Local Configuration System

Overview

A robust local configuration management system has been implemented to preserve system-specific settings when updating from GitHub. This ensures your local device configurations, network settings, and preferences are never overwritten by git pulls.

Problem Solved

Previously, pulling updates from GitHub could potentially overwrite local system configurations, requiring manual reconfiguration after each update. This system separates:

- **Shared code and templates** (tracked in Git) - Safe to update
- **Local system settings** (gitignored) - Preserved across updates

Configuration Files

Template Files (Tracked in Git) ✅

Located in `config/` directory:

1. **`local.template.json`**
   - System settings (network, ports, features)
   - Device connection parameters
   - Logging and backup configuration
   - Default values for new installations

2. **`devices.template.json`**
   - Device inventory structure
   - Wolfpack matrix layout
   - IR device templates
   - Streaming device configs

3. **`sports-teams.template.json`**
   - Team monitoring settings
   - League preferences
   - Notification configuration

Local Files (Gitignored) ❌

Same names with `.local.json` extension:

- `config/local.local.json` - Your system settings
- `config/devices.local.json` - Your device inventory
- `config/sports-teams.local.json` - Your team preferences

These files are automatically created from templates and NEVER committed to Git.

Quick Start

Initial Setup

```
cd /home/ubuntu/Sports-Bar-TV-Controller

# Create local config files from templates
./scripts/init-local-config.sh

# Edit for your system
nano config/local.local.json
nano config/devices.local.json
nano config/sports-teams.local.json
```

After GitHub Updates

Your local configuration is automatically preserved:

```
git pull origin main

# Your local files remain untouched!
# If new options were added, run:
./scripts/init-local-config.sh  # Only adds missing options, keeps existing values
```

Configuration Structure

`local.local.json`

```
{
  "system": {
    "name": "Your Bar Name",
    "location": "City, State",
    "timezone": "America/New_York"
  },
  "network": {
    "apiPort": 3000,
    "devPort": 3001,
    "allowedOrigins": ["http://localhost:3000"]
  },
  "wolfpack": {
    "ip": "192.168.1.100",
    "port": 4999,
    "protocol": "tcp",
    "enabled": true
  },
  "features": {
    "aiAnalysis": true,
    "autoScheduling": true,
    "sportsFinder": true
  }
}
```

`devices.local.json`

```
{
  "wolfpack": {
    "inputs": [
      {
        "id": "input-1",
        "channelNumber": 1,
        "label": "DirecTV Main",
        "deviceType": "directv"
      }
    ],
    "outputs": [
      {
        "id": "output-1",
        "channelNumber": 1,
        "label": "TV 1 - Main Bar"
      }
    ]
  },
  "directv": {
    "receivers": [
      {
        "ip": "192.168.1.101",
        "name": "Main Receiver"
      }
    ]
  }
}
```

How It Works

File Priority

1. **Template** (`*.template.json`) - Default values
2. **Local** (`*.local.json`) - Your overrides
3. **Environment** (`.env`) - Secrets and sensitive data
4. **Database** - Runtime configurations from UI

Later sources override earlier ones.

Git Ignore Rules

Added to `.gitignore` :

```
# local configuration files
config/*.local.json
config/*.local.js
```

Initialization Script

`scripts/init-local-config.sh` :
- Creates `.local.json` files from templates
- Only creates missing files (safe to run multiple times)
- Preserves existing local configuration
- Reports what was created/skipped

Configuration Workflow

For End Users (Bar Owners/Managers)

1. **Initial Setup:**
   ```
   bash
     ./scripts/init-local-config.sh
   ```

2. **Configure via Web UI:**
   - Navigate to Settings pages
   - Add devices, configure network
   - Settings automatically save to `.local.json` files

3. **Update System:**
   ```
   bash
     git pull origin main
     yarn install  # If dependencies changed
     pm2 restart sports-bar-app
   ```
   Your local config is preserved!

For Developers

1. **Adding New Configuration Options:**
   - Update `*.template.json` with new options
   - Document in `config/README.md`
   - Commit templates to Git
   - End users run `init-local-config.sh` to merge new options

2. **Testing Configuration:**
   ```bash
   # View current config
   cat config/local.local.json

# Validate JSON
cat config/local.local.json | jq .

# Test with config
yarn dev
```

Security

What Goes Where?

Template files (`*.template.json`):
- Non-sensitive defaults
- Structure and examples
- Safe for public repositories

Local files (`*.local.json`):
- System-specific values
- Internal IP addresses
- Device identifiers
- NOT sensitive credentials

Environment file (`.env`):
- API keys
- Database passwords
- Session secrets
- OAuth tokens

Never commit:
- `.env` files
- `.local.json` files
- Database files (`*.db`)
- Uploaded files

Backup

Local configuration is backed up automatically:

```
# Location
/home/ubuntu/Sports-Bar-TV-Controller/backups/

# Manual backup
tar -czf ~/config-backup-$(date +%Y%m%d).tar.gz \
  config/*.local.json \
  .env \
  prisma/dev.db
```

Migration Guide

From Old System (Manual Config)

If you previously had device configurations in code:

1. **Export current settings:**
   ```
   bash
      # Via UI: Settings → Export Configuration
      # Or manually copy from database
   ```

2. **Initialize local config:**
   ```
   bash
      ./scripts/init-local-config.sh
   ```

3. **Import settings:**
   ```
   bash
      # Via UI: Settings → Import Configuration
      # Or manually edit config/*.local.json
   ```

4. **Verify:**
   ```
   bash
      # Check devices appear in UI
      # Test TV control
      # Verify schedules work
   ```

Troubleshooting

Configuration Not Loading

Check file exists:

```
ls -la config/*.local.json
```

Validate JSON syntax:

```
cat config/local.local.json | jq .
```

Check application logs:

```
pm2 logs sports-bar-app --lines 100 | grep -i config
```

Lost Configuration After Update

If you accidentally committed local files:

```
# Remove from Git (keeps local files)
git rm --cached config/*.local.json
git commit -m "Remove local config from tracking"
git push
```

Restore from backup:

```
cd /home/ubuntu/Sports-Bar-TV-Controller
tar -xzf ~/config-backup-YYYYMMDD.tar.gz
```

Missing New Configuration Options

After pulling updates with new config fields:

```
# Re-run init (safe, only adds missing options)
./scripts/init-local-config.sh

# Compare with template to see what's new
diff config/local.template.json config/local.local.json
```

Best Practices

✅ DO:

- Run `init-local-config.sh` after first clone
- Edit `.local.json` files for your system
- Keep templates as reference
- Back up `.local.json` files regularly
- Document custom settings in comments

❌ DON'T:

- Commit `.local.json` files to Git
- Put API keys in config files (use `.env`)
- Edit `.template.json` with system-specific values
- Delete template files
- Modify the gitignore rules

Files Reference

Added/Modified Files

```
Sports-Bar-TV-Controller/
├── .gitignore (updated)
│   └── Added: config/*.local.json exclusion
├── config/
│   ├── README.md (new)
│   ├── local.template.json (new)
│   ├── devices.template.json (new)
│   ├── sports-teams.template.json (new)
│   ├── local.local.json (created locally, gitignored)
│   ├── devices.local.json (created locally, gitignored)
│   └── sports-teams.local.json (created locally, gitignored)
└── scripts/
    └── init-local-config.sh (new)
```

File Sizes

- Templates: ~1-5 KB each
- Local files: Varies with configuration
- README: ~5 KB

Integration with Existing Features

Database vs Config Files

- **Database:** Runtime data, user-created content (schedules, logs)
- **Config files:** System structure, device inventory, preferences

Web UI Integration

Configuration pages automatically read/write to `.local.json` files:

- **Matrix Control:** Saves to `devices.local.json`
- **Settings:** Saves to `local.local.json`
- **Sports Teams:** Saves to `sports-teams.local.json`

API Endpoints

Config is accessible via API (when needed):

```
// GET /api/config/system
// GET /api/config/devices
// POST /api/config/update
```

Future Enhancements

Planned Features

1. **Config import/export UI** - Download/upload config files
2. **Config validation** - Check for errors before saving
3. **Version migration** - Automatic config updates for new versions
4. **Multi-location support** - Manage configs for multiple bars
5. **Cloud backup** - Optional sync to cloud storage

Versioning

When template structure changes:

```
{
  "version": "2.0.0",
  "system": { ... }
}
```

Migration scripts will handle version upgrades automatically.

Support

For help with configuration:

1. **Read this doc:** `LOCAL_CONFIG_SYSTEM.md`
2. **Check config README:** `config/README.md`
3. **View examples:** `config/*.template.json`
4. **Check logs:** `pm2 logs sports-bar-app`
5. **GitHub issues:** Open an issue with `[config]` tag

Summary

This configuration system provides:

- ✅ Safe GitHub updates without losing local settings
- ✅ Clear separation of shared vs local configuration
- ✅ Easy setup with initialization script
- ✅ Automatic preservation of local files
- ✅ Template-based defaults for new installations
- ✅ Comprehensive documentation

Your local configuration is now protected from Git updates!

Implemented: October 1, 2025
Version: 1.0.0
Status: Active