NovaRax Tenant Manager - Installation & Setup Guide

📦 What We've Built So Far

✅ Completed Files (Core Utilities)

1. **novarax-tenant-manager.php** - Main plugin file
2. **includes/class-database-manager.php** - Database operations
3. **includes/class-tenant-operations.php** - Tenant CRUD operations
4. **includes/class-tenant-validator.php** - Input validation & sanitization
5. **includes/class-logger.php** - Logging system
6. **includes/class-security.php** - Encryption & security utilities
7. **includes/class-email-notifications.php** - Email templates & sending
8. **includes/class-provisioning-queue.php** - Asynchronous provisioning

---

🚀 Installation Steps

Step 1: Server Preparation

1.1 SSH into your server

```bash
ssh root@your-server-ip
```

1.2 Create directory structure

```bash
cd /var/www/app.novarax.ae/wp-content/mu-plugins/
mkdir -p novarax-tenant-manager/includes
mkdir -p novarax-tenant-manager/admin/views/partials
mkdir -p novarax-tenant-manager/assets/css
mkdir -p novarax-tenant-manager/assets/js
mkdir -p novarax-tenant-manager/assets/images
mkdir -p novarax-tenant-manager/languages
mkdir -p novarax-tenant-manager/templates/emails
```

Step 2: Upload Files

2.1 Upload all PHP files to the server:

```
/var/www/app.novarax.ae/wp-content/mu-plugins/novarax-tenant-manager/
├── novarax-tenant-manager.php
└── includes/
    ├── class-database-manager.php
    ├── class-tenant-operations.php
    ├── class-tenant-validator.php
    ├── class-logger.php
    ├── class-security.php
    ├── class-email-notifications.php
    └── class-provisioning-queue.php
```

2.2 Set correct permissions

```bash
cd /var/www/app.novarax.ae/wp-content/mu-plugins/
chmod -R 755 novarax-tenant-manager/
chmod 644 novarax-tenant-manager/*.php
chmod 644 novarax-tenant-manager/includes/*.php
chown -R www-data:www-data novarax-tenant-manager/
```

Step 3: Configure wp-config.php

Add these constants to your `/var/www/app.novarax.ae/wp-config.php` file:

```php
```

```
/**
 * NovaRax Tenant Manager Configuration
 */

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

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

// Security - Encryption key (IMPORTANT: Generate unique 32-character 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

// Debug mode (set to false in production)
define('NOVARAX_DEBUG', true);
```

Generate Encryption Key:

```
bash

php -r "echo bin2hex(random_bytes(16)) . PHP_EOL;"
```

Step 4: Database Setup

The plugin will automatically create database tables on activation. However, you can manually verify:

```
sql
```

```
-- Connect to MySQL
mysql -u root -p

-- Check if tables were created
USE your_database_name;
SHOW TABLES LIKE 'wp_novarax_%';

-- You should see:
-- wp_novarax_tenants
-- wp_novarax_modules
-- wp_novarax_tenant_modules
-- wp_novarax_api_keys
-- wp_novarax_audit_logs
-- wp_novarax_usage_stats
-- wp_novarax_logs
```

Step 5: Verify Installation

5.1 Check WordPress Admin

1. Go to `https://app.novarax.ae/wp-admin/`
2. Login with your admin credentials
3. Navigate to **Dashboard** → The plugin should be automatically active (MU-plugins are always active)

5.2 Verify Database Tables

Run this query to check tables:

```
sql

SELECT TABLE_NAME
FROM information_schema.TABLES
WHERE TABLE_SCHEMA = 'your_database_name'
AND TABLE_NAME LIKE 'wp_novarax_%';
```

5.3 Check Logs

```
bash
```

```
# Check if log directory was created
ls -la /var/www/app.novarax.ae/wp-content/uploads/novarax-logs/

# View logs
tail -f /var/www/app.novarax.ae/wp-content/uploads/novarax-logs/novarax-info-*.log
```

🧪 Testing the Installation

Test 1: Create a Test Tenant (via PHP)

Create a test file: `/var/www/app.novarax.ae/test-tenant-creation.php`

```php
<?php
require_once __DIR__ . '/wp-load.php';

// Test tenant creation
$tenant_ops = new NovaRax_Tenant_Operations();

$test_data = array(
    'full_name' => 'John Doe',
    'email' => 'john@example.com',
    'username' => 'johndoe',
    'password' => 'SecurePassword123!@#',
    'company_name' => 'Doe Enterprises',
    'phone_number' => '+1234567890',
);

echo "Creating test tenant...\n";

$result = $tenant_ops->create_tenant($test_data);

if ($result['success']) {
    echo "✅ SUCCESS!\n";
    echo "Tenant ID: " . $result['tenant_id'] . "\n";
    echo "User ID: " . $result['user_id'] . "\n";
    echo "Subdomain: " . $result['subdomain'] . "\n";
} else {
    echo "❌ FAILED!\n";
    echo "Error: " . $result['error'] . "\n";
}
```

Run the test:

```
bash

cd /var/www/app.novarax.ae/
php test-tenant-creation.php
```

Test 2: Validate Username Availability (AJAX Test)

Test the validator:

```
php

<?php
require_once __DIR__ . '/wp-load.php';

$validator = new NovaRax_Tenant_Validator();

// Test valid username
$result = $validator->check_username_availability('testuser123');
print_r($result);

// Test invalid username
$result = $validator->check_username_availability('admin');
print_r($result);
```

Test 3: Test Logger

```
php

<?php
require_once __DIR__ . '/wp-load.php';

NovaRax_Logger::info('Test info message');
NovaRax_Logger::warning('Test warning message');
NovaRax_Logger::error('Test error message');

echo "Check logs at: wp-content/uploads/novarax-logs/\n";
```

Test 4: Test Security Functions

```
php
```

```
<?php
require_once __DIR__ . '/wp-load.php';

// Test encryption
$data = 'Sensitive information';
$encrypted = NovaRax_Security::encrypt($data);
$decrypted = NovaRax_Security::decrypt($encrypted);

echo "Original: {$data}\n";
echo "Encrypted: {$encrypted}\n";
echo "Decrypted: {$decrypted}\n";
echo "Match: " . ($data === $decrypted ? 'YES' : 'NO') . "\n";

// Test JWT
$payload = array(
    'user_id' => 123,
    'username' => 'testuser',
);

$token = NovaRax_Security::generate_jwt($payload, 3600);
echo "\nJWT Token: {$token}\n";

$verified = NovaRax_Security::verify_jwt($token);
print_r($verified);
```

Test 5: Test Email (Send Test Email)

```php
<?php
require_once __DIR__ . '/wp-load.php';

$result = NovaRax_Email_Notifications::send_test_email('your-email@example.com');

if ($result) {
    echo "✅ Test email sent successfully!\n";
} else {
    echo "❌ Failed to send test email\n";
}
```

⚙️ Configuration Options

Email Settings

Add to WordPress admin or wp-config.php:

```php
// Email configuration
update_option('novarax_tm_from_email', 'noreply@novarax.ae');
update_option('novarax_tm_from_name', 'NovaRax Platform');
update_option('novarax_tm_email_logo', 'https://novarax.ae/logo.png');
update_option('novarax_tm_email_primary_color', '#0073aa');
```

Tenant Defaults

```php
// Tenant default settings
update_option('novarax_tm_tenant_storage_limit', 5368709120); // 5GB
update_option('novarax_tm_tenant_user_limit', 10);
update_option('novarax_tm_subdomain_suffix', '.app.novarax.ae');
update_option('novarax_tm_tenant_db_prefix', 'novarax_tenant_');
update_option('novarax_tm_auto_provision', true);
update_option('novarax_tm_grace_period_days', 7);
```

Logging Settings

```php
// Logging configuration
update_option('novarax_tm_log_to_database', true);
update_option('novarax_tm_log_to_file', true);
update_option('novarax_tm_log_retention_days', 30);
```

🔧 Troubleshooting

Issue 1: Tables Not Created

Solution:

```bash
```

```
# Manually trigger table creation
cd /var/www/app.novarax.ae/
wp eval 'do_action("activate_plugin_novarax-tenant-manager/novarax-tenant-manager.php");'
```

Or via PHP:

```
php

<?php
require_once __DIR__ . '/wp-load.php';
$db_manager = new NovaRax_Database_Manager();
$db_manager->create_tables();
```

Issue 2: Permission Denied Errors

```
bash

# Fix permissions
chown -R www-data:www-data /var/www/app.novarax.ae/
chmod -R 755 /var/www/app.novarax.ae/wp-content/
chmod 644 /var/www/app.novarax.ae/wp-config.php
```

Issue 3: Can't Create Tenant Databases

Check MySQL permissions:

```
sql

SHOW GRANTS FOR 'root'@'localhost';

-- Grant all privileges if needed
GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Issue 4: Logs Not Being Created

```
bash

# Create log directory manually
mkdir -p /var/www/app.novarax.ae/wp-content/uploads/novarax-logs
chmod 755 /var/www/app.novarax.ae/wp-content/uploads/novarax-logs
chown www-data:www-data /var/www/app.novarax.ae/wp-content/uploads/novarax-logs
```

Issue 5: Emails Not Sending

Test SMTP configuration:

```bash
# Install WP Mail SMTP plugin
wp plugin install wp-mail-smtp --activate

# Configure SMTP settings in WordPress admin
```

📝 Next Steps

Now that the core utilities are installed, we need to create:

Phase 2: Admin Interface (Next Priority)

1. ✅ **class-admin-interface.php** - Admin menu and pages
2. ✅ **class-tenant-list-table.php** - Tenant list table
3. ✅ **admin/views/tenant-list.php** - Tenant list page
4. ✅ **admin/views/tenant-add.php** - Add tenant form
5. ✅ **admin/views/tenant-edit.php** - Edit tenant form
6. ✅ **admin/views/dashboard.php** - Main dashboard

Phase 3: WooCommerce Integration

1. **class-woocommerce-integration.php** - WooCommerce hooks
2. **class-subscription-handler.php** - Subscription management

Phase 4: REST API

1. **class-api-handler.php** - API routing
2. **class-api-endpoints.php** - API endpoints
3. **class-api-authentication.php** - API authentication

📊 Current System Capabilities

With what we've built, the system can now:

✅ Tenant Management

- Create new tenants with full validation
- Store tenant data in isolated database records
- Track tenant status (pending, active, suspended, cancelled)
- Update tenant information
- Delete tenants (soft or hard delete with backup)

✅ **Database Operations**

- Create separate databases for each tenant
- Import WordPress schema automatically
- Manage database users and permissions
- Backup and restore tenant databases
- Track database size

✅ **Security**

- Encrypt sensitive data (AES-256-CBC)
- Generate and verify JWT tokens
- Rate limiting
- Brute force protection
- IP blocking
- XSS detection
- Security headers

✅ **Logging**

- Log to database and files
- Multiple log levels (debug, info, warning, error, critical)
- Automatic log rotation
- Export logs to CSV
- Clean old logs
- View logs by level, user, date

✅ **Validation**

- Username availability checking
- Email validation (including disposable email detection)
- Password strength checking
- Phone number validation
- Subdomain validation
- Input sanitization

✅ **Email Notifications**

- Welcome emails
- Activation emails
- Renewal reminders
- Expiration warnings
- Cancellation notifications
- Suspension notifications
- Password reset emails
- Support ticket emails
- Beautiful HTML templates

✅ **Provisioning Queue**

- Asynchronous tenant provisioning
- Automatic retry on failure
- Queue statistics
- Manual queue management
- Prevents timeout issues

🎉 Summary

You now have a solid foundation for the NovaRax Tenant Manager!

The core infrastructure is complete and ready for:

- Creating admin interfaces

- Integrating with WooCommerce
- Building REST APIs
- Developing tenant dashboards

Would you like me to continue with the Admin Interface next?

This will give you the WordPress admin pages to manage tenants through a user-friendly interface!