[VIDEO PLAYER]: Content Schedule

Application Service Functional Specification

# Team Information

Application Services Team

|  |  |  |
| --- | --- | --- |
| Role | Alias | Signed Off? |
| Program Manager | Jesse Squire |  |
| Lead Developer | [PERSON] |  |
| Lead Tester | [PERSON] |  |
| Team Ownership | [PERSON] |  |

Video Player Application Team

|  |  |  |
| --- | --- | --- |
| Role | Alias | Signed Off? |
| Program Manager | [PERSON] |  |
| Lead Developer | [PERSON] |  |
| Lead Tester | [PERSON] |  |
| Lead Designer | [PERSON] |  |

# References

|  |  |  |
| --- | --- | --- |
| Reference | Owner | Link |
| [REDACTED] |  |  |
| [REDACTED] |  |  |

# Executive Summary

One of the design tenants of the [VIDEO PLAYER] application is that a user should never be presented with “dead air” – meaning that a content item should always be playing. As part of the descision process for which content to play for a given user, an ordered list of content personalized to the user will be produced, referred to as the content schedule.

The content schedule will be comprised of content identified by programmers as featured, the current user’s personal selection list, and recommendations for content that the user may be interested in. The content schedule will not directly store content; rather it will track references to content sourced from external systems such as a content management system (vCMS) and external streaming partners. The source systems are considered the system of record for their content, including all of its associated metadata. The content schedule is authoritative only as it pertains to the sequence of content that will be considered for presentation to a given user at the current point in time.

# Definitions & Acronyms

|  |  |
| --- | --- |
| Term | Definition |
| Content Schedule | An ordered list of references to content specific to a given user used by the [VIDEO PLAYER] application as part of its decision making process for determining what content should be played for a user that hasn’t explicitly chosen an item. |

# Scenarios

**Startup Experience**

After purchasing [VIDEO PLAYER], River launches the application for the first time intending to check out all of the cool things that her brother has told her about. When the application starts, video content begins playing automatically, offering a TV-like experience. In order to personalize the experience for River, a content schedule is assembled specifically for her which [VIDEO PLAYER] uses as a driver for selection of the content that River will see. Because she has not yet watched any videos in [VIDEO PLAYER], River is shown the first item of featured content. River is pleasantly surprised when a video starts playing without her needing to take the time to select one; she begins watching and realizes after a few minutes that she’s hooked on “Honey, Where are my Pants?”

**[VIDEO PLAYER] Reports What a User has Watched**

Simon opens [VIDEO PLAYER] and begins watching an episode of “OMG, Pokemon!”. As he nears the end of the video, he passes the point in the timeline when [VIDEO PLAYER] considers the video to have been watched. In order to help the content schedule be better customized for him, [VIDEO PLAYER] notifies the content schedule service that the video has been watched by Simon.

**Users are Presented with Interesting Content to Watch**

Mal is a busy individual; he has a ship to captain and a crew to look after. While he enjoys spending his downtime watching videos in [VIDEO PLAYER], his schedule prevents him from doing so often. Since his free time is limited, he appreciates that [VIDEO PLAYER] usually presents him content that he hasn’t already seen. To support this goal, when the content schedule is prepared for Mal, content that he has already watched is filtered out whenever possible leaving him with only items that he hasn’t seen.

As Mal continues to watch videos in [VIDEO PLAYER], his content schedule will periodically be refreshed, ensuring that content he has watched is pruned and that new, unwatched content can be presented as it becomes available. In the event that no unwatched content is available for Mal, [VIDEO PLAYER] will present him a random set of content that he has already watched.

**No New Content is Available**

There is a man that they call Jayne who is obsessed with shooters, especially Doom. Once he found out about [VIDEO PLAYER], Jayne began binge watching content at an impressive pace. Within the first month, he had watched every bit of [VIDEO PLAYER] content available to him. When attempting to assemble a content schedule for Jayne, no unwatched content can be found. Jayne’s content schedule is constructed with a set of random content that Jayne has already watched.

As Jayne continues to re-watch videos, [VIDEO PLAYER] will periodically attempt to refresh his content schedule. If new [VIDEO PLAYER] content is published during Jayne’s session, it will appear as part of his content schedule, allowing [VIDEO PLAYER] to once again present the man they call Jayne with videos that he has not yet watched.

# Out of Scope

* Storage, creation, modification, deletion, or preview of content itself; these are the purview of the source system.
* Storage, creation, localization, modification, or deletion of metadata associated with content; these are the purview of the source system.
* Allowing previewing or editing of any of the individual content lists that comprise the content schedule; the editing and preview experience is the purview of the source system.
* Determination or enforcement of a user’s right to view the referenced content.
* Determination of the validity of content in the source system; if a content item appears in one of the content schedule source lists, it is considered valid for selection.
* Consideration of a user’s region, ratings restrictions, parental controls or other factors in the selection of content; if a content item appears in one of the content schedule source lists, it is considered valid for selection.
* Detailing of recommendations; this functionality will be covered in a dedicated specification.
* Notifying consumers of changes to the underlying featured content items in the source system or the watched state being tracked for a given user.
* Retention of content schedule history; only the current state is known.
* Selection between multiple sets of featured content in the content management system; it is assumed that any items identified as active featured content are current and valid for use.
* Performing source system queries for additional content during content schedule preparation logic to inspect related items; it is expected that all relevant information for content schedule preparation are present on the content instances contained in the root featured content and Video On Demand catalog queries.
* Integration with external streaming partners services; the content schedule will allow external stream references to be tracked but will not interface with the source APIs.

# Requirements

|  |  |  |
| --- | --- | --- |
| Priority | Requirement | Release |
| P1 | Allow retrieval of an ordered list of content references, personalized to a given user | Vertical Slice |
| P1 | Enable references to content identified as active featured in the video content management system (vCMS) content to be used in the content schedule list | Vertical Slice |
| P1 | Enable references to content from a given user’s selection list to be used in the content schedule list | Vertical Slice |
| P1 | Personalize the content schedule for a given user by combining references from the featured content list, the items in the user’s selection list, and Video On Demand content in the [VIDEO PLAYER] catalog | Vertical Slice |
| P1 | Selection logic for the content schedule is consistent across all client platforms | Vertical Slice |
| P2 | The content references returned should include an identifier of the target system (vCMS, external streaming service, etc.) as well as an identifier for the list from which it was selected into the content schedule (Featured Content, User Selections, etc.) | Vertical Slice |
| P1 | Apply business logic to the content schedule personalization to select and order content items from the different content sources that comprise the list | Beta |
| P1 | Ensure that there are no duplicate content references in the content schedule for a given user | Beta |
| P1 | Ensure the personalized content schedule does not contain references to content that was previously watched by the user unless specifically allowed by use cases in the selection business logic | Beta |
| P2 | Enable an administrative user to be able to impersonate any other user for all service endpoints to support debugging and sustainment needs | Beta |
| P3 | Enable references to recommended content for a given user to be included in the content schedule | RTM |
| P3 | Return a subset of associated metadata from the source system with the content reference, for content sourced from the vCMS; this metadata should be localized to the user’s preferred culture.  This item is only necessary if the client’s intended usage pattern changes | RTM |

# Golden Path

* An API endpoint is available which will return a personalized content schedule for a given user, following the defined algorithm.
* An API endpoint is available which will allow a caller to record that a given user has watched a given content item.
* Target response time is met for the anticipated scale required by 500 users.

# Givers

|  |  |
| --- | --- |
| Team | Reason for Dependency |
| [TEAM NAME] | The video content management system (vCMS) is the primary source of content and its associated metadata referenced by the content schedule. |
| [TEAM NAME] | The [VIDEO PLAYER] team is responsible for defining the vCMS queries used in assembly of the content schedule, described in the functional design. |

# Takers

|  |  |
| --- | --- |
| Team | Reason for Dependency |
| [TEAM NAME] | The [VIDEO PLAYER] client applications are the intended primary consumers of the service. |

# Functional Design

## Overview

The service will attempt to assemble an ordered list of video content that has been personalized to a given user, dubbed a content schedule. The content schedule will be used by [VIDEO PLAYER] as a potential set of content to present the user; it is not the authority on what will be shown. The [VIDEO PLAYER] application manages content outside of the content schedule, and will use its own internal algorithms for determining the most appropriate content to present to the user. For example, if a user chooses to watch the first episode of a series from the channel guide, [VIDEO PLAYER] will prioritize showing the other episodes from the season that that series over content from the content schedule.

The content schedule content will be selected from multiple sources in a best effort to produce a list containing the requested number of unwatched items. In the event that enough unwatched items are not available from the sources of content, the content schedule will begin using a random set of watched content from the available Video On Demand items in the [VIDEO PLAYER] catalog, excluding those items that are considered stale.

Content schedule items are not content themselves, rather references to external content. In order to uniquely identify content and allow a client to locate it, the content reference is a composite value that indicates the content’s source system and a unique identifier for the content within the source system. Neither the content nor the metadata related to the content will be stored or owned by the content schedule; these data items are considered the purview of the content’s source and should be allowed to evolve without the need to keep items in sync between systems.

The service assumes no knowledge of the content within its source system and therefore can take no steps to validate that a given identifier uniquely identifies a content item nor that the identified content is allowable for use by a particular user or within the client application. The service also assumes no responsibility for verifying that the identified content is appropriate for the user according to his/her region, parental controls, or other preferences nor that the user is entitled to view the content.

The service will provide a set of RESTful APIs that allow callers to request a content schedule and to track the content that has been watched. A JWT bearer token is required for standard users, and will be used to enforce the restriction that a user may only interact with his/her own content schedule. For administrative scenarios, a client certificate will be accepted and allow the caller to request a content schedule or track the watched state of content for any user.

## Content Schedule Video Sources

Content schedule items will be sourced from the items in the user’s selection list, the set of [VIDEO PLAYER] featured content in the vCMS, and the set of Video On Demand content in the vCMS. The service does not have any knowledge of the organization or importance or relevence of content within the vCMS. From the service’s perspective, content to be retrieved from the vCMS is understood only as a set of vCMS queries.

It is expected that the featured content list and set of Video On Demand items can each be retrieved using a single vCMS query, expressed as a url. Combined with the user selection list items, the query results are considered by the service to be the authoritative set for inclusion in the content schedule, subject to the rules of the assembly algorithm.

The set of items for content schedule inclusion should ideally be as recent as possible, at minimum no more than 2 minutes old. The exception to this is when the source system specifies a cache policy. In that case, the source should be treated as the authority. For example, if featured content list is retrieved from the vCMS, and the vCMS asserts that the content may be cached for 5 minutes, it is permissible to cache featured content for the 5 minutes.

## Determination of Content Freshness

Each content item used for preparation of the content schedule will either be an instance of the Video Media content type from the vCMS or contain a direct reference to a Video Media instance. Each Video Media instance will have an attribute that represents the UTC date/time that it was published and an attribute that represents the number of days that the item is considered fresh after it has been published. For most content, the “freshness days” attribute will not be populated, indicating that the content is always considered fresh.

When determining if an item of content is fresh, the following rules are applied:

* If the “freshness days” attribute is empty or null, the content is considered fresh
* If the “freshness days” attribute is zero or a negative value, the content is considered stale
* If the publish date attribute is empty or null, the content is considered stale *(Note: this should never happen in practice)*
* If the result of adding the “freshness days” to published date is greater than or equal to Now, the content is considered fresh. The time component of the published date and now is relevant and taken into account for this comparison.

## Content Schedule Assembly

Given a requested content size and a user, the content schedule will be assembled by the following steps:

1. From the set of featured content, remove items that the user has watched while maintaining the order of the list as it was returned from the vCMS. Select the number of unwatched items needed in order. If there are not enough unwatched featured content items available, then select as many as possible.
2. If content schedule meets the requested content size, then it is now considered complete.
3. If more items are needed, then the user’s selection list will be considered. From the user’s selection list, remove the items that the user has watched and those that appear in the content schedule while maintaining the order of the selection list. Select the number of unwatched items needed, appending them to the existing featured content items selected for the content schedule. If there are not enough unwatched selection list items available, then select as many as possible.
4. If content schedule meets the requested content size, then it is now considered complete.
5. If more items are needed, then the set of Video On Demand content will be considered. From the Video On Demand content, remove the items that the user has watched, those that are considered stale, and those that already appear in the set of content selected for the content schedule. Select the number of unwatched items randomly from the available set, appending them to the existing featured content and user selection list items selected for the content schedule. If there are not enough unwatched list items, then select as many as possible.
6. If content schedule meets the requested content size, then it is now considered complete.
7. If more items are needed, then items the user has watched in set of Video On Demand content will be considered. From the Video On Demand content, remove the items that are considered stale and those that already appear in the set of content selected for the content schedule. Select the number of unwatched items randomly from the available set, appending them to the existing featured content, user selection list, and unwatched items selected for the content schedule. If there are not enough items in the available set, then select as many as possible.
8. The content schedule is now considered complete. It may or may not satisfy the requested content size. In the event that there are not enough fresh Video On Demand items available in the content catalog, it is possible that the content schedule may be empty or contain fewer than the requested number of items.

The flow can be pictured as follows:

# Service APIs

## Get Content Schedule

|  |  |  |
| --- | --- | --- |
| **Method** | **URI** | **What is it?** |
| GET | /{title}/users/{user}/schedule | Allows a caller to retrieve the personalized content schedule for a user |

If a bearer token is provided and client certificate auth was not explicitly requested, the service will verify that the user identified by the bearer token matches the {user} specified in the URL. If the user components do not represent the same user, the request will be refused as unauthorized.

If a bearer token is not provided, then a client certificate will be required. If only a client certificate was provided or client certificate auth was requested, it is assumed that the call is being made by an administrator on behalf of the {user} specified in the URL. In this case, no further identity checks will be performed but the certificate will be validated as follows:

* The certificate must be valid (not expired, chained to a trusted root, etc.)
* The certificate must be authorized for this API

If both a bearer token and client certificate are provided and the caller has not specific which authorization/authorization method should be used then the bearer token will take precedence.

### When is a consumer expected to call this API

* When the [VIDEO PLAYER] application launches, it will retrieve the content schedule for consideration when presenting content to the user.
* As the user finishes watching video content, the [VIDEO PLAYER] application will invoke this API to refresh the data.

### How should a consumer be calling this API

* Content schedule data should not be refreshed more than once every five minutes for a particular user on a particular device.

### Performance Targets

|  |  |  |
| --- | --- | --- |
| Requirement | Target | Notes |
| Response time @ peak load (ms) | 250 ms |  |
| Peak load (RPS) | 16 RPS | Please see [Performance Calculations and Approach](#_Performance_Calculations_and) for assumptions and details. |
| API complete load (RPS) | 1 RPS | Please see [Performance Calculations and Approach](#_Performance_Calculations_and) for assumptions and details. |

**User impact if not at target response time:**

Users may be presented with content that isn’t ideal, such as a limited set of items or items that have already been watched.

### Reliability Targets

|  |  |  |
| --- | --- | --- |
| Requirement | Target | Notes |
| Availability | 99.9 % | The default service availability target is 99.9%, < 10.1 minutes down per week.  99.9% = 10.1 minutes / week, total of 8.76 hours of downtime / year.  99.95% = 5.04 minutes / week, total of 4.38 hours of downtime / year.  99.99% = 1.01 minutes / week, total of 52.56 minutes of downtime / year. |

**User impact if service is unavailable:**

Users may be presented with content that isn’t ideal, such as a limited set of items or items that have already been watched.

## Set Content Schedule Item State

|  |  |  |
| --- | --- | --- |
| **Method** | **URI** | **What is it?** |
| PATCH | /{title}/users/{user}/schedule/{sourceSystem}/{sourceId} | Allows a caller to update the state relative to a user's content schedule interactions |

If a bearer token is provided and client certificate auth was not explicitly requested, the service will verify that the user identified by the bearer token matches the {user} specified in the URL. If the user components do not represent the same user, the request will be refused as unauthorized.

If a bearer token is not provided, then a client certificate will be required. If only a client certificate was provided or client certificate auth was requested, it is assumed that the call is being made by an administrator on behalf of the {user} specified in the URL. In this case, no further identity checks will be performed but the certificate will be validated as follows:

* The certificate must be valid (not expired, chained to a trusted root, etc.)
* The certificate must be authorized for this API

If both a bearer token and client certificate are provided and the caller has not specific which authorization/authorization method should be used then the bearer token will take precedence.

### When is a consumer expected to call this API

* As the user finishes watching video content, the [VIDEO PLAYER] application will invoke this API to record that the video was watched.

### How should a consumer be calling this API

* It is safe to call this API when unsure if state is currently being tracked for this user or content item. If a tracking item doesn’t exist, it will be implicitly created. If it does exist, it will be updated.
* This API is idempotent; if an attempt is made to update a tracking item in with the same data, no changes will be made to the item or selection list.

### Performance Targets

|  |  |  |
| --- | --- | --- |
| Requirement | Target | Notes |
| Response time @ peak load (ms) | 250 ms |  |
| Peak load (RPS) | 12 RPS | Please see [Performance Calculations and Approach](#_Performance_Calculations_and) for assumptions and details. |
| API complete load (RPS) | 1 RPS | Please see [Performance Calculations and Approach](#_Performance_Calculations_and) for assumptions and details. |

**User impact if not at target response time:**

Users may be presented with an experience where the [VIDEO PLAYER] client responds slower than expected.

### Reliability Targets

|  |  |  |
| --- | --- | --- |
| Requirement | Target | Notes |
| Availability | 99.9 % | The default service availability target is 99.9%, < 10.1 minutes down per week.  99.9% = 10.1 minutes / week, total of 8.76 hours of downtime / year.  99.95% = 5.04 minutes / week, total of 4.38 hours of downtime / year.  99.99% = 1.01 minutes / week, total of 52.56 minutes of downtime / year. |

**User impact if service is unavailable:**

Users may be presented with content that isn’t ideal, such as a limited set of items or items that have already been watched.

# Testability

* Any dependencies on the execution environment, other services, data persistence, and the video content management system should allow tests to override or inject those dependencies to enable mocking.
* A test hook which allows for retrieval of a tracking item for a given user/content pairing should exist, as there is no API for this operation.
* A test hook which allows a tracking item for a given user/content pairing to be deleted should exist, as there is no API for this operation.

# Security

The following security restrictions will be imposed:

* A user may view only his/her own content schedule.
* A user may update the tracking state only for him/herself.
* Administrators may retrieve a content schedule and manage the tracking state on behalf of any given user.

Should these restrictions be violated, the impact to a user would be that their selection list may not be in the expected state; it may contain extra items or have items missing.

# Privacy

Content state tracking data is matched to a given user by their unique identifier in the [VIDEO PLAYER] ecosystem, which will be persisted. No other personal data for a given user is tracked, though unauthorized access to a user’s content tracking state may expose information about that user’s content interests or the timing of the user’s activities. Because only an opaque content reference is tracked, in order to be able to make inductions about a user’s interest, each content reference would also need to be matched to the actual content item in its source system.

# Global Readiness

Not applicable as only opaque references to content are referenced. It is considered the purview of the source system to provide proper localization support for the content itself.

# Telemetry Requirements

|  |  |  |
| --- | --- | --- |
| Pri | Question to answer | What will you do differently based on the result? |
| 3 | What is the average response time for vCMS queries? | Consider caching vCMS data locally for a period of time. |
| 3 | What is the average number of items assembled for a content schedule from each content source? | Consider revising the algorithm for assembling the content schedule. |

# Risks & Open Issues

## Risks

|  |  |  |
| --- | --- | --- |
| Description | Likelihood | Impact |
|  |  |  |

## Open Issues

|  |  |  |  |
| --- | --- | --- | --- |
| Pri | Owner | Description | Resolution |
|  |  |  |  |

# Appendix

## Performance Calculations and Approach

The projection for maximum concurrent users of 13,000 was taken from the launch concurrency projections in the [OTHER APPLICATION] Services Scale and Capacity Planning workbook.

The following assumptions were made:

* Peak concurrent users is the projected number for [VIDEO PLAYER].
* API Complete calculations assume 500 concurrent users.
* All users sign in within a one hour period.
* At any given time 80% of the users are watching a video.
* Each video that a user watches is 15 minutes long.
* Each video has a point in the timeline to denote that the user is considered to have fully watched the video.

Calculations for API Complete:

* Get Content Schedule = 0.582 RPS
  + StartUp: ((500 / 60 min / 60 sec) \* 1) = 0.138 RPS
  + Watching Videos: (((500 \* 0.80) / 60 min / 60 sec) \* (60 min / 15 min)) = 0.444 RPS
* Set Content Schedule Item State = (((500 \* 0.80) / 60 min / 60 sec) \* (60 min / 15 min)) = 0.444 RPS

Calculations for Peak:

* Get Content Schedule = 15.161 RPS
  + StartUp: ((13,000 / 60 min / 60 sec) \* 1) = 3.61 RPS
  + Watching Videos: (((13,000 \* 0.80) / 60 min / 60 sec) \* (60 min / 15 min)) = 11.55 RPS
* Set Content Schedule Item State = (((13,000 \* 0.80) / 60 min / 60 sec) \* (60 min / 15 min)) = 11.55 RPS