



Deployment Instructions for Workigom API Fix



Work Completed

1. Verified all backend routes are correctly configured
 2. Compiled TypeScript successfully
 3. Created test script to verify routes
 4. Committed changes to git locally
 5. **Push to GitHub required** (see below)
-



Step 1: Push to GitHub

Your code has been committed locally but needs to be pushed to GitHub. Choose one of these methods:

Option A: Using GitHub CLI (Recommended)

```
cd /home/ubuntu/workigom

# Configure git if needed
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"

# Push using gh CLI
gh auth login
git push origin master
```

Option B: Using Personal Access Token

```
cd /home/ubuntu/workigom

# Remove old remote
git remote remove origin

# Add new remote with your token
# Get a new token from: https://github.com/settings/tokens
git remote add origin https://YOUR_TOKEN@github.com/volkanakbulut73/workigom.git

# Push
git push origin master
```

Option C: Using SSH (If SSH key is set up)

```
cd /home/ubuntu/workigom

# Change remote to SSH
git remote set-url origin git@github.com:volkanakbulut73/workigom.git

# Push
git push origin master
```

Option D: Manual Upload

If the above methods don't work, you can manually upload the files:

1. Go to <https://github.com/volkanakbulut73/workigom>
2. Upload these files:
 - backend/test-routes.js
 - API_JOBS_404_FIX_SUMMARY.md
 - API_JOBS_404_FIX_SUMMARY.pdf



Step 2: Redeploy on Render

Once the code is pushed to GitHub:

1. Access Render Dashboard

- Go to: <https://dashboard.render.com>
- Sign in to your account
- Find your `workigom-backend` service

2. Clear Cache & Deploy

1. Click on your backend service
2. Click **"Manual Deploy"** button (top right)
3. Select **"Clear build cache & deploy"**
4. Click **"Deploy"**

3. Monitor Deployment

Watch the logs for:

```
=====
🚀 Workigom Backend API Server
=====
📍 Environment: production
🌐 Server running on: http://localhost:XXXX
❤️ Health Check: http://localhost:XXXX/api/health
=====
✅ Database connection successful
```

4. Wait for Completion

- Build typically takes 2-5 minutes
- Look for "Live" status indicator

- Green checkmark means successful deployment

Step 3: Test the Fix

Test 1: Health Check

```
curl https://workigom-backend.onrender.com/api/health
```

Expected: 200 OK with JSON response

Test 2: Jobs Endpoint

```
curl https://workigom-backend.onrender.com/api/jobs
```

Expected: 200 OK with paginated jobs data

Test 3: Browser Console

Open <https://workigom.vercel.app>, then in browser console:

```
fetch('https://workigom-backend.onrender.com/api/jobs')  
  .then(res => res.json())  
  .then(data => console.log('✅ Success:', data))  
  .catch(err => console.error('❌ Error:', err))
```

Test 4: Frontend UI

1. Go to <https://workigom.vercel.app>
2. Navigate to the jobs listing page
3. Jobs should now load without errors
4. Check browser DevTools Network tab - should see 200 responses

Troubleshooting

If Still Getting 404:

1. Check Render Logs:

- Go to Render Dashboard → Your Service → Logs
- Look for any startup errors
- Verify routes are being mounted

2. Verify Build:

- In Render logs, confirm: `npm run build` completed successfully
- Check for TypeScript compilation errors

3. Check Environment Variables:

- Verify `DATABASE_URL` is set
- Verify `CORS_ORIGIN` includes your frontend URL
- All required secrets are configured

4. Clear Browser Cache:

- Hard refresh: Ctrl+Shift+R (Windows/Linux) or Cmd+Shift+R (Mac)
- Clear site data in DevTools

5. Test Directly:

Use Postman or curl to isolate frontend vs backend issues:

```
bash
```

```
curl -v https://workigom-backend.onrender.com/api/jobs
```

If CORS Errors:

Check that `CORS_ORIGIN` environment variable on Render includes:

```
https://workigom.vercel.app
```

If Database Errors:

1. Verify PostgreSQL is running
2. Check `DATABASE_URL` format:
3. Ensure Prisma migrations are up to date:

```
postgresql://user:password@host:port/database?schema=public
```

```
bash
```

```
npm run prisma:migrate:deploy
```



Render Configuration Checklist

Ensure your Render service has these exact settings:

Service Settings

- **Name:** workigom-backend
- **Region:** Choose closest to users
- **Branch:** master
- **Root Directory:** (leave empty or set to `backend` if using backend subdirectory)

Build & Start Commands

Build Command:

```
npm install && npm run prisma:generate && npm run build
```

Start Command:

```
npm start
```

Environment Variables (Required)

Variable	Example Value	Status
<code>NODE_ENV</code>	production	Required
<code>DATABASE_URL</code>	postgresql://...	Required
<code>JWT_SECRET</code>	your_secret_key	Required
<code>JWT_REFRESH_SECRET</code>	your_refresh_secret	Required
<code>CORS_ORIGIN</code>	https://workigom.vercel.app	Required
<code>JWT_EXPIRES_IN</code>	7d	Optional
<code>JWT_REFRESH_EXPIRES_IN</code>	30d	Optional

Auto-Deploy Settings

- ☒ Enable auto-deploy
- ☒ Auto-deploy branch: master
- ☒ GitHub repository: volkanakbulut73/workigom



Files Changed

The following files were added/modified:

```
workigom/
├── backend/
│   └── test-routes.js           # New: Route verification script
├── API_JOBS_404_FIX_SUMMARY.md # New: Complete fix documentation
├── API_JOBS_404_FIX_SUMMARY.pdf # New: PDF version
└── DEPLOY_INSTRUCTIONS.md      # New: This file
```








What Was Verified

- ☒ All route files exist and are compiled
- ☒ `backend/src/routes/job.routes.ts` defines GET /
- ☒ `backend/src/routes/index.ts` mounts /jobs
- ☒ `backend/src/app.ts` mounts /api
- ☒ TypeScript compilation successful
- ☒ Dist directory contains all compiled routes
- ☒ Route chain verified: `/api/jobs` → `index.ts` → `job.routes.ts` → `job.controller.ts`

Expected Outcome

After completing the steps above:

1.  GitHub will have the latest code
 2.  Render will rebuild with verified configuration
 3.  `/api/jobs` endpoint will return 200 OK
 4.  Frontend will successfully fetch and display jobs
 5.  No more 404 or `ERR_CONNECTION_REFUSED` errors
-

Need Help?

If you encounter any issues:

1. Check the logs:

- Render: Dashboard → Service → Logs
- Browser: DevTools → Console & Network tabs

2. Verify the basics:

```
```bash
Is the backend running?
curl https://workigom-backend.onrender.com/api/health
```

# Can you reach the endpoint?







```
curl -I https://workigom-backend.onrender.com/api/jobs
```
```

1. Review the documentation:

- See `API_JOBS_404_FIX_SUMMARY.md` for detailed analysis
 - Check route structure and configuration
-

Success Indicators

You'll know it's working when:

-  Health check returns `{"success": true, "database": "connected"}`
 -  `/api/jobs` returns `{"success": true, "data": [...], "pagination": {...}}`
 -  Browser console shows no 404 or network errors
 -  Frontend displays jobs list
 -  Render logs show:  Database connection successful
-

Last Updated: October 25, 2025

Status: Ready for Deployment 