

Location Block Organization Pattern

Complete Guide for Server Files

📄 What is the Location Block Pattern?

MAKE SURE THAT ANY INTRODUCTORY STATEMENTS ABOUT THE FUNCTION AND DEFINITION OF FILES ARE AT THE TOP OF THE FILE!!!!

The ****Location Block Pattern**** organizes code into clear, logical sections that mirror your actual workflow. Instead of having code scattered throughout the file, everything related to a specific function or workflow step is grouped together in a clearly labeled "Location Block."

✅ Benefits:

1. ****Easy to Find****: Need to modify the Payment workflow? Go to Location Block 2.
2. ****Easy to Debug****: Issue with authentication? Check Location Block 1.
3. ****Easy to Scale****: Adding a new workflow card? Just add a new Location Block.
4. ****Easy to Understand****: New developers can immediately see how the system is organized.

🗺️ How Location Blocks Map to Your Lawyers Workflow

****Your Lawyers Workflow (From PDF):****

...

Step 1: Terms of Business (TOB)

- └─ Review & Sign Terms
- └─ Insert Signatures
- └─ Preview Signed PDF
- └─ Complete & Return to Dashboard

Step 2: Payment

- └─ Select Service
- └─ Complete Payment (75% upfront)
- └─ Generate Invoice

Step 3: Client Consent Form

- └─ Fill Client Details
- └─ Preview Form
- └─ Send to Client
- └─ Wait for Client Return

Step 4: Case Referral & Instructions

- └─ Complete Instructions
- └─ Upload Law Firm Forms
- └─ Submit Referral

Step 5: Document Library

- └─ Upload Medical Records
- └─ Upload Legal Documents
- └─ Browse Library

Step 6: Client Management Hub

- └─ View All Clients

...

How This Maps to Server Location Blocks:

LawyersDashboard server.js

...

Location Block A: Imports & Configuration
└─ Sets up the foundation for everything

Location Block B: Server & Database Setup
└─ Connects to qolae_lawyers database

Location Block C: Middleware & Plugins
└─ CORS, Cache-busting, JWT, GDPR

Location Block 1: Authentication & Routing
└─ Login, health checks, redirects

Location Block 2: Main Dashboard Route
└─ Serves the lawyers-dashboard.ejs file

Location Block 3: Bootstrap API Endpoint
└─ Provides initial data to dashboard

Location Block 4: Lawyer Data API Endpoints
└─ GET lawyer profile, status, workflow stage

Location Block 5: Workflow Modal Routes
├─ TOB Modal (Step 1)
├─ Payment Modal (Step 2)
├─ Consent Modal (Step 3)
├─ Referral Modal (Step 4)
└─ Document Upload Modal (Step 5)

Location Block 6: Server Startup
└─ Starts the server on port 3002

...

API fastify_server.js (SSOT)

...

Location Block A: Imports & Configuration
└─ Sets up centralized API services

Location Block B: Server & Database Setup
└─ Connects to all QOLAE databases

Location Block C: Middleware & Plugins
└─ CORS, JSON parser, JWT, Multipart

Location Block 1: Root & Health Endpoints
└─ API information, health checks

Location Block 2: Authentication Endpoints
└─ JWT token generation, verification

Location Block 3: Signature Endpoints
└─ Canvas signature drawer, save signatures

Location Block 4: PDF Processing Endpoints

└─ PDF flattening, TOB PDF operations

Location Block 5: Route Module Registration

- └─ Email Routes
- └─ Document Routes
- └─ WebSocket Routes
- └─ Auth Routes
- └─ Workspace Routes
- └─ SSOT Services (PIN Validation, Health Monitor, GDPR)

Location Block 6: Server Startup & Shutdown

- └─ Starts server, handles graceful shutdown

...

🎯 Direct Comparison: Before vs After

BEFORE (Disorganized):

```
```javascript
// Code scattered everywhere
import fastify from 'fastify';
server.get('/dashboard', ...);
import path from 'path';
server.get('/api/lawyer', ...);
const pool = new Pool(...);
server.post('/api/save-signature', ...);
server.register(cors, ...);
```
```

✗ Hard to find anything

✗ Hard to debug

✗ Hard to maintain

AFTER (Location Block Pattern):

```
```javascript
// =====
// LOCATION BLOCK A: IMPORTS & CONFIGURATION
// =====
import fastify from 'fastify';
import path from 'path';

// =====
// LOCATION BLOCK B: SERVER & DATABASE SETUP
// =====
const server = fastify({ logger: true });
const pool = new Pool(...);

// =====
// LOCATION BLOCK C: MIDDLEWARE & PLUGINS
// =====
server.register(cors, ...);

// =====
// LOCATION BLOCK 1: AUTHENTICATION & ROUTING
// =====
server.get('/dashboard', ...);
```
```

```
// =====
// LOCATION BLOCK 4: LAWYER DATA API ENDPOINTS
// =====
server.get('/api/lawyer', ...);

// =====
// LOCATION BLOCK 3: SIGNATURE ENDPOINTS
// =====
server.post('/api/save-signature', ...);
'''
```

- ✓ Easy to find
- ✓ Easy to debug
- ✓ Easy to maintain

📝 Naming Convention Rules

Setup Blocks (Always at Top):

- **Location Block A**: Imports & Configuration
- **Location Block B**: Server & Database Setup
- **Location Block C**: Middleware & Plugins

Functional Blocks (Numbered):

- **Location Block 1, 2, 3...**: Specific workflow functions
- Each block handles ONE clear responsibility
- Blocks mirror your actual workflow steps

Shutdown Block (Always at Bottom):

- **Location Block N**: Server Startup & Shutdown

🔍 How to Use This Pattern When Troubleshooting

Example 1: Payment Modal Not Loading

1. **First Check**: Location Block 5 (Workflow Modal Routes)
 - Look for `/paymentModal` route
 - Verify it's serving the correct EJS file
2. **Second Check**: Location Block C (Middleware)
 - Verify view engine is configured correctly
 - Check static file serving
3. **Third Check**: Location Block 2 (Main Dashboard Route)
 - Verify dashboard is loading correctly
 - Check if workflow_stage allows payment access

Example 2: Lawyer Data Not Showing

1. **First Check**: Location Block 4 (Lawyer Data API Endpoints)
 - Look for `/lawyers-dashboard/api/lawyer` route
 - Check database query
2. **Second Check**: Location Block B (Database Setup)
 - Verify PostgreSQL pool connection

- Check database credentials

3. **Third Check**: Location Block 3 (Bootstrap API)
 - Verify bootstrap endpoint returns correct data

Example 3: JWT Authentication Failing

1. **First Check**: Location Block 2 (Authentication Endpoints)
 - Verify `/auth/token` route works
 - Check JWT signing
2. **Second Check**: Location Block C (Middleware)
 - Verify JWT plugin is registered
 - Check JWT secret in environment
3. **Third Check**: Location Block 3 (Bootstrap API)
 - Verify token is being stored/retrieved correctly

🌈 How This Mirrors Your EJS Files

Your lawyers-dashboard.ejs Structure:

```

<html
  <!-- Location Block 0: ALL CSS STYLES -->
  <style>...</style>

  <!-- Location Block 1: TOB WORKFLOW CARD -->
  <div class="workflow-card" id="cardTOB">...</div>
  <script>
    // TOB-specific JavaScript
  </script>

  <!-- Location Block 2: PAYMENT WORKFLOW CARD -->
  <div class="workflow-card" id="cardPayment">...</div>
  <script>
    // Payment-specific JavaScript
  </script>

  <!-- Location Block 3: CONSENT WORKFLOW CARD -->
  <div class="workflow-card" id="cardConsent">...</div>
  <script>
    // Consent-specific JavaScript
  </script>

  <!-- Location Block 7 & 8: BOOTSTRAP & UTILITY FUNCTIONS -->
  <script>
    // Bootstrap API call
    // Socket.IO setup
    // Utility functions
  </script>

```

Your Server Files Now Match This Pattern:

```

<javascript
  // Location Block A: Imports (like CSS at top of EJS)

  // Location Block 1: TOB-related routes

```

```
server.get('/tobModal', ...);

// Location Block 2: Payment-related routes
server.get('/paymentModal', ...);

// Location Block 3: Consent-related routes
server.get('/consentModal', ...);

// Location Block 6: Startup (like utility functions at bottom of EJS)
'''

---
```

Implementation Checklist

```
### **For LawyersDashboard server.js:**
- [ ] Location Block A: All imports at top
- [ ] Location Block B: Database connection setup
- [ ] Location Block C: All middleware registration
- [ ] Location Block 1: Authentication & routing
- [ ] Location Block 2: Main dashboard route
- [ ] Location Block 3: Bootstrap API
- [ ] Location Block 4: Lawyer data endpoints
- [ ] Location Block 5: All workflow modal routes
- [ ] Location Block 6: Server startup
```

```
### **For API fastify_server.js:**
- [ ] Location Block A: All imports at top
- [ ] Location Block B: Database connections
- [ ] Location Block C: All middleware registration
- [ ] Location Block 1: Root & health endpoints
- [ ] Location Block 2: Authentication endpoints
- [ ] Location Block 3: Signature endpoints
- [ ] Location Block 4: PDF processing endpoints
- [ ] Location Block 5: Route module registration
- [ ] Location Block 6: Server startup & shutdown
```

Next Steps

1. **Review the reorganized files** in the artifacts above
2. **Compare to your current server files** to see the difference
3. **Test the reorganized structure** in your development environment
4. **Apply the pattern to any remaining files** that need organization

Key Takeaway

The Location Block Pattern is about organizing your code the same way you organize your workflow.

Just like your EJS files have clear sections for each workflow card, your server files now have clear sections for each functional responsibility. This makes your entire codebase consistent, maintainable, and easy to understand.

When everything follows the same pattern, you can navigate your entire system with confidence!

