

PatchWall Integration Guidelines

Communication	3
About	4
Availability	4
PatchWall UI Architecture	5
Overview	7
Player Access	8
PatchWall Customization	14
Dolby Atmos Support Identification	19
Dolby Digital Plus Identification	19
Checklist for Content Partner for Google TV Adoption	20
Change Log	21

Communication

CONTACTS

Person	Department	Role	E-mail
PatchWall India	GTM - Product	Group Email	ind-patchwall@xiaomi.com
Anushree	GTM - Product	Product Technical	anushreetaori@xiaomi.com
Mono	GTM - Product	Product	<u>manojitballav@xiaomi.com</u>

NOTE:

Partners are requested to add the group email address (ind-patchwall@xiaomi.com) across all communications
Partners are requested to get in touch with regional contact person for any queries or issues

About

PatchWall is a proprietary content discovery service of Xiaomi which is available across Smart Televisions and Streaming Devices sold under the brand of Xiaomi and Redmi. Work on PatchWall started around 2017 and we launched it with our first devices in early 2018. As of writing this document PatchWall is available across 8M+ households in the India Region. PatchWall has over 30 OTT partners in India powering the platform's content discovery service across multiple content types, languages and genres.

Availability

Operating System

PatchWall is available across AOSP 7, androidtv 8.1, 9, 10, 11, Google TV on androidtv 11

Smart TV

Mi Smart TV

- Mi LED TV 4 | 4 Pro Series
- Mi LED TV 4A | 4A Pro Series
- Mi LED TV 4X Series
- Mi TV Q Series

Redmi Smart TV

- Redmi Smart TV Series
- Redmi TV X Series

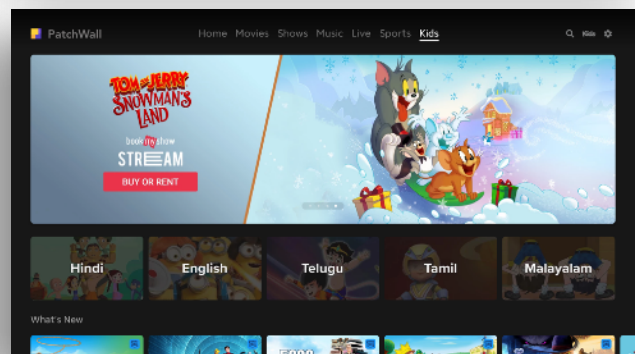
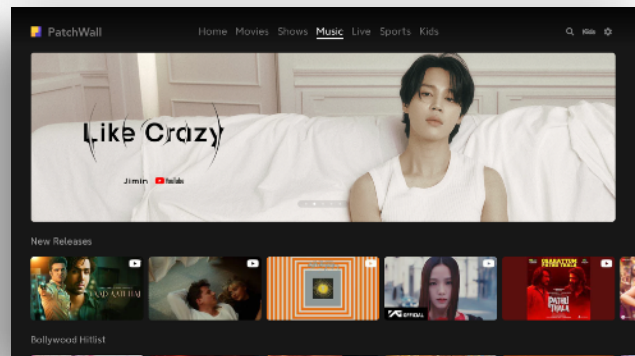
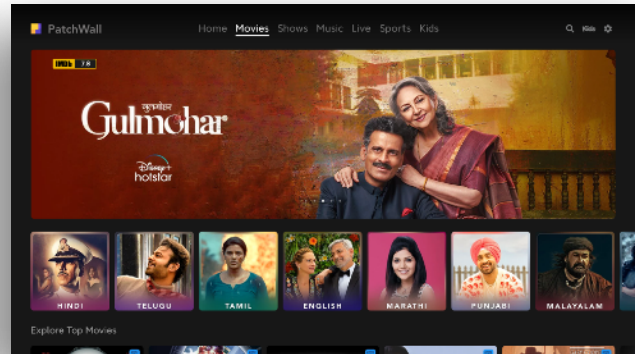
Xiaomi Smart TV

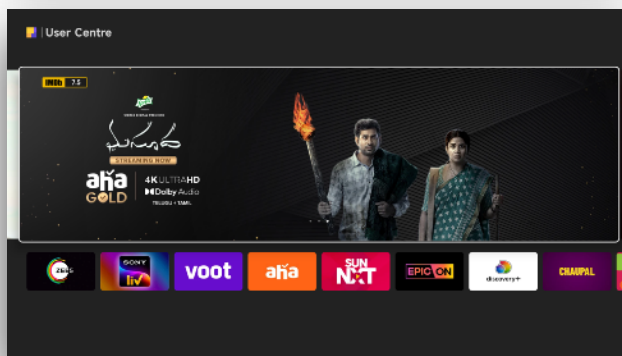
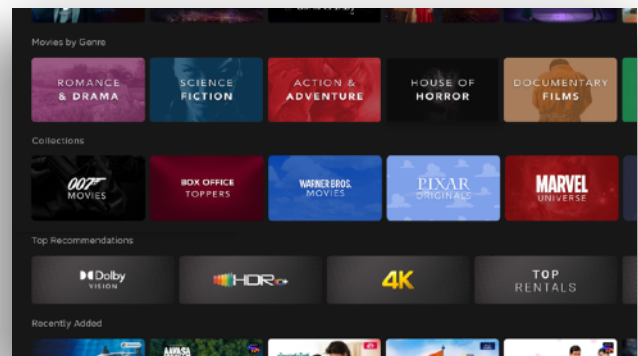
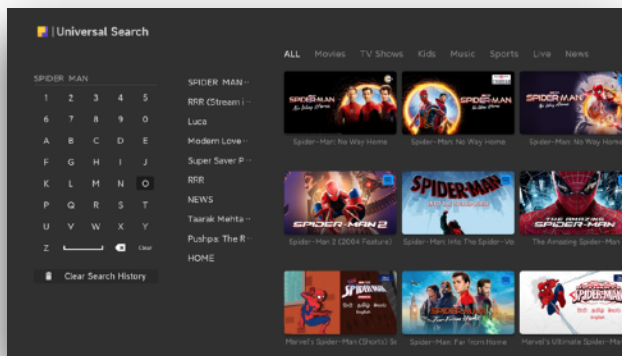
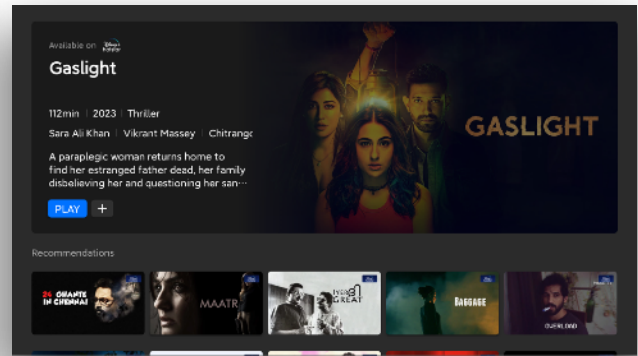
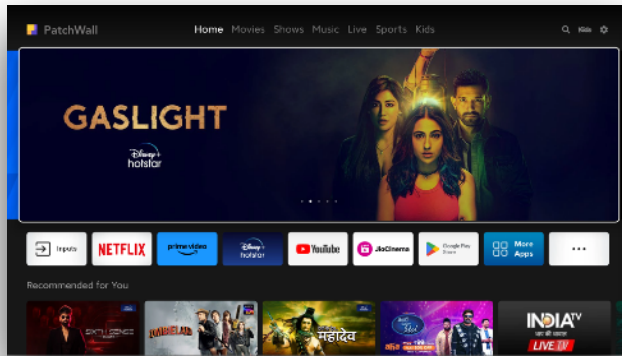
- Xiaomi Smart TV 5X Series
- Xiaomi Smart TV 5A | 5A Pro Series
- Xiaomi Smart TV X Series | X Pro Series
- Xiaomi OLED Vision

Streaming Devices

- Xiaomi TV Stick 4K
- Mi Box 4K - *In consideration*
- Mi TV Stick - *In consideration*

PatchWall UI Architecture





Overview

PatchWall Integration Guidelines is a set of guides and process designed to allow content partners to integrate their contents into PatchWall in a effective and efficient manner. It will be the content partner's responsibility to ensure that the integration fully complies with PatchWall Integration Guidelines including it subsequent changes. Any major changes to this document, will be communicated to the Partner well in advance.

Integration method: Player Access

This is a standard integration method where the content is played within the content partner app. In this integration method, the content partner shares the content metadata through an API using which PatchWall showcases contents across its tabs for users to discover content. This integration method is widely suitable for content partners having a large library of content which are behind a paywall.

Preferred for AVOD, SVOD, TVOD & PVOD

Integration method: F.A.S.T Channels

Please request your PatchWall Business Partner to share Integration documents for PatchWall+ Integration

Preferred for Live, FAST channels

Player Access

Requirements

1. Google TV Play Store | androidtv Play Store Listed Package
2. Content API
3. PatchWall customization on APK
4. Subscription Attribution

Play Store Listing

After the recent changes in Google Policies for OEM's and App Developers, every content partner of PatchWall is required to have their packages published in androidtv | Google TV Play Store and enabled for India Region.

Content API

The content api is expected to contain all the meta data related to each content , this metadata will be ingested in the PatchWall Content Management System where the metadata will be used by our recommendation engine and human operators to promote a content.

PatchWall Content Partners are required to structure the Content API as per the guidelines discussed in this topic. The content api is expected to follow the reference structure shared later in the document.

The content API should be organized by resource type and constitute of 2 endpoints.

1. Title ID
2. Title Details

TITLE ID

The Title ID list resource/endpoint is expected to contain a list of title id's which should contain new content id and changed content id in a given time interval. Partners can share title id's based on content update time, however in situations where the time parameter is absent in the request, it is expected from the API to return all available title id's.

API URL

`http(s)://${domain}/content/titleidlist?startTime=1500173517&endTime=1500360397&pageNo=1&pageSize=50`

METHOD

HTTPS | GET

Request Parameters

Key	Description	Type	Requirement
domain	Domain of the API	String	REQUIRED
startTime	Start time of the availability of the title or title data change	Unix timestamp seconds	REQUIRED
endTime	End time of the availability of the title or title data change	Unix timestamp seconds	REQUIRED
pageNo	Page number	Integer (default value 1)	REQUIRED
pageSize	Number of items listed in a page	Integer (default value 500)	REQUIRED

Response Parameters

Key	Description	Type	Requirement
message	Response message	String	REQUIRED
code	0: success 1: invalid parameters 2: empty data 3: forbidden 4: server exception	Integer	REQUIRED
timestamp	Server response time	Unix timestamp second	REQUIRED
id	Title id	Integer	REQUIRED
isOnline	Is title in use	Boolean	REQUIRED
pageCount	Total number of pages for the query response	Integer	REQUIRED
hasMore	States availability of more data for the query	Boolean	REQUIRED
pageSize	States number of items in the current page	Integer	REQUIRED
pageNo	States current page number	Integer	REQUIRED

TITLE DETAILS

Title Detail interface should contain details of each title id from the Title ID interface.

We request 2 types of response parameters:

- Movies & Individual Videos
- TV Shows with Episode Data

API URL

`http(s)://${domain}/content/title/${titleId}`

METHOD

HTTPS GET

RESPONSE PARAMETERS FOR MOVIES & INDIVIDUAL VIDEOS

Field	Description	Type	Requirement
message	Response message	String	REQUIRED
code	0:success 1:invalid parameters 2:empty data 3:forbidden 4:server exception	integer	REQUIRED
timestamp	Server response time	Unix timestamp second	REQUIRED
data field			
id	unique identifier of the title	integer	REQUIRED
title	general title of the video	string	REQUIRED
originalTitle	original title of the video	string	REQUIRED
titleType	Type of the tile. movie, shows, music, short, trailers, news, sports or N/A if not applicable	string	REQUIRED
description	Description of the title	string	REQUIRED
shortDescription	Shortened description of the title	string	RECOMMENDED
hrPosterURL	url of the horizontal orientation of poster. (640*360)	url	REQUIRED
vrPosterURL	url of the vertical orientation of poster (400*600)	url	REQUIRED
releaseDate	the actual release date of the tile	yyyy-mm-dd	REQUIRED
startYear	Applicable for Shows, the year of first air time	yyyy	REQUIRED

endYear	Applicable for Shows, the year of last air time	yyyy	REQUIRED
runtime	total duration of the title in seconds	integer	REQUIRED
resolution	Max. resolution of the content 0 - No info 1 - HD 2 - FHD 3 - 4K 4 - 8K	integer	REQUIRED
pictureQuality	Max. picture quality of the content 0 - No info 1 - HDR 2 - HDR 10+ 3 - Dolby Vision 4 - IMAX Enhanced	integer	REQUIRED
soundEffect	0 = Normal 1 = Dolby 2 = Dolby Atmos 3 = DTS	integer	REQUIRED
genres	include upto 5 genre type	string array	REQUIRED
keywords	keyword associated with the title	string array	RECOMMENDED
languages	content language	string array	REQUIRED
regions	contains the region in which the title was produced	string array	REQUIRED
channels	name of the channel to which the title belongs	string array	RECOMMENDED
directors	contains name of the director(s)	string array	REQUIRED
writers	Contains name of the producer(s)/writer(s)	string array	RECOMMENDED
principalCast	contains name of the lead actors	string array	REQUIRED
averageRating	average individual rating on a scale of 0 to 10	float	RECOMMENDED
playingType	Type of video playback: 0 for VOD 1 for LIVE 2 for Time shifted	integer	REQUIRED
payType	Type of payment: 0 for free 1 for free with ADS 2 for Pay-per-view 3 for subscription	integer	REQUIRED
numEpisodes	Applicable for Shows, share number of episode available for the target show	integer	REQUIRED
isFinal	Indicates if a TV Series ends. True by default	boolean	REQUIRED
age	minimum viewing age required	integer	REQUIRED
deeplink	intent to the particular content.	string	REQUIRED

RESPONSE PARAMETER FOR SHOWS - EPISODES

Field	Description	Type	Requirement
message	Response message	String	REQUIRED
code	0:success 1:invalid parameters 2:empty data 3:forbidden 4:server exception	integer	REQUIRED
timestamp	Server response time	Unix timestamp second	REQUIRED
data field			
id	unique identifier of the title	integer	REQUIRED
title	general title of the video	string	REQUIRED
originalTitle	original title of the video	string	REQUIRED
titleType	Type of the tile. movie, shows, music, short, trailers, music, news, sports or N/A if not applicable	string	REQUIRED
description	Description of the title	string	REQUIRED
shortDescription	Shortened description of the title	string	RECOMMENDED
hrPosterURL	url of the horizontal orientation of poster. (640*360)	url	REQUIRED
vrPosterURL	url of the vertical orientation of poster (400*600)	url	REQUIRED
releaseDate	the actual release date of the tile	yyyy-mm-dd	REQUIRED
startYear	Applicable for Shows, the year of first air time	yyyy	REQUIRED
endYear	Applicable for Shows, the year of last air time	yyyy	OPTIONAL
resolution	Max. resolution of the content 0 - No info 1 - HD 2 - FHD 3 - 4K 4 - 8K	integer	REQUIRED
pictureQuality	Max. picture quality of the content 0 - No info 1 - HDR 2 - HDR 10+ 3 - Dolby Vision	integer	REQUIRED
runtime	total duration of the title in seconds	integer	REQUIRED
genres	Can include up to 5 genre types	string array	REQUIRED
keywords	Keywords associated with the title	string array	RECOMMENDED
languages	content language	string array	REQUIRED
regions	contains the region in which the title was produced	string array	OPTIONAL
channels	name of the channel to which the title belongs	string array	RECOMMENDED
directors	contains name of the director(s)	string array	REQUIRED
writers	Contains name of the producer(s)/writer(s)	string array	RECOMMENDED

principalCast	contains name of the lead actors	string array	REQUIRED
seasonNumber	Applicable only for Shows, 0 for title types which are not Shows	integer	REQUIRED
averageRating	average individual rating on a scale of 0 to 10	float	RECOMMENDED
playingType	Type of video playback: 0 for VOD 1 for LIVE 2 for Time shifted	integer	REQUIRED
payType	Type of payment: 0 for free 1 for free with ADS 2 for Pay-per-view 3 for subscription	integer	REQUIRED
numEpisodes	Applicable for Shows, state the number of episode available.	integer	REQUIRED
isFinal	Indicates if a TV Series ends. True by default	boolean	REQUIRED
age	minimum viewing age required	integer	REQUIRED
episodes			
id	unique identifier of the title	integer	REQUIRED
parentTitleId	unique identifier of the parent , actual show title id	integer	REQUIRED
episodeNumber	number of the episode	integer	REQUIRED
title	title of the episode	string	REQUIRED
description	description of the episode	string	REQUIRED
releaseDate	date of release for this episode	yyyy-mm-dd	REQUIRED
downloadable	Is the episode downloadable or not	boolean	OPTIONAL
playDuration	total duration of the episode in seconds	integer	REQUIRED
previewUrl	poster url of the episode	url	REQUIRED
soundEffect	0 = Normal 1 = Dolby 2 = Dolby Atmos 3 = DTS	integer	REQUIRED
deeplink	intent to the particular episode	string	REQUIRED

Note

- The deep link provided in the API should open the player directly. The intent should not open a details page within the app, in such cases the Test Team would flag the mismatch in requirements
- Partners are requested to use android intents instead of web url for deep links
- Incase of partners unable to produce metadata , please request waiver for appropriate metadata.
- Image poster links shared in the API should be devoid of any additional branding , such as advertisements, Ratings, Status tags such as 'Free' etc.
- Content Rating issued by CBFC, Age and Description are mandated by Ministry of Electronics and Information Technology, Government of India. PatchWall will not be able to serve contents to users which do not meet the government mandates and a waiver will not be shared.
- Any content having Age < 16 will be considered safe for Kids viewing, therefore please ensure the age rating is attributed to appropriate content.
- Content Partners are expected to check the API before sharing it for integration, PatchWall Team will not be responsible for testing API. PatchWall will team only test the final API endpoint shared for Production Integration.

PatchWall Customization

Overview

PatchWall allows content partner to access PatchWall libraries and use them to add better user experience.

- Play History
- Subscription History
- Content Purchase History
- Back button journey
- Xiaomi Attribution

Requirements

- Add the following permissions to your app manifest
 - `<uses-permission android:name="com.mitv.patchwall.permission.MANAGE_MEDIA"/ >`
 - `<package android:name = "com.mitv.tvhome.atv"/ >`
- Integrate the PatchWall JAR file which can be found at [Google Drive Link](#) (Clickable)
- Code Samples can be found here: [Google Drive Link](#) (Clickable)

Play History

PatchWall supports play history support synchronization for all content partners. Partners can leverage this interface to offer users the ability to resume their contents. The play history from the partner would be available in the Recent row on User Centre

REQUIREMENTS

Partners need to integrate PatchWall-media-access.jar into the package. For access to the file, please contact your Integration Partner in PatchWall Team. Once the PatchWall-media-access jar file is integrated, use the following classes for history record related data processing in PatchWall Media Database.

```
import com.mitv.patchwall.support.media.HistoryVideo;  
import com.mitv.patchwall.support.media.PatchWallUtils;
```

Insert action is to add new content into the database and update action is to modify already added content in the database.

Method Name	Parameter Type	Description
setMediaId	string	Set media ID for history
setTitle	string	Set title for history
setCP	int	Set CP ID for history
setHorizontalPoster	string	Set horizontal poster for history
setEpisodeMediaId	string	Set episode ID for history

setEpisodeId	string	Set episode series number for history
setSeasonId	string	Set season number for history
setPosition	long	Set position for history
setDuration	long	Set duration for history
setAppLinkIntentUri	string	Set intent for history
setAge	string	Set age info for history
setExtra	string	Set extra field for history

```
import com.mitv.patchwall.support.media.HistoryVideo;
import com.mitv.patchwall.support.media.PatchWallUtils;
```

```
# insert play history
```

```
private void insertHistoryVideo(String history_id) {
    Intent intent = new Intent("com.mitv.patchwall.mediaaccess.INTENT");
    HistoryVideo.Builder builder = new HistoryVideo.Builder(history_id, intent.toUri(0), "0-18");
    builder.setTitle("Sample History")
        .setCP(200001)
        .setVerticalPoster("https://cdn.ksyru0-fusion.fds.api.mi-img.com/mitv/10007/0/cf96c2475c6b76b9fd68c0ef83408af3.webp")
        .setHorizontalPoster("https://cdn.ksyru0-fusion.fds.api.mi-img.com/mitv/10007/0/cf96c2475c6b76b9fd68c0ef83408af3.webp")
        .setEpisodeId("6")
        .setEpisodeMediaId("355546")
        .setSeasonId("2")
        .setExtra("")
        .setPosition(800)
        .setDuration(2768);
    if (PatchWallUtils.existHistoryVideo(this, history_id)) {
        PatchWallUtils.updateHistoryVideo(this, builder.build());
    } else {
        PatchWallUtils.insertHistoryVideo(this, builder.build());
    }
}
```

```
# query play history by media id
```

```
private HistoryVideo queryHistoryVideo(String history_id) {
    return PatchWallUtils.queryHistoryVideo(this, history_id);
}
```

```
# query all play history records
```

```
private List<HistoryVideo> queryHistoryVideos() {
    return PatchWallUtils.queryHistoryVideos(this);
}
```

```
# delete play history record by media id
```

```
private void deleteHistoryVideo(String history_id) {
    PatchWallUtils.deleteHistoryVideo(this, PatchWallUtils.VOD_HISTORY, history_id);
}
```

```
# delete all play history records
```

```
private void deleteHistoryVideos() {
    PatchWallUtils.deleteAllHistoryVideos(this);
}
```

Subscription History

PatchWall supports subscription history support synchronization for all content partners. This interface allows partners to share a subscription status of the user.

REQUIREMENTS

Partners need to integrate PatchWall-media-access.jar into the package. For access to the file, please contact your Integration Partner in PatchWall Team. Once the PatchWall-media-access jar file is integrated, use the following classes for history record related data processing in PatchWall Media Database.

```
import com.mitv.patchwall.support.media.PatchWallUtils;
import com.mitv.patchwall.support.media.Subscription;
```

When a user signs in to the app with an active subscription, partner is required to update the subscription status in the database. This subscription data stays locally on the device and will be used to curate content for the users. Please use the following method:

Method Name	Parameter Type	Description
setSubscriptionId	string	Set subscription ID for history
setTitle	string	Set title for history
setHorizontalPoster	string	Set horizontal poster for history
setAppLinkIntentUri	string	Set intent for history
setExtra	string	Set extra field for history

```
import com.mitv.patchwall.support.media.PatchWallUtils;
import com.mitv.patchwall.support.media.Subscription;
```

```
# insert subscription
private void insertSubscription(String subscriptionId) {
    Intent intent = new Intent("com.mitv.patchwall.mediaaccess.INTENT");
    Subscription.Builder builder = new Subscription.Builder(subscriptionId, intent.toUri(0));
    builder.setTitle("Subscription")
        .setHorizontalPoster("https://timgsa.baidu.com/timg?
image&quality=80&size=b9999_10000&sec=1548839506397&di=ab308660ad64b59126bde361b578cc09&imgtype=0&src=http%3A
%2F%2Fmmbiz.qpic.cn%2Fmmbiz_jpg%2FicFOjK989UV62pZQINfhKeNsVXIRkKicLz1U2VkrDFDdr6VMvleHrbekMuL0DicNbQDMR8shv
XSo54O3SYqYp1akA%2F640%3Fwx_fmt%3Djpeg")
        .setDescription("lalala")
        .setBeginTime(155555555)
        .setDueTime(655555555)
        .setCP(6114)
        .setExtra("");
    if (PatchWallUtils.existSubscription(this, subscriptionId)) {
        PatchWallUtils.updateSubscription(this, builder.build());
    } else {
        PatchWallUtils.insertSubscription(this, builder.build());
    }
}
```

```
# query subscription basis subscription id
private Subscription querySubscription(String subscriptionId) {
```



```

        return PatchWallUtils.querySubscription(this, subscriptionId);
    }
# query all subscriptions
private List<Subscription> querySubscriptions() {
    return PatchWallUtils.querySubscriptions(this);
}
# delete subscription record basis subscription id
private void deleteSubscription(String subscriptionId) {
    PatchWallUtils.deleteSubscription(this, subscriptionId);
}
# delete all subscription records
private void deleteSubscriptions() {
    PatchWallUtils.deleteSubscriptions(this);
}

```

Content Purchase History

PatchWall supports content purchase history synchronization for all content partners. This interface allows partners to share a purchase history of the user. This should be used in event of an rental or purchase of content.

```

import com.mitv.patchwall.support.media.PatchWallUtils;
import com.mitv.patchwall.support.media.TransactionVideo;

```

Method	Parameter Type	Description
type	int	The unique ID to describe the history type. The value is 0 for TVOD
transaction_id	string	The purchased content ID which is same as in the content API
title	string	The title of the purchased content
pay_time	long	The time when the user completes the transaction
begin_time	long	The start time of the transaction
due_time	long	The due time of the transaction
price	float	The price of the transaction
vertical_poster	string	The vertical poster of the content
horizontal_poster	string	The horizontal poster of the content
app_link_intent_uri	string	The deeplink to the video playback of that content
age	string	The age restriction applicable to the purchased content. Eg: all,16+,18+
extra	string	Extra field for future implementations

```

import com.mitv.patchwall.support.media.PatchWallUtils;
import com.mitv.patchwall.support.media.TransactionVideo;

#inserting content purchase
private void insertTvod(String transactionId) {
    long currentTime = System.currentTimeMillis();
    Intent intent = new Intent("com.mitv.patchwall.mediaaccess.INTENT");
    TransactionVideo.Builder builder = new TransactionVideo.Builder(transactionId, intent.toUri(0), "0-18");
    builder.setTitle("Sample Tvod")
        .setPayTime(currentTime)
        .setBeginTime(currentTime)
        .setDescription("Sample Description")
        .setDueTime(currentTime + DateUtils.DAY_IN_MILLIS * 3)
        .setPrice(10)
        .setVerticalPoster("https://cdn.ksyu0-fusion.fds.api.mi-img.com/mitv/10007/0/45bd36e8dacaca8776530e97a82a68c2.webp");

    if (PatchWallUtils.existTransactionalVideo(this, transactionId)) {
        PatchWallUtils.updateTransactionalVideo(this, builder.build());
    } else {
        PatchWallUtils.insertTransactionalVideo(this, builder.build());
    }
}

#query using transaction id
private TransactionVideo queryTvod(String transactionId) {
    return PatchWallUtils.queryTransactionalVideo(this, transactionId);
}

#query all transaction history records
private List<TransactionVideo> queryTvods() {
    return PatchWallUtils.queryTransactionalVideos(this, null);
}

#delete using transaction id
private void deleteTvod(String transactionId) {
    PatchWallUtils.deleteTransactionalVideo(this, transactionId);
}

#delete all transaction records
private void deleteTvods() {
    PatchWallUtils.deleteTransactionalVideos(this);
}

```

Back button journey

- Content Partners are expected to have a custom back button journey when the brand is identified to be Xiaomi, this custom back button journey is expected to be following standard android back button navigation. Where the active state returns to the last activity.
Eg: If a journey starts from PatchWall activity, it should return to PatchWall activity.

Xiaomi Business Attribution

- For any attribution, please use brand to find attribution to Xiaomi

```

String brand = android.os.Build.BRAND;
if(!"Xiaomi".equals(brand)){
    // Do something
}

```

Ref: <https://developer.android.com/reference/android/os/Build#BRAND>

Dolby Atmos Support Identification

In case content partner uses | enables Dolby Atmos for their contents, it may sometime be confusing to identify whether a particular Xiaomi Smart TV Model support Dolby Atmos. The standard method of identifying support for Dolby Atmos may mislead the content partner app, therefore we request content partner to use the below method to identify if a Xiaomi Smart TV Platform can support Dolby Atmos.

```
import org.json.JSONException;
import org.json.JSONObject;
import android.provider.Settings;

private boolean getNrdpAtmosEnabled(ContentResolver mResolver) {
    boolean isEnabled = false;
    String audioCap = Settings.Global.getString(mResolver, "nrdp_audio_platform_capabilities");
    if (audioCap == null) {
        Log.i(TAG, "audiocap is null");
        return false;
    }

    try {
        JSONObject rootObject = new JSONObject(audioCap);
        JSONObject audioCapsObject = rootObject.getJSONObject("audiocaps");
        JSONObject atmosObject = audioCapsObject.getJSONObject("atmos");
        if (atmosObject.has("enabled")) {
            isEnabled = atmosObject.getBoolean("enabled");
        }
        else
        {
            isEnabled = false;
        }
    } catch (org.json.JSONException e) {
        e.printStackTrace();
    }
    return isEnabled;
}
```

Dolby Digital Plus Identification

Using android AudioManager.getDevices Function

```
AudioManager audio = (AudioManager) mContext.getSystemService(Context.AUDIO_SERVICE);
AudioDeviceInfo[] devices = audio.getDevices(audio.GET_DEVICES_OUTPUTS);
for (AudioDeviceInfo device: devices) {
    Log.i(TAG, "Supported Encodings" + Arrays.toString(device.getEncodings()));
}
```

Checklist for Content Partner for Google TV Adoption

1. Package name should be same as the package deployed on androidtv | Google TV Play Store
2. Package to include the following PatchWall features
 - Play History
 - Subscription History
 - Content Purchase History
 - Back button journey
3. Attribution to be added basis OEM equals to 'Xiaomi'
4. Content | Catalog API shared by Content Partners to include deep links which work with the Google TV Play Store Package.

Change Log

4.2.1 [18 May 2023]

- Added Detection reference for Dolby Digital Plus
- Added checklist for Content partner moving to Play Store Packages

4.2.0 [3 April 2023]

- Corrected poster resolution data in API responses
- Stream Access Integration method removed

4.1.0 [23 January 2023]

- Consolidated Integration
- Google TV Adoption

4.0.5 [17 October 2022]

- Added package instructions for new androidtv versions
- Clear demarcation for API & APK Requirements

4.0.4 [27 September 2022]

- Added library for TVOD DB Operations

4.0.3 [17 August 2022]

- Updated new Device Series

4.0.2 [27 April 2022]

- Updated new Device Series

4.0.1 [22 November 2021] Revision

- Added critical info for integration
- Corrected hr_poster resolution

4.0.0 [14 October 2021] Urgent Use case

- Adopting to PatchWall Global Policies

2.4.7 [10 August 2021]

- Clarification on Live Channels for stream and in-app playback added

2.4.6 [28 July 2021]

- Added LIVE channel API section

2.4.5 [7 July 2021]

- Deprecated support for vertical poster on Play History

2.4.4 [23 June 2021]

- Added checklist table for Package Customizations

2.4.3 [18 June 2021]

- Added separate format for TV Show's Title Details to avoid miscommunications
- Added fields for :
 - Resolution
 - Picture Quality
 - Sound Quality

2.4.2 [20 April 2021]

- Added changes to TVOD history
- Added min Poster Resolution

2.4.1[3 Feb 2021]

- Added Status Check on Mi Pay

2.4.0 [17 September 2020]

- Mi Pay Integration Added (not final)

2.3.0 [21 April 2020]

- Subscription Status Operations Added
- TVOD Status Operations Added
- Improved guidelines for Package