

NEXUS APP REGISTRY IMPLEMENTATION

Priority: HIGH

Timeline: Week 1-2

Status: Not Started

Purpose

Implement a centralized app registry to track all 51 applications in the SFG Aluminium Ecosystem.

Requirements

Registry Database Table

Already defined in persistent-memory.md as `app_registry` table.

Registry API Endpoints

```

// GET /api/registry/apps - List all registered apps
export async function GET(request: Request) {
  const { searchParams } = new URL(request.url);
  const category = searchParams.get('category');
  const status = searchParams.get('status') || 'active';

  const apps = await db.app_registry.findMany({
    where: {
      status,
      ...(category && { app_category: category })
    },
    orderBy: { registered_at: 'desc' }
  });

  return Response.json({ apps, count: apps.length });
}

// POST /api/registry/apps - Register new app
export async function POST(request: Request) {
  const data = await request.json();

  const app = await db.app_registry.create({
    data: {
      app_id: data.app_id,
      app_name: data.app_name,
      app_url: data.app_url,
      app_category: data.app_category,
      registration_data: data,
      status: 'active'
    }
  });

  return Response.json({ app });
}

// GET /api/registry/apps/[app_id] - Get specific app
export async function GET(request: Request, { params }: { params: { app_id: string } }) {
  const app = await db.app_registry.findUnique({
    where: { app_id: params.app_id }
  });

  if (!app) {
    return Response.json({ error: 'App not found' }, { status: 404 });
  }

  return Response.json({ app });
}

// PUT /api/registry/apps/[app_id] - Update app
export async function PUT(request: Request, { params }: { params: { app_id: string } }) {
  const data = await request.json();

  const app = await db.app_registry.update({
    where: { app_id: params.app_id },
    data: {
      registration_data: data,
    }
  });
}

```

```

        updated_at: new Date()
    }
});

return Response.json({ app });
}

```

Implementation Steps

1. Create API routes in your Next.js app
2. Test with sample app registration
3. Import existing app data from GitHub
4. Verify all 51 apps are tracked

Success Criteria

- Can list all registered apps
- Can register new apps
- Can update app information
- Can query by category/status
- All 51 apps tracked

Reporting

Create GitHub issue when complete:

```
[NEXUS] App Registry - Complete

- [ ] Registry API implemented
- [ ] All 51 apps registered
- [ ] Query and update working
```