08 - Developer Tools & Deployment

# 🗂 Folder Structure

erpnext\_shop\_sync/  
├── erpnext\_shop\_sync/  
│ ├── \_\_init\_\_.py  
│ ├── hooks.py  
│ ├── config/  
│ │ └── desktop.py  
│ ├── modules/  
│ │ └── shop\_sync/  
│ │ ├── doctype/  
│ │ │ ├── shop\_server/  
│ │ │ ├── sync\_log/  
│ │ └── page/  
│ │ └── shop\_sync\_dashboard/  
│ └── scripts/  
│ ├── push\_master\_data.py  
│ ├── pull\_shop\_data.py  
│ └── sync\_queue.py  
├── public/  
│ └── js/  
├── templates/  
│ └── pages/  
└── README.md

# ⚙️ Installation Steps

1. Clone repo inside `frappe-bench/apps`  
2. Run:  
 bench get-app erpnext\_shop\_sync [GIT\_URL]  
 bench --site [yoursite] install-app erpnext\_shop\_sync

# ✅ Developer Setup Checklist

- Create API user per shop (role: Shop Sync Agent)  
- Assign API Key & Secret  
- Whitelist shop IP (optional)  
- Ensure site has required doctypes installed (Shop Server, Sync Log)  
- Run sync script manually to test connection

# 🧪 Unit Testing Example

# test\_shop\_server.py  
import frappe  
import unittest  
  
class TestPushItem(unittest.TestCase):  
 def test\_sync\_item\_structure(self):  
 item = frappe.get\_doc("Item", {"item\_code": "TEST-ITEM"})  
 self.assertIn("item\_code", item)  
 self.assertEqual(item.doctype, "Item")

# 🚀 Deployment Best Practices

- Use environment variables to store secrets  
- Separate production vs staging sync apps  
- Log all API calls  
- Backup before first sync  
- Use a sync test environment for validation