

Testing Music Mobile Apps – Part 1

Caching, Content & Playback

What's special about music apps...?

- Playback functionality
- Queuing/downloading strategies
 - Batch favorites
- Online/Offline
- Caching Strategy – first track, second track
- Network Connections – different connections including low-bandwidth
- Content
- Images – album art images with metadata
- Metadata – is this accurate?
- Audio quality – bit rates, skipping
- Algorithms – recommendation engines, third party apis?
- UI/UX – how people consume music?
- Authorization – different user access levels
- Discovering screens – loading a lot of content vertically scrolling, refreshing
- Duplications, Inconsistent data
- Flickering – loading time of pages of content
- Content ingestion – different encodings
- Music Library – adding tracks here
- Sharing tracks to FB, other users
- Playlist Creations

Music Tech functionality

- **Playback functionality**
 - Queuing
 - Online/Offline Mode
 - Playback controls (forward, previous, shuffle, repeat)
 - Low-bandwidth
- **Caching strategy**
 - Streaming vs. Progressive download
 - How many tracks are cached?
 - Batching strategy
- **Content Ingestion**
 - Duplications
 - Missing metadata
 - Understanding ingestion formats/encodings

Testing Approach

- Is more complex than just a visual inspection on the front-end
- Understanding root-cause of issues requires a holistic view of entire system
- And how data moves through various integration points, check data formats in portals or databases
- Inquire about original point of entry into system to final point of client facing

Testing Results

Hope this gives some insights into
approaches for testing music
functionality!

Understanding the source provides
repeatable approaches and quicker
feedback loops to find bugs quickly!

Happy Testing!