

API Keys & Subscriptions - Complete Backup Guide

Overview

YES! ALL your API keys, credentials, and subscription configurations are now automatically backed up and preserved during updates.

What API Keys & Credentials Are Backed Up

1. AI Provider API Keys (Database)

Location: `prisma/dev.db` → `ApiKey` table

Stored in the database with encryption:

- **Claude API Keys**
 - Provider: Anthropic
 - Used for: AI-powered device insights, sports team search
 - Managed via: `/ai-keys` page
- **ChatGPT/OpenAI API Keys**
 - Provider: OpenAI
 - Used for: AI chat, bar layout analysis
 - Managed via: `/ai-keys` page
- **Grok API Keys**
 - Provider: X.AI
 - Used for: Alternative AI provider
 - Managed via: `/ai-keys` page
- **Local AI Settings**
 - Provider: Ollama (local)
 - Used for: Privacy-focused AI processing
 - Managed via: `/ai-keys` page

Status:  Automatically backed up (in database)

2. Soundtrack Your Brand API (Database)

Location: `prisma/dev.db` → `SoundtrackConfig` table


Stored credentials:

- API Key/Token
- Account ID
- Account Name (business name)

- Connection status
- Last tested timestamp

Used for: Background music management and playback control

Managed via: `/soundtrack` or `/audio` page

Status:  Automatically backed up (in database)

3. Streaming Service Credentials (Data Files)

Location: `data/streaming-credentials.json`


Format:

```
[
  {
    "id": "cred_1759239456681",
    "platformId": "nfhs-network",
    "username": "your-email@example.com",
    "passwordHash": "encrypted-password",
    "encrypted": true,
    "lastUpdated": "2025-09-30T13:37:36.681Z",
    "status": "active",
    "lastSync": "2025-09-30T13:37:36.681Z"
  }
]
```

Supported Services:

- **NFHS Network** (High School Sports)
- Username/email
- Encrypted password
- Auto-sync status

- **Future streaming services** can be added

Status:  Automatically backed up (data file)

4. Device Subscription Configurations (Data Files)

Location: `data/device-subscriptions.json`

Tracks which premium services are active on each device:

DirecTV Subscriptions:

- DIRECTV CHOICE™
- Sports Pack
- NFL Sunday Ticket
- NBA League Pass
- NHL Center Ice
- MLB Extra Innings
- Custom sports packages

Cable/Spectrum Subscriptions:

- Basic packages
- Sports tier

- Premium channels
- Regional sports networks

Streaming Service Subscriptions:

- ESPN+
- Paramount+
- Peacock Premium
- YouTube TV
- FuboTV
- DAZN

Format:

```
{
  "devices": [
    {
      "deviceId": "directv_1759187217790",
      "deviceType": "directv",
      "deviceName": "Direct TV 1",
      "subscriptions": [
        {
          "id": "sports-pack",
          "name": "Sports Pack",
          "type": "sports",
          "status": "active",
          "provider": "DIRECTV",
          "cost": 14.99,
          "description": "NFL RedZone, NBA TV, NHL Network..."
        }
      ]
    }
  ]
}
```

Status: ☒ Automatically backed up (data file)

5. ☒ External Service API Keys (.env)

Location: .env file

Environment-based API keys:

- Wolfpack host/port
- Database URL
- NextAuth secrets
- Custom integrations
- Third-party APIs

Example:

```
# Wolfpack Matrix
WOLFPACK_HOST=192.168.1.100
WOLFPACK_PORT=23

# Security
ENCRYPTION_KEY=your-encryption-key

# API Base
API_BASE_URL=http://localhost:3000
```


Status:  Automatically backed up (.env file)

6. Gracenote/Spectrum API Credentials

Location: Configuration stored in database or local config

If you're using:

- Spectrum Business API
- Gracenote TV listings
- TMS (Tribune Media Services)

Status:  Automatically backed up (database/config)

7. TV Provider Guide Subscriptions (Database)

Location: `prisma/dev.db` → `TVProvider` table

Your configured TV providers:

- Provider name (e.g., "Spectrum Business & Sports Package")
- Provider type (cable, satellite, streaming)
- Channel lineup
- Subscription tier
- Regional sports networks

Status:  Automatically backed up (in database)

How Backup Works

Automatic Backup on Every Update

```
./update_from_github.sh
```

Creates backup of:

```
~/sports-bar-backups/config-backup-YYYYMMDD-HHMMSS.tar.gz
├─ config/*.local.json      ← System settings
├─ .env                    ← Environment API keys
├─ prisma/dev.db           ← All database API keys
├─ data/streaming-credentials.json ← Streaming logins
├─ data/device-subscriptions.json ← Subscription configs
├─ data/directv-devices.json ← DirecTV configs
├─ data/firetv-devices.json ← FireTV configs
├─ data/ir-devices.json    ← IR device configs
├─ data/tv-layout.json     ← Layout configs
├─ data/scene-logs/       ← Scene history
└─ data/atlas-configs/    ← Atlas presets
```

Retention Policy

- **Automatic cleanup:** Keeps last 7 backups
- **Location:** `~/sports-bar-backups/`
- **Frequency:** Every time you run update script

Manual Backup

Create an additional backup anytime:

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

# Quick API keys backup
tar -czf ~/api-keys-backup-$(date +%Y%m%d).tar.gz \
  prisma/dev.db \
  data/streaming-credentials.json \
  data/device-subscriptions.json \
  .env

# Full backup with everything
tar -czf ~/full-backup-$(date +%Y%m%d).tar.gz \
  config/ \
  .env \
  prisma/ \
  data/ \
  uploads/
```

Protected by .gitignore

All credential files are protected from Git:

```
# .gitignore entries
.env
.env*.local
config/*.local.json
prisma/*.db
prisma/*.db-journal
data/*.json      ← NEW!
data/scene-logs/ ← NEW!
data/atlas-configs/*.json ← NEW!
```

What this means:

-  Never committed to GitHub

- ✓ Never overwritten by `git pull`
- ✓ Never shared publicly
- ✓ Always stay on your local machine

Template System

How It Works

GitHub stores **templates**, not actual data:

```
data/
├── streaming-credentials.template.json ← In Git
├── streaming-credentials.json         ← Your data (NOT in Git)
├── device-subscriptions.template.json ← In Git
├── device-subscriptions.json          ← Your data (NOT in Git)
└── ...
```

On first install/update:

```
./update_from_github.sh

📁 Initializing data files...
✓ Created streaming-credentials.json from template
✓ Created device-subscriptions.json from template
```

After you configure:

- Your actual credentials go in `.json` files
- Templates remain unchanged
- Your data is preserved forever

Restore from Backup

Restore All API Keys

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

# Stop the server
kill -f "npm.*start"

# Restore from latest backup
tar -xzf ~/sports-bar-backups/config-backup-YYYYMMDD-HHMMSS.tar.gz

# Restart server
npm start
```

Restore Specific Items

```
# Restore only AI API keys (database)
tar -xzf ~/sports-bar-backups/config-backup-YYYYMMDD-HHMMSS.tar.gz prisma/dev.db

# Restore only streaming credentials
tar -xzf ~/sports-bar-backups/config-backup-YYYYMMDD-HHMMSS.tar.gz data/streaming-credentials.json

# Restore only .env
tar -xzf ~/sports-bar-backups/config-backup-YYYYMMDD-HHMMSS.tar.gz .env
```



Verification



Check Current API Keys

```
cd /home/ubuntu/Sports-Bar-TV-Controller
./test-preservation.sh
```

Output includes:

```
System:
  API Keys: 4

Data Files:
  device-subscriptions.json: 
  streaming-credentials.json: 
  Streaming credentials count: 1

Configuration Files:
  local.local.json exists: 
  .env exists: 
```

Verify Database API Keys

```
sqlite3 prisma/dev.db <<EOF
SELECT name, provider, isActive
FROM ApiKey;
EOF
```

Example output:

```
Claude|anthropic|1
ChatGPT|openai|1
Grok|xai|1
Local AI|ollama|1
```

Verify Streaming Credentials

```
cat data/streaming-credentials.json | jq '[] | {platform: .platformId,
status: .status}'
```

Example output:

```
{
  "platform": "nfhs-network",
  "status": "active"
}
```

Security

Encryption

- **AI API Keys:** Encrypted in database using `ENCRYPTION_KEY` from `.env`
- **Streaming passwords:** Base64 encoded with encryption flag
- **Database:** SQLite with file permissions (600)
- **Backups:** Compressed tar.gz with restricted permissions

Best Practices

1. Keep backups secure:

```
bash
chmod 600 ~/sports-bar-backups/*.tar.gz
```

2. Regular external backups:

```
```bash
Copy to external drive
cp ~/sports-bar-backups/config-backup-*.tar.gz /media/backup-drive/
```

# Or upload to secure cloud storage

```
rclone copy ~/sports-bar-backups/ remote:sports-bar-backups/
```
```

1. **Don't share backups** - they contain sensitive credentials
2. **Rotate encryption keys periodically** (advanced)

Migration Example

Scenario: New Server Setup

Old server:

```
cd /home/ubuntu/Sports-Bar-TV-Controller
./update_from_github.sh # Creates backup
# Backup is at: ~/sports-bar-backups/config-backup-20251001-120000.tar.gz
```

Copy backup to new server:

```
scp ~/sports-bar-backups/config-backup-20251001-120000.tar.gz user@newserver:/tmp/
```

New server:











```
# Install app
cd /home/ubuntu
git clone https://github.com/your-username/Sports-Bar-TV-Controller.git
cd Sports-Bar-TV-Controller
./install.sh

# Restore your data
cd /home/ubuntu/Sports-Bar-TV-Controller
tar -xzf /tmp/config-backup-20251001-120000.tar.gz

# Start app
npm start
```

Result: All API keys and subscriptions restored! 

What Gets Backed Up Summary

| Type | Location | Backup | Gitignore |
|-----------------------|---------------------------------|---|---|
| AI API Keys | prisma/dev.db |  |  |
| Soundtrack API | prisma/dev.db |  |  |
| Streaming Credentials | data/streaming-credentials.json |  |  |
| Device Subscriptions | data/device-subscriptions.json |  |  |
| Environment Keys | .env |  |  |
| TV Providers | prisma/dev.db |  |  |
| Device Configs | data/*.json |  |  |
| Scene Logs | data/scene-logs/ |  |  |
| Atlas Configs | data/atlas-configs/ |  |  |
| Layout Files | uploads/ |  |  |

Testing

Test Backup & Restore

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

# 1. Check current state
./test-preservation.sh > before.txt

# 2. Run update (creates backup)
./update_from_github.sh

# 3. Verify preserved
./test-preservation.sh > after.txt

# 4. Compare (should be identical)
diff before.txt after.txt
```

Expected result: No differences! All API keys preserved.

Test Template Initialization

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

# 1. Rename actual data file
mv data/streaming-credentials.json data/streaming-credentials.backup

# 2. Run update
./update_from_github.sh

# Result: Template is copied to create new file
# 📁 Initializing data files...
# ✅ Created streaming-credentials.json from template

# 3. Restore your actual data
mv data/streaming-credentials.backup data/streaming-credentials.json
```

Common Questions

Q: Are API keys visible in GitHub?

A: No! All API keys are in `.gitignore` and never pushed to GitHub.

Q: What if I lose my backup?

A: The files still exist on your local machine:

- `prisma/dev.db` (AI keys, Soundtrack API)
- `data/streaming-credentials.json` (streaming logins)
- `.env` (environment keys)

Just create a new backup: `./update_from_github.sh`

Q: Can I share my GitHub repo publicly?

A: Yes! None of your credentials are in Git. Only code and templates.

Q: How do I change an API key?

A:

- **AI keys:** Go to `/ai-keys` page
- **Soundtrack:** Go to `/soundtrack` page
- **Streaming:** Edit `data/streaming-credentials.json`
- **Environment:** Edit `.env` file

Changes are automatically preserved on next update.

Q: What if I reset my database?

A: Your API keys are GONE. Restore from backup:

```
tar -xzf ~/sports-bar-backups/config-backup-LATEST.tar.gz prisma/dev.db
```

Q: Can I automate backups?

A: Yes! Add to cron:

```
# Daily backup at 2 AM
0 2 * * * cd /home/ubuntu/Sports-Bar-TV-Controller && tar -czf ~/daily-backup-$(date +
\%Y\%m\%d).tar.gz config/ .env prisma/dev.db data/
```

Troubleshooting

Missing API Keys After Update

Check backup:

```
tar -tzf ~/sports-bar-backups/config-backup-LATEST.tar.gz | grep "dev.db"
```

Restore:

```
cd /home/ubuntu/Sports-Bar-TV-Controller
tar -xzf ~/sports-bar-backups/config-backup-LATEST.tar.gz prisma/dev.db
```

Streaming Credentials Not Working

Verify file exists:

```
ls -la data/streaming-credentials.json
cat data/streaming-credentials.json | jq .
```

Check encryption:

```
# Should show encrypted: true
cat data/streaming-credentials.json | jq '.[].encrypted'
```

Soundtrack API Not Found

Check database:

```
sqlite3 prisma/dev.db "SELECT * FROM SoundtrackConfig;"
```

If empty, configure via app:

1. Go to `/soundtrack` page
2. Enter API key
3. Test connection
4. Save

Summary








Your Question

“Are all API keys and app subscriptions able to be saved like this and backed up?”






The Answer

YES! 

Everything is backed up:

1.  **AI API Keys** (Claude, ChatGPT, Grok, Local AI) - Database
2.  **Soundtrack API** credentials - Database
3.  **Streaming service** logins (NFHS, etc.) - Data files
4.  **Device subscriptions** (DirecTV, Cable, Streaming) - Data files
5.  **Environment API keys** (.env) - .env file
6.  **TV provider** subscriptions - Database
7.  **All other credentials** - Database/Config files

Automatic on every update:

-  Backed up before update
-  Protected by .gitignore
-  Verified after update
-  Templates for fresh installs
-  Listed in success message

Configure once, backed up forever! 🎉

Related Documentation:

- `USER_DATA_PRESERVATION.md` - All preserved data
- `UPDATE_SCRIPT_GUIDE.md` - Update process
- `LOCAL_CONFIG_SYSTEM.md` - Config system
- `test-preservation.sh` - Test script

Last Updated: October 1, 2025