Hard Techno Agent v1.1 — Beatport JSON + Suno + ElevenLabs Pipeline

Project Documentation

Hard Techno Agent v1.1 — Beatport JSON + Suno + ElevenLabs Pipeline ## 1. Overview The *Hard Techno Agent* connects multiple AI-based components into one automation and creative pipeline: - **n8n** for orchestration and automation - **Python FastAPI** microservice (Beatport Data Parser) - **Suno v5** for generative music production - **ElevenLabs** for speech and vocal synthesis ## 2. Current Architecture The current version implements a dual-layer system: 1. **Beatport Data Layer (Python API)** — scrapes Beatport's Top 100 charts and returns clean JSON. 2. **AI Music Layer (Suno + ElevenLabs)** — generates music and vocals, coordinated by n8n. --- ## 3. Beatport Data Parser (Python FastAPI) ### Endpoints - `/health` — basic heartbeat - `/execute` — accepts JSON actions (e.g. `"beatport_top"`, `"mix_seed"`) ### Example JSON Response ```json { "ok": true, "source": "beatport", "genre": "techno", "count": 3, "items": [{ "id": "21312385", "artist": "HNTR", "title": "Shook Ones Pt. III", "url_artist": "https://www.beatport.com/artist/hntr/862283", "url_title": "https://www.beatport.com/track/shook-ones-pt-iii/21312385" }] } ``` ### Full main.py (v1.1)

```
@app.post("/execute")
async def execute(request: Request):
    payload = await request.json()
    action = payload.get("action")
    params = payload.get("params", {}) or {}
    if not action or action not in ACTIONS:
        return {"ok": False, "error": f"unknown action {action}"}
    try:
        result = ACTIONS[action](params)
        return {"ok": True, "action": action, "result": result}
    except Exception as e:
        return {"ok": False, "error": str(e)}
```

--- ## 4. n8n Workflow Integration ### Workflow Nodes 1. **Webhook** — receives POST `/pyapi-exec` 2. **HTTP Request** — calls Python API (`http://pyapi:8000/execute`) 3. **Respond to Webhook** — returns JSON response ### PowerShell Test ```powershell irm http://localhost:5678/webhook/pyapi-exec ` -Method POST ` -Body (@{action="beatport_top"; params=@{genre="techno"; limit=10}} | ConvertTo-Json) ` -ContentType 'application/json' ``` --- ## 5. Suno v5 Integration Suno v5 is used for **music generation** based on structured prompts from the Beatport data: - Input: Genre, mood, artist reference - Output: Full mastered song (MP3/WAV) - Perfect for: Hard Techno, Melodic Techno, Industrial, etc. --- ## 6. ElevenLabs Integration ElevenLabs provides **high-quality voice synthesis** for intros, drops, and branding. Typical usage within n8n: 1. Generate voice clip via ElevenLabs API 2. Merge with music output via `ffmpeg` 3. Deliver as final audio asset --- ## 7. Combined Audio Pipeline **Planned Automation v1.2:** 1. Beatport → select trending tracks 2. Generate new composition with Suno 3. Create voice intro with ElevenLabs 4. Merge and master automatically (via `ffmpeg`) 5. Upload or archive final file --- ## 8. Versioning | Version | Description | Date | |---------|-------------|------| | v1.0 | Initial Agent setup with n8n & FastAPI | 2025■10■01 | | v1.1 | Beatport JSON + Suno + ElevenLabs Pipeline | 2025■10■05 | | v1.2 | Planned: Audio■Merge & Auto■Mixdown | Upcoming | --- ## 9. Next Steps - Implement `ffmpeg` audio merge node in n8n. - Extend Python API to handle local caching of Beatport JSON. - Add parameterized Suno prompt generation. - Create ElevenLabs voice templates for DJ intros. ---