

# NovaRax Tenant Manager Plugin - Complete File Structure

## Installation Path

```
/var/www/app.novarax.ae/wp-content/mu-plugins/novarax-tenant-manager/
```

## Complete Directory Tree

```
novarax-tenant-manager/
|
|   novarax-tenant-manager.php      # Main plugin file (already created)
|   readme.txt                    # Plugin documentation
|   LICENSE                       # License file
|   .gitignore                     # Git ignore file
|
|   includes/                      # Core functionality
|       class-database-manager.php # Database operations (already created)
|       class-tenant-operations.php # Tenant CRUD operations (already created)
|       class-tenant-validator.php # Input validation
|       class-subdomain-manager.php # Subdomain handling
|       class-provisioning-queue.php # Asynchronous provisioning
|       class-logger.php           # Logging system
|       class-email-notifications.php # Email templates and sending
|       class-security.php         # Security utilities
|
|   api/                           # REST API
|       class-api-handler.php     # Main API handler
|       class-api-authentication.php # API authentication
|       class-api-endpoints.php   # API endpoint definitions
|
|   integrations/                 # Third-party integrations
|       class-woocommerce-integration.php # WooCommerce hooks
|       class-subscription-handler.php    # Subscription management
|
|   admin/                         # Admin interface
|       class-admin-interface.php    # Main admin menu and pages
|       class-tenant-list-table.php  # Tenant list table (WP_List_Table)
|       class-admin-ajax.php        # AJAX handlers
|       class-settings.php          # Settings page
|       class-bulk-actions.php      # Bulk operations
|
```

```
|   └── views/           # Admin view templates
|       ├── dashboard.php      # Main dashboard
|       ├── tenant-list.php    # Tenants list page
|       ├── tenant-add.php     # Add new tenant form
|       ├── tenant-edit.php    # Edit tenant form
|       ├── tenant-view.php    # Single tenant details
|       ├── settings.php       # Settings page
|       ├── analytics.php      # Analytics dashboard
|       └── partials/          # Reusable partials
|           ├── header.php
|           ├── footer.php
|           ├── tenant-stats-widget.php
|           └── recent-activity.php
|
|   └── meta-boxes/        # Admin meta boxes
|       ├── class-tenant-info-metabox.php
|       ├── class-subscription-metabox.php
|       └── class-usage-metabox.php
|
└── assets/              # Static assets
    ├── css/
    |   ├── admin.css          # Admin styles
    |   ├── admin.min.css       # Minified admin styles
    |   └── admin-rtl.css       # RTL support
    |
    ├── js/
    |   ├── admin.js            # Admin JavaScript
    |   ├── admin.min.js         # Minified admin JavaScript
    |   ├── tenant-form.js       # Tenant form validation
    |   └── charts.js           # Analytics charts
    |
    └── images/
        ├── logo.png
        ├── icon-tenant.svg
        └── placeholder-avatar.png
|
└── languages/            # Translations
    ├── novarax-tenant-manager.pot  # Translation template
    ├── novarax-tenant-manager-ar.po # Arabic translation
    └── novarax-tenant-manager-ar.mo # Compiled Arabic
|
└── tests/                # Unit tests (PHPUnit)
    ├── bootstrap.php
    └── test-database-manager.php
```

```
|   └── test-tenant-operations.php  
|   └── test-api-endpoints.php  
  
└── docs/           # Documentation  
    ├── api-reference.md  
    ├── hooks-filters.md  
    ├── developer-guide.md  
    └── troubleshooting.md
```

---

## Files Created So Far

- novarax-tenant-manager.php** - Main plugin file with initialization
  - includes/class-database-manager.php** - Complete database management
  - includes/class-tenant-operations.php** - Full tenant CRUD operations
- 

## Next Files to Create (Priority Order)

### Phase 1: Core Functionality (Week 1)

#### 1. **includes/class-tenant-validator.php**

- Validate usernames, emails, subdomains
- Check for duplicates
- Sanitize inputs

#### 2. **includes/class-logger.php**

- Log to database and files
- Different log levels (info, warning, error)
- Log rotation

#### 3. **includes/class-security.php**

- Encryption/decryption methods
- IP detection
- Nonce validation
- SQL injection prevention

#### 4. **includes/class-email-notifications.php**

- Welcome email template

- Activation email template
- Suspension/cancellation emails
- Invoice emails

## 5. **includes/class-subdomain-manager.php**

- Validate subdomain availability
- Generate unique subdomains
- Handle subdomain conflicts

## 6. **includes/class-provisioning-queue.php**

- Queue system for async provisioning
  - Process queue in background
  - Handle failures and retries
- 

## Phase 2: Admin Interface (Week 2)

### 7. **admin/class-admin-interface.php**

- Register admin menus
- Enqueue scripts/styles
- Load admin pages

### 8. **admin/class-tenant-list-table.php**

- Extend WP\_List\_Table
- Display tenants in table format
- Sortable columns
- Bulk actions

### 9. **admin/views/tenant-list.php**

- Tenant list page HTML
- Search and filters
- Pagination

### 10. **admin/views/tenant-add.php**

- Add new tenant form
- Field validation

- Submit handler

## 11. **admin/views/tenant-edit.php**

- Edit tenant form
- Pre-populate fields
- Update handler

## 12. **admin/views/tenant-view.php**

- Single tenant dashboard
- Display all tenant info
- Quick actions

## 13. **admin/views/dashboard.php**

- Admin dashboard widgets
- Statistics and charts
- Recent activity

## 14. **admin/class-admin-ajax.php**

- AJAX handlers for admin
- Check username availability
- Quick actions (suspend, activate)

---

## Phase 3: WooCommerce Integration (Week 3)

### 15. **includes/integrations/class-woocommerce-integration.php**

- Hook into WooCommerce events
- Order completed hook
- Subscription status changes
- Product-to-module mapping

### 16. **includes/integrations/class-subscription-handler.php**

- Monitor subscription status
- Handle renewals
- Handle cancellations

- Grace period management
- 

## Phase 4: REST API (Week 4)

### 17. **includes/api/class-api-handler.php**

- Register REST routes
- Handle requests
- Format responses

### 18. **includes/api/class-api-authentication.php**

- API key validation
- JWT token validation
- Rate limiting

### 19. **includes/api/class-api-endpoints.php**

- Define all endpoints
  - `/validate-session`
  - `/check-license`
  - `/tenant-info`
- 

## Phase 5: Settings & Configuration (Week 4)

### 20. **admin/class-settings.php**

- Settings page
- Options registration
- Settings validation

### 21. **admin/views/settings.php**

- Settings form HTML
  - Tabbed interface
  - Save settings
-

## Helper/Utility Classes

These can be created as needed:

- **includes/class-installer.php** - Handle plugin installation/upgrade
  - **includes/class-uninstaller.php** - Clean uninstall process
  - **includes/class-cron-jobs.php** - Scheduled tasks
  - **includes/class-cache-manager.php** - Cache management
  - **admin/class-notices.php** - Admin notices system
  - **admin/class-help-tabs.php** - Context help tabs
- 

## Asset Files

**CSS Files to Create:**

**assets/css/admin.css**

```
css

/* Admin dashboard styles */
.novarax-dashboard {}
.novarax-tenant-card {}
.novarax-stats-widget {}
/* etc. */
```

**JavaScript Files to Create:**

**assets/js/admin.js**

```
javascript

// Admin functionality
// AJAX calls
// Form validation
// Charts
```

**assets/js/tenant-form.js**

```
javascript
```

```
// Real-time username availability check  
// Form validation  
// Auto-generate subdomain preview
```

---

## Testing Files

### tests/bootstrap.php

```
php  
  
// PHPUnit bootstrap file  
// Load WordPress test environment
```

### tests/test-database-manager.php

```
php  
  
// Test database operations  
// Test table creation  
// Test database provisioning
```

---

## Important Configuration Notes

### Required wp-config.php Additions

Add to `/var/www/app.novarax.ae/wp-config.php`:

```
php
```

```

/**
 * NovaRax Tenant Manager Configuration
 */

// Database root credentials for creating tenant databases
define('DB_ROOT_USER', 'root');
define('DB_ROOT_PASSWORD', 'your_root_password');

// Tenant configuration
define('NOVARAX_TENANT_CODEBASE_PATH', '/var/www/tenant-dashboard');
define('NOVARAX_SUBDOMAIN_SUFFIX', '.app.novarax.ae');

// Encryption key (generate unique key)
define('NOVARAX_ENCRYPTION_KEY', 'your-unique-32-character-key-here');

// API configuration
define('NOVARAX_API_ENABLED', true);
define('NOVARAX_API_RATE_LIMIT', 1000); // Requests per hour

```

## Development Workflow

### Step 1: Create Core Files

```

bash

cd /var/www/app.novarax.ae/wp-content/mu-plugins/
mkdir -p novarax-tenant-manager
cd novarax-tenant-manager

# Create main directories
mkdir -p includes/{api,integrations}
mkdir -p admin/{views/{partials},meta-boxes}
mkdir -p assets/{css,js,images}
mkdir -p languages
mkdir -p tests
mkdir -p docs

```

### Step 2: Copy Files

Copy the created PHP files to their respective locations.

### Step 3: Set Permissions

```
bash
```

```
chmod 755 -R /var/www/app.novarax.ae/wp-content/mu-plugins/novarax-tenant-manager/  
chmod 644 /var/www/app.novarax.ae/wp-content/mu-plugins/novarax-tenant-manager/*.php
```

### Step 4: Activate

MU-plugins are automatically active. Just refresh WordPress admin.

---

## Git Repository Structure

```
.git/  
.gitignore  
README.md  
CHANGELOG.md  
composer.json      # For dependencies  
package.json       # For npm scripts  
webpack.config.js # For asset compilation  
phpunit.xml        # PHPUnit configuration
```

### .gitignore Contents:

```
# WordPress  
/wordpress/  
/wp-content/*  
!/wp-content/mu-plugins/
```

```
# Dependencies  
/node_modules/  
/vendor/
```

```
# Build files  
/assets/css/*.min.css  
/assets/js/*.min.js
```

```
# Environment  
.env  
/backups/
```

```
# IDE
```

```
.vscode/  
.idea/  
*.sublime-*  
  
# OS  
.DS_Store  
Thumbs.db  
  
# Logs  
*.log  
/logs/
```

---

## Next Steps

1. **Create remaining core classes** (Validator, Logger, Security)
2. **Build admin interface** (Menu, list table, forms)
3. **Implement WooCommerce integration**
4. **Create REST API endpoints**
5. **Add email templates**
6. **Write tests**
7. **Create documentation**

Let me know which files you want me to create next!