations..."
 "Starting server..."
```

Step 5: Test Health Endpoint

Once deployed:

```
curl https://your-service.railway.app/api/health
```

Expected response:

```
{
  "status": "ok",
  "timestamp": "2025-10-23T19:00:00.000Z",
  "uptime": 123.456,
  "database": "connected"
}
```

Configuration Files Comparison

Root `railway.json` (CAUSES PROBLEM):

```
{
  "build": {
    "dockerfilePath": "Dockerfile" ← Resolves to ./Dockerfile [frontend]
  },
  "deploy": {
    "startCommand": "nginx -g 'daemon off;'" ← Frontend command
  }
}
```

Backend `railway.json` (WHAT WE WANT):

```
{
  "build": {
    "dockerfilePath": "Dockerfile" ← Resolves to backend/Dockerfile [✓]
  },
  "deploy": {
    "startCommand": "node dist/server.js" ← Backend command [✓]
  }
}
```

Backend `railway.toml` (ALTERNATIVE):

```
[build]
builder = "dockerfile"
dockerfilePath = "Dockerfile" ← Resolves to backend/Dockerfile [✓]

[deploy]
startCommand = "/app/start.sh" ← Backend command [✓]
healthcheckPath = "/api/health"
```

How to Debug Future Issues

1. Check which config Railway is using:

```
cd /home/ubuntu/workigom

# List all Railway config files
find . -name "railway.*" -o -name ".railway*"
```

2. Check Railway dashboard settings:

- Root Directory: Should be `backend/` for backend service
- Builder: Should be `Dockerfile`
- Branch: Should be `master` or your main branch

3. Check deployment logs:

Look for the actual Dockerfile content being executed:

- Frontend indicators: `nginx`, `COPY nginx.conf`, `daemon off`
- Backend indicators: `prisma`, `node dist/server.js`, `start.sh`

4. Test locally:

```
# Test backend Dockerfile locally
cd /home/ubuntu/workigom/backend
docker build -t workigom-backend .
docker run -p 3001:3001 workigom-backend

# Should see:
# "🚀 Starting Workigom Backend..."
# "📦 Running Prisma migrations..."
# "✅ Starting server..."
```



Why Railway Dashboard Settings Were Ignored

Railway configuration precedence is:

1. Configuration files in the repository (highest priority)

- `railway.json` (root)
- `railway.toml` (root)

2. Dashboard settings (medium priority)

- Root Directory
- Build settings

3. Auto-detection (lowest priority)

- Detects Dockerfile automatically
- Detects language and framework

Since you had a `railway.json` in the root, it **overrode all dashboard settings**, including your Root Directory setting.



Expected Result After Fix

After removing the root `railway.json`:

1. Railway respects Dashboard “Root Directory = `backend/`” setting
2. Railway looks in `backend/` directory
3. Railway finds `backend/railway.json` or `backend/railway.toml`
4. Railway uses `backend/Dockerfile`
5. Build includes Prisma generation
6. Deployment runs `/app/start.sh`
7. Migrations run automatically
8. Health check responds on `/api/health`

9.  Backend API is live! 

If Problems Persist

Check 1: Git Status

```
cd /home/ubuntu/workigom
git status
```

Make sure the root `railway.json` is actually removed/modified and committed.

Check 2: Railway Cache

Railway might cache the old configuration. Try:

- Disconnect and reconnect the repo
- Or create a new Railway service

Check 3: Branch Mismatch

Make sure Railway is watching the correct branch:

- Dashboard shows `master` branch
- Your changes are pushed to `master`

Check 4: File Permissions

```
ls -la /home/ubuntu/workigom/backend/
```

Make sure `backend/Dockerfile` and `backend/railway.json` are readable.

Additional Resources

- [Railway Root Directory Docs](https://docs.railway.app/deploy/deployments#root-directory) (<https://docs.railway.app/deploy/deployments#root-directory>)
- [Railway Configuration Precedence](https://docs.railway.app/deploy/config-as-code) (<https://docs.railway.app/deploy/config-as-code>)
- [Railway Dockerfile Builds](https://docs.railway.app/deploy/dockerfiles) (<https://docs.railway.app/deploy/dockerfiles>)

Verification Checklist

After applying the fix:

- [] Root `railway.json` removed or modified
- [] Changes committed and pushed to GitHub
- [] Railway triggered redeploy
- [] Deployment logs show Prisma commands
- [] Deployment logs show `start.sh` execution
- [] No nginx-related errors in logs
- [] Health endpoint responds

- [] Backend API is accessible
-

Generated: October 23, 2025

Issue: Railway ignoring Root Directory setting due to conflicting `railway.json` in repo root

Fix: Remove root `railway.json` to allow dashboard settings to take effect