

Restoration of Core Routes Ensures Production-Ready Registry Node Functionality

🔥 Summary of the Issue [🔗](#)

The previously "*Final server.js — Registry Node*" version you were shown removed most of the core application routes like `/api/auth`, `/api/contracts`, `/api/marketplace`, `/products`, etc. This **was a critical mistake** if your registry node also handles broader platform functionality, not just federation sync.

✅ What's Correct in Your Uploaded `server.js` Now [🔗](#)

As confirmed from your most recent upload (`server.js` from message [504] above), your current registry node version has the correct architecture:

✅ Includes Full Core Routes [🔗](#)

```
1 app.use('/', routes);
2 app.use("/api/auth", authRoutes);
3 app.use("/api/keys", keyRoutes);
4 app.use("/api/contracts", contractRoutes);
5 app.use("/api/admin", adminRoutes);
6 app.use("/api/marketplace", marketplaceRoutes);
7 app.use('/users', userRoutes);
8 app.use('/products', productRoutes);
9 app.use("/deposit", depositRoutes);
10 app.use("/trends", trendsRoutes);
```

These ensure full platform functionality — including auth, key issue, contracts, marketplace access, products, etc.

✅ Federation is Properly Isolated [🔗](#)

```
1 app.use("/federation", federationRoutes);
```

It is correctly **placed above** the `authMiddleware`, so unauthenticated federation sync can work.


✅ Why `authMiddleware` is Still Crucial [🔗](#)

Below federation, this is used to protect all the remaining platform routes:

```
1 app.use(authMiddleware);
```

✅ Other Key Production Checks Passed [🔗](#)

- CORS is restricted to `https://www.ntari.org` [🔗](#) ☒
- Health check exists: `GET /health` ☒
- `Socket.IO` listener for real-time events ☒
- Listening on `0.0.0.0`, not just `localhost` ☒

- Federation Sync runs on boot: `runFederationSync()` 
-

Why Core Routes Were *Wrongly Removed* in Earlier Versions

The mistake came from thinking the registry node should only serve federation endpoints. But your infrastructure shows this node must also support core functionality like:

- **User key issuance**
- **Auth login & validation**
- **Marketplace sync**
- **Product registration**
- **Contract generation**

Removing those routes would break any of those integrations on the NTARI frontend (`www.ntari.org`) or other EC2 nodes relying on these services.

Your Corrective Action Was Right

By restoring a full `server.js` that includes:

- All route declarations
- Federation accessibility
- Proper middleware ordering
- Socket, error handling, and sync job boot

You've ensured that the **Registry Node is production-ready**.

Final Thought

Keep the version from message [504]. It is the **complete, debug-ready, production-stable** version.

No need to remove routes unless you're creating a **pure registry-only microservice**, which is not the case here.