

✓ Backend/src/main.tsx Deletion Verification - COMPLETE

Date: October 24, 2025 - 21:20 UTC

Status: ✓ ALL CHECKS PASSED - File deletion confirmed

📋 Summary of Actions Taken

Step 1: Local File System Check ✓

```
ls -la /home/ubuntu/workigom/backend/src/main.tsx
```

Result: ls: cannot access 'backend/src/main.tsx': No such file or directory

Conclusion: File does NOT exist locally

Step 2: Git Status Check ✓

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

Result: No backend/src/main.tsx in tracked files

Found: Untracked files and modified documentation (unrelated to issue)

Step 3: Recent Commits Check ✓

```
git log --oneline -10
```

Latest Relevant Commits:

1. 0f26bc3 - chore: Update cache bust and add Render cache issue documentation
2. c4139e1 - fix: Remove frontend main.tsx from backend src directory ★
3. f54b291 - fix: Move all backend files to backend directory for Render.com deployment


Conclusion: File was properly deleted in commit c4139e1

Step 4: Commit Details Verification ✓

```
git show c4139e1
```

Commit Details:

- **Author:** volkanakbulut73

- **Date:** Fri Oct 24 21:10:24 2025 +0000
- **Changes:** 1 file changed, 7 deletions(-)
- **File:** backend/src/main.tsx  DELETED

Content That Was Deleted:

```
import { createRoot } from "react-dom/client";
import App from "./App.tsx";
import "./index.css";

createRoot(document.getElementById("root")!).render(<App />);
```

Step 5: Remote Repository Sync Check

```
git log origin/master -5
```


Result: Remote is at commit 0f26bc3 (includes the deletion)

Conclusion: Changes successfully pushed to GitHub

Step 6: Git Tree Verification

```
git ls-tree -r HEAD --name-only | grep -i main.tsx
```








Result: Only src-frontend/main.tsx exists

Conclusion: No main.tsx in backend/src directory 

Step 7: Backend Directory Contents Check

```
git ls-tree -r HEAD --name-only backend/src/
```

Files Present:

- app.ts 
- config/database.ts 
- config/multer.ts 
- controllers/controller.ts 
- middleware/middleware.ts 
- routes/routes.ts 
- server.ts 
- types/d.ts 
- utils/*.ts 


Files NOT Present:

- main.tsx  (correctly deleted)

Step 8: File History Check

```
git log --all --full-history --oneline -- backend/src/main.tsx
```

Result:

- f54b291 - File was added (during restructuring)
- c4139e1 - File was deleted 

Conclusion: Clean deletion history confirmed

Step 9: Local vs Remote Diff Check

```
git diff HEAD origin/master
```

Result: No differences

Conclusion: Local and remote are perfectly in sync



Additional Actions Taken



Updated Cache Bust Value

File: backend/Dockerfile

Change: Updated CACHE_BUST from 202510232325 to 202510242120

Purpose: Force Render.com to invalidate Docker layer cache

Commit: 0f26bc3 - chore: Update cache bust and add Render cache issue documentation



Created Documentation

Files Created:

1. RENDER_CACHE_ISSUE_SUMMARY.md - Detailed analysis and solutions
2. RENDER_CACHE_ISSUE_SUMMARY.pdf - PDF version for reference
3. VERIFICATION_COMPLETE.md - This file



Pushed All Changes to GitHub

Latest Commits:

```
0f26bc3 chore: Update cache bust and add Render cache issue documentation
c4139e1 fix: Remove frontend main.tsx from backend src directory
```



Root Cause Analysis

What Happened:

The file `backend/src/main.tsx` was accidentally created during a previous restructuring operation (f54b291). This React frontend file should never have been in the backend directory.

Why Render Still Shows Errors:

Even though the file is properly deleted from GitHub, **Render.com is using Docker layer caching**. The cached layers from the previous build still contain the file, causing TypeScript compilation errors.

The Errors You're Seeing:

```
src/main.tsx(2,30): error TS2307: Cannot find module 'react-dom/client'
src/main.tsx(3,19): error TS2307: Cannot find module './App.tsx'
```

These errors occur because:

1. The cached build contains `backend/src/main.tsx`
2. TypeScript tries to compile it as part of the backend
3. Backend doesn't have React dependencies
4. Compilation fails

✓ Verification Results

Check	Status	Details
File exists locally	✗ No	File deleted from local filesystem
Committed properly	✓ Yes	Commit c4139e1 shows proper deletion
Pushed to GitHub	✓ Yes	Remote repository is up to date
File in backend/src	✗ No	Git tree verification confirms absence
Local/Remote sync	✓ Yes	No differences between local and remote
Cache bust updated	✓ Yes	Dockerfile updated to force rebuild
Documentation created	✓ Yes	Complete analysis and solutions documented

IMMEDIATE ACTION REQUIRED

The file is properly deleted from GitHub, but **Render.com needs to clear its build cache**. You have three options:

Option 1: Manual Redeploy (RECOMMENDED) 🖱️

1. Go to your Render dashboard: <https://dashboard.render.com>
2. Click on the `workigom` service
3. Click **“Manual Deploy”** dropdown
4. Select **“Clear build cache & deploy”**
5. Wait for the fresh build to complete

Option 2: Clear Cache via Settings

1. In Render dashboard, go to service **Settings**
2. Scroll to **“Build & Deploy”** section
3. Click **“Clear Build Cache”** button
4. Go back to **Deployments**
5. Click **“Manual Deploy”**

Option 3: Wait for Automatic Deploy

The cache bust update (`202510242120`) will force a rebuild on the next automatic deployment. However, this may take time.



Expected Outcome

After clearing Render’s cache and redeploying:



Build Should Succeed

- TypeScript compilation will not see `main.tsx`
- No errors about `react-dom/client`
- No errors about `./App.tsx`
- Backend will build successfully



Deployment Should Complete

- Docker build will complete without errors
- Service will start properly
- Health checks will pass
- Application will be accessible



If Issues Persist

If you still see errors after clearing cache:

1. **Check build logs** - Verify which commit Render is using
 2. **Check branch** - Ensure it’s building from `master` branch
 3. **Check root directory** - Should be `backend/` (as shown in screenshots)
 4. **Verify webhook** - Ensure GitHub webhook is properly configured
 5. **Contact support** - If cache clearing doesn’t work, contact Render support
-



Key Takeaways



What We Confirmed:

1. File `backend/src/main.tsx` is **completely deleted**
2. Deletion was **properly committed** (c4139e1)
3. Changes are **successfully pushed** to GitHub
4. Repository is **clean and in sync**
5. Cache bust value has been **updated to force rebuild**



What You Need to Do:

1. **Clear Render's build cache** (see options above)
2. **Trigger a fresh deployment**
3. **Monitor the build logs** for success
4. **Verify the service starts** without errors



Final Status:



ALL VERIFICATION COMPLETE - The issue is entirely on Render's side due to cached Docker layers. Clearing the cache will resolve the problem.

Generated: October 24, 2025 - 21:20 UTC

Next Action: Clear Render build cache and redeploy

Expected Resolution Time: 5-10 minutes after cache clear