




# DasHouse Supabase Integration Setup Guide

---

## Prerequisites

-  Supabase account created
-  DasHouse project created
-  Database password: `slhAvH0Qyx1R4ifB`

## Step 1: Get Your Supabase Project Credentials

### 1.1 Access Your Project

1. Go to [supabase.com](https://supabase.com) and sign in
2. Click on your "DasHouse" project
3. Navigate to **Settings** → **API** in the left sidebar

### 1.2 Copy Required Information

You'll need these values from your project:

- **Project URL:** `https://[YOUR-PROJECT-ID].supabase.co`
- **Anon (public) key:** Starts with `eyJ...`
- **Service role key:** Starts with `eyJ...` (keep this secret!)

### 1.3 Database Connection Details

- **Host:** `db.[YOUR-PROJECT-ID].supabase.co`
- **Database:** `postgres`
- **User:** `postgres`
- **Password:** `slhAvH0Qyx1R4ifB`
- **Port:** `5432`

## Step 2: Update Configuration File

### 2.1 Edit `admin/config/supabase.php`

Replace the placeholder values with your actual credentials:

```
define('SUPABASE_URL', 'https://[YOUR-PROJECT-ID].supabase.co');
define('SUPABASE_ANON_KEY', '[YOUR-ANON-KEY]');
define('SUPABASE_SERVICE_ROLE_KEY', '[YOUR-SERVICE-ROLE-KEY]');
define('SUPABASE_DB_HOST', 'db.[YOUR-PROJECT-ID].supabase.co');
```

### 2.2 Example with Real Values

```
define('SUPABASE_URL', 'https://abcdefghijklmnop.supabase.co');
define('SUPABASE_ANON_KEY', 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...');
define('SUPABASE_SERVICE_ROLE_KEY',
'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...');
define('SUPABASE_DB_HOST', 'db.abcd efghijklmnop.supabase.co');
```

## Step 3: Set Up Database Tables

### 3.1 Run the Setup Script

1. Open your browser and go to: [your-domain.com/admin/setup-database.php](#)
2. Click the "Setup Database" button
3. The script will create all necessary tables automatically

### 3.2 Verify Tables Created

The setup script will create these tables:

- **users** - Customer and admin user accounts
- **categories** - Menu categories (Breakfast, Snacks, Beverages, etc.)
- **menu\_items** - Individual menu items with prices
- **orders** - Customer orders
- **order\_items** - Items within each order
- **reservations** - Table reservations
- **photos** - Photo gallery management
- **settings** - Restaurant configuration

## Step 4: Test the Integration

### 4.1 Test Database Connection

Visit [admin/setup-database.php](#) and check if you see:

```
Database connection successful!
Database setup completed!
```

### 4.2 Test Admin Panel

1. Go to [admin/index.php](#)
2. Login with: [admin](#) / [dashouse2024](#)
3. Check if the dashboard loads without errors

## Step 5: Import Your Existing Menu Data

### 5.1 Create a Data Import Script

The setup includes default categories. You can now:

1. Use the admin panel to add menu items
2. Upload photos through the photo manager
3. Configure restaurant settings

## 5.2 Sample Menu Item Insertion

```
// Example: Add a menu item
$menuItem = [
    'category_id' => 1, // Breakfast & Waffles
    'name' => 'Wiener Frühstück | Viennese Breakfast',
    'description' => 'Semmel, Butter, Marmelade / Breadroll, Butter, Jam',
    'price' => 6.90,
    'is_vegetarian' => true,
    'sort_order' => 1
];

$db->insert('menu_items', $menuItem);
```

## Step 6: Frontend Integration

### 6.1 Update Your Main Page

You can now fetch menu data dynamically:

```
// Example: Fetch menu items from Supabase
fetch('/admin/api/menu-items.php')
    .then(response => response.json())
    .then(data => {
        // Update your menu display
        displayMenu(data);
    });
```

### 6.2 Create API Endpoints

The admin panel includes API endpoints for:

- Menu items
- Categories
- Photos
- Settings

## Troubleshooting

### Common Issues

#### 1. Database Connection Failed

- Check your database password

- Verify the host URL format
- Ensure your IP is whitelisted in Supabase

## 2. Tables Not Created

- Check PHP error logs
- Verify database permissions
- Ensure PDO PostgreSQL extension is enabled

## 3. API Calls Failing

- Verify your API keys
- Check CORS settings in Supabase
- Ensure proper authentication headers

## Getting Help

- Check Supabase documentation: [supabase.com/docs](https://supabase.com/docs)
- Review PHP error logs
- Test database connection separately

## Security Notes

### Important Security Practices

1. **Never commit API keys to version control**
2. **Use environment variables in production**
3. **Restrict database access to necessary IPs only**
4. **Regularly rotate your service role key**

## Production Deployment

Before going live:

1. Set `IS_DEVELOPMENT = false` in config
2. Use environment variables for sensitive data
3. Enable SSL connections
4. Set up proper CORS policies

## Next Steps

After successful integration:

1. **Customize the admin panel** to match your needs
2. **Add more menu items** through the admin interface
3. **Set up photo management** for your gallery
4. **Configure reservation system** if needed
5. **Test all functionality** thoroughly

## Support

If you encounter issues:

1. Check this guide first
  2. Review Supabase documentation
  3. Check PHP error logs
  4. Test database connection manually
- 

**Remember:** Keep your database password and API keys secure and never share them publicly!