

Soundtrack API Integration Build Fixes

Issue Summary

The application was failing to build due to TypeScript compilation errors in the Soundtrack Your Brand integration. The errors were caused by mismatched method names and incorrect GraphQL query structures.

Errors Fixed

1. TypeScript Compilation Errors

Error in `/api/soundtrack/players/route.ts` :

```
Type error: Property 'listPlayers' does not exist on type 'SoundtrackYourBrandAPI'
Type error: Property 'updatePlayer' does not exist on type 'SoundtrackYourBrandAPI'
```

Root Cause: The Soundtrack API uses “Sound Zones” terminology, but the route was calling non-existent `listPlayers()` and `updatePlayer()` methods.

Fix: Updated the route to use the correct method names:

- `listPlayers()` → `listSoundZones()`
- `updatePlayer()` → `updateSoundZone()`

2. GraphQL Schema Compatibility Issues

Error during build:

```
Error: Field 'id' does not exist on type 'Viewer'
Error: Field 'accounts' does not exist on type 'Viewer'
Error: Field 'soundZones' does not exist on type 'Viewer'
```

Root Cause: The GraphQL queries were using field structures that don't exist on the Soundtrack API's `Viewer` type.

Fix: Simplified the GraphQL query structure:

Before:

```
query {
  me {
    accounts {
      soundZones {
        # ...
      }
    }
  }
}
```

After:

```

query {
  me {
    soundZones {
      # ...
    }
  }
}

```

3. Stations API Missing Required Parameter

Error:

Type error: Expected 1 arguments, but got 0

Root Cause: The `listStations()` method requires an `accountId` parameter, but the route was calling it without arguments.

Fix: Added logic to fetch account info first, then pass the account ID to `listStations()` :

```

const api = getSoundtrackAPI(config.apiKey)
const account = await api.getAccount()
const accountId = account.accounts?.[0]?.id || account.id
const stations = await api.listStations(accountId)

```

4. Static Build Errors

Error:

Soundtrack account error: Error: Field `soundZones` does **not** exist on **type** `Viewer`

Root Cause: API routes were being pre-rendered during the build process, but no valid API configuration existed at build time.

Fix: Added dynamic rendering configuration to prevent static pre-rendering:

```

// Force dynamic rendering - don't pre-render during build
export const dynamic = 'force-dynamic'

```

Files Modified

1. `src/lib/soundtrack-your-brand.ts`
 - Updated `listSoundZones()` to use simplified GraphQL query
 - Modified `getAccount()` to extract account info from sound zones
 - Added proper error handling and fallback logic
2. `src/app/api/soundtrack/players/route.ts`
 - Changed `listPlayers()` → `listSoundZones()`
 - Changed `updatePlayer()` → `updateSoundZone()`
 - Updated response mapping to match sound zone structure
3. `src/app/api/soundtrack/stations/route.ts`
 - Added account info fetch before listing stations

- Provided required `accountId` parameter to `listStations()`
- Added dynamic rendering configuration

4. `src/app/api/soundtrack/account/route.ts`

- Added dynamic rendering configuration

API Structure

The Soundtrack Your Brand API uses the following terminology:

- **Sound Zone** = A music player/zone in your physical space
- **Station** = A curated music playlist/channel
- **Account** = Your Soundtrack Your Brand account (can contain multiple sound zones)

GraphQL Query Structure

The correct structure for accessing sound zones:

```
query {
  me {
    soundZones {
      id
      name
      account {
        id
        name
      }
      currentPlayback {
        station {
          id
          name
        }
        playing
        volume
      }
    }
  }
}
```

Build Status

- ✓ **Build Successful** - All TypeScript compilation errors resolved
- ✓ **Static Generation** - 124 pages generated successfully
- ✓ **API Routes** - All soundtrack API routes properly configured for dynamic rendering

Testing Notes

The Soundtrack integration requires a valid API token to fully test. The fixes ensure:

1. **Compilation Success** - No TypeScript errors
2. **Runtime Safety** - Proper error handling when no API key is configured
3. **API Compatibility** - GraphQL queries match the actual API schema
4. **Build Process** - No attempts to pre-render dynamic API routes

Next Steps



1. Configure Soundtrack Your Brand API token in the web interface
2. Test player/sound zone listing
3. Test station listing and playback control
4. Verify “now playing” information display
5. Test bartender remote music control features

References

- [Soundtrack Your Brand API Documentation](https://api.soundtrackyourbrand.com/v2/docs) (https://api.soundtrackyourbrand.com/v2/docs)
 - [Soundtrack Example App](https://github.com/soundtrackyourbrand/api-example-app) (https://github.com/soundtrackyourbrand/api-example-app)
 - Project Documentation: `SOUNDTRACK_INTEGRATION_GUIDE.md`
 - Setup Instructions: `SOUNDTRACK_SETUP.md`
-

Commit: c5fff7d

Date: October 1, 2025

Status:  Build Passing |  Ready for Testing