**Metadata Bank – UAT3 - Release Notes:**

1. **UAT3 Environment Information:**

**baseurl**: <https://metadata-qa.services.pbs.org/v1>

**apikey:** LNFJ1pyhblxfVETh8xGf

**x-api-key:** 6rlxabgCi1cpdG0joyyh5lnRYxoukxp7vG7fGu88

| Artifacts Name | Artifacts |
| --- | --- |
| Postman Collection |  |
| Postman – Environment information |  |
| JSON Schema |  |
| Github Location | https://github.com/pbs-metadata/Enterprise-Metadata-Model/ metadata-bank |

1. **CRs in UAT2 Endpoints:**

* The UID is removed from ContentIDList of Series, Season, Episode, Manifestation and Compilation.
* Restricted the duplicate ContentIds of all entities.
* The "EntityType", "CreateDataTime", "UpdateDateTime" and "Published Status" fields are not allowed in POST & PUT payload of all entities.
* Implemented 2 level[AWS API Gateway & CA Live Creator] authentication & authorization for all endpoints.
* The **FileList** is added to Manifestation GET payload.
* New search features are added for existing endpoints.
  1. **Episode** - /epsiode/search?title={title}&**contentId={contentId}**.
  2. **Manifestation** - /manifestation/search?title={title}&contentId={contentId}&usewindowtermstartdate={usewindowtermstart}&usewindowtermenddate={usewindowtermenddate}
  3. **Compilation** - /compilation/search?title={title}&**contentId={contentId}&entrycontentuid={entrycontentuid}**
  4. **Policy -** /policy/search?title={title}&**entryuid={entryuid}**

1. **UAT2 Bug Fixes:**

| Jira Defect# | Description |
| --- | --- |
| MBank-331 | Language fields accept any String value |
| MBank-362 | Various required fields will pass validation as an empty String |
| MBank-503 | POST /compilation - Can add non-existing Compilations to the EntryList |
| MBank-498 | GET /policy/{uid} returns a 200 response code when UID does not exist |
| MBank-503 | POST /compilation - Can add non-existing Compilations to the EntryList |
| MBank-504 | POST /policy/{PolicyUID}/rule - Can set the RuleUID in the payload |
| MBank-511 | Problem updating EpisodeList in a Season |

1. **New endpoints in UAT3:**
   1. **Manifestation**:
      * **POST** - /manifestation/{uid}/usewindowterm – Add new UseWindowTerm in existing Manifestation.
   2. **File**:
      * **GET** - /file/{uid} – Get an existing file.
      * **POST** - /file – Add new file.
      * **PUT** - /file/{uid} – Updates an existing file.
      * **DELETE** - /file/{uid} – Deletes an existing file.
      * **POST** - /file/{uid}/alternateassetidregistration – Register a new alternate asset ID.
   3. **Events**:
      * **GET** - /event/{event-uid} - Get an existing event.
      * **POST** - /event – Add a new Event
      * **PUT** - /event/{event-uid} – Updates an existing Event.
      * **DELETE** - /event/{event-uid} – Deletes an existing Event.
      * **POST** - /event/{event-uid}/linear – Add a new Linear to existing Event.
      * **PUT** - /event/{event-uid}/linear/{alternate-event-id} – Updates an existing Linear information except playrun & for an Event.
      * **PUT** - /event/{event-uid}/linear/{alternate-event-id}/playrun – Updates only a playrun attribute from existing Linear of an Event.
      * **PUT** - /event/{event-uid}/linear/altereventidregistration/{previous-alternate-event-id} - Updates only a **AlternateEventID** attribute for existing Linear of an Event.

* + - **POST** - /event/{event-uid}/vod - Add a new VOD to existing Event.
    - **PUT** - /event/{event-uid}/vod/{alternate-event-id} - Updates an existing VOD information for an Event.
    - **PUT** - /event/{event-uid}/vod/altereventidregistration/{previous-alternate-event-id} - - Updates only a **AlternateEventID** attribute for existing VOD of an Event.
  1. **Show Schedule:**
     + **GET** - /showschedule?eventdate={eventdate}&episodeId={episodeUID}&manifestationId={manifestationUID}
     + **GET** - /showschedule/vod?startdatetime={startdatetime}&enddatetime={enddatetime}&ProviderSource={ProviderSource}
     + **GET** - /showschedule/linear?providerSource={ProviderSource}&startdatetime={startdatetime}
  2. **Catalog:**
     + **GET** - /catalog/search?episode-title={episode-title}&season-title={season-title}&series-title={series-title}&contentId={contentId} – Search the respective title with ‘OR’ condition and returns array of Series Object. If contentId is present along with these title then it will consider only contentId and search it in Series, Season and Episode’s ContentId.
     + **GET** - /catalog/series/{series-uid} – Search exact SeriesUID and return Series Object.
     + **GET** - /catalog/season/{season-uid} – Search Exact SeasonUID and returns appropriate Series Object
     + **GET** - /catalog/episode/{episode-uid} – Search exact EpisodeUID and returns appropriate Series Object.