FRUITION
The future begins today...

USER GUIDE

Product Review and Rating Manager

Fruition Consultancies
info@fruitionconsultancies.com

Contents

Introduction

The **Product Reviews Management App** is an Adobe App Builder–based solution developed by **Fruition** to manage product ratings and reviews for Adobe Commerce SaaS storefronts.

As Adobe Commerce SaaS no longer provides native backoffice support for product reviews, this app acts as a **centralized, enterprise-ready replacement**, allowing merchants to collect, moderate, migrate, and manage reviews while maintaining full control over brand content and customer trust.

Who Should Use This App

This app is designed for:

- **Ecommerce administrators** managing catalog and customer feedback
- **Marketing teams** focused on conversion, trust, and brand reputation
- **Operations teams** migrating from legacy or PaaS platforms
- **Support teams** responding to customer feedback

Core Capabilities

1. Collect Product Reviews

The app enables review collection through secure APIs:

- Customers can submit **ratings and reviews** directly from the storefront
- Admin users can **create reviews manually**, useful for:
    - Offline feedback
    - Curated testimonials
    - Customer support–generated reviews

All submitted reviews are stored centrally and processed through moderation workflows.

2. Moderate and Control Reviews

To ensure brand-safe content, the app provides:

- A moderation queue for **pending reviews**
- Actions to **approve or reject** reviews before publication
- Clear visibility into review status and history

This ensures only validated, compliant reviews appear on the storefront.

3. Reply to Reviews as a Brand

Admin users can:

- Respond to customer reviews using the **“Reply as Brand”** feature

- Provide clarifications, support responses, or acknowledgements

Brand replies are exposed via APIs and can be rendered on the storefront alongside customer reviews.

4. Import Reviews from External Systems

The app supports **CSV-based review import**, allowing merchants to:

- Migrate reviews from legacy PaaS systems
- Import ratings from third-party platforms
- Preserve historical customer feedback during replatforming

This ensures continuity of social proof during platform transitions.

5. Review Summary and APIs

The app exposes separate APIs for:

- **Detailed Product Reviews** (individual ratings and comments)
- **Product Review Summary** (average rating, total count, distribution)

These APIs can be used by:

- Storefront applications
- Headless commerce setups
- Reporting and analytics systems

---

6. Admin Dashboard

The Admin dashboard provides:

- A consolidated view of all reviews
- Review summaries per product
- Actionable queues for moderation (approve/reject)

This allows teams to manage reviews efficiently without manual tracking.

---

7. Audit Logs and Traceability

For governance and compliance, the app maintains:

- Audit logs for review creation and updates
- Visibility into moderation actions

- Traceability of admin activity

This supports internal audits and accountability.

How the App Fits into Your Commerce Stack

- Built natively on **Adobe App Builder**
- Designed to integrate seamlessly with **Adobe Commerce SaaS**
- API-first architecture supports:
  - Headless storefronts
  - AEM boilerplate based frontends
  - Custom UI implementations

Benefits for Your Business

- Restores essential product review functionality removed from SaaS backoffice
- Improves customer trust and conversion through social proof
- Gives marketing teams control over brand engagement
- Simplifies review management during platform migrations
- Provides enterprise-grade moderation and auditability

Prerequisites

Admin UI SDK – Installation & Configuration

Installation:

- **PaaS**
  - Please follow the instructions Install or update Adobe Commerce Admin UI SDK
  - Admin UI SDK version 3.0 or higher
- **SaaS**
  - Included already

IMS Module for Authentication

For PaaS: Please follow the instructions Adobe Identity Management Service (IMS) for Adobe Commerce

For SaaS: SaaS instances already include IMS configuration

Note: the request schemas, response schemas and mesh.json needs to be created in the api mesh app builder project before deploying the app to the workspace mentioned in below section of document.

Project Setup

Installing from App Page

This step walks you through installing the app for **Adobe Commerce** via the Adobe Exchange marketplace.

1. Go to Adobe Exchange, select "Experience Cloud" and search for **Product Review and Rating Manager**.
2. Click button **Buy** or **Install** and follow the prompts. The application will be automatically deployed to your Adobe App Builder environment.

Please refer to the page https://developer.adobe.com/developer-distribution/experience-cloud/docs/guides/discoverAndManage/app-builder-discover

Setting up the required configurations

- Adobe Commerce Base URL

  Note: When configuring the COMMERCE_BASE_URL environment variable, the format differs between PaaS and SaaS:

  For PaaS (On-Premise/Cloud):

  - Must include your base site URL + /rest/ suffix
  - Example: https://[environment-name].us-4.magentosite.cloud/rest/

  For SaaS:

  - Must be the REST API endpoint provided by Adobe Commerce
  - Example: https://na1-sandbox.api.commerce.adobe.com/[tenant-id]/

  Make sure to use your actual environment name or tenant ID in the URL **and the URL should end in a slash(/)**. The examples above use placeholder values.

Please provide below configurations

- OAUTH_CLIENT_ID
- OAUTH_CLIENT_SECRETS
- OAUTH_TECHNICAL_ACCOUNT_ID
- OAUTH_TECHNICAL_ACCOUNT_EMAIL
- OAUTH_IMS_ORG_ID
- COMMERCE_CONSUMER_KEY (PaaS, OAuth, if IMS Auth isn't being used)
- COMMERCE_CONSUMER_SECRET (PaaS, OAuth, if IMS Auth isn't being used)
- COMMERCE_ACCESS_TOKEN (PaaS, OAuth, if IMS Auth isn't being used)

- COMMERCE_ACCESS_TOKEN_SECRET (PaaS, OAuth, if IMS Auth isn't being used)

Note:

For PaaS:

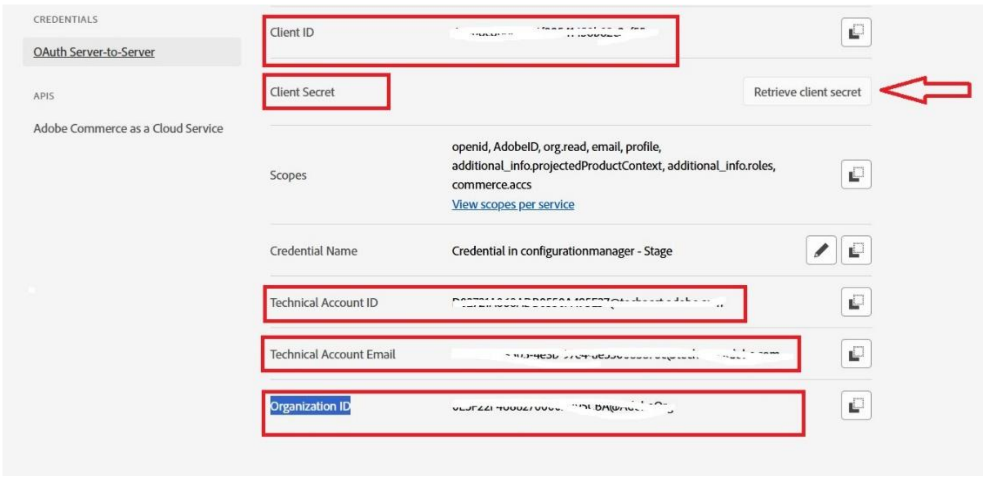
Case1: With IMS Auth - Given that the IMS module will be enabled for use with Admin UI SDK, the OAuth credentials could also be used to authenticate with PaaS. This process requires a Commerce instance with [Adobe Identity Management Service (IMS) for Adobe Commerce]() configured.

Case2: If IMS Auth isn't being used, but OAuth is used – The values (COMMERCE_CONSUMER_KEY, COMMERCE_CONSUMER_SECRET, COMMERCE_ACCESS_TOKEN, COMMERCE_ACCESS_TOKEN_SECRET) can be retrieved from a Commerce integration, please refer to https://experienceleague.adobe.com/en/docs/commerce-admin/systems/integrations

For SaaS: SaaS instances already include IMS configuration

Use the following steps to create OAuth credentials for App Builder authentication:

- Go to Adobe Developer Console
    - https://developer.adobe.com/console
    - Log in with your Adobe ID (same org where your project is created).
- Access your IMS credentials through the Adobe Developer Console. Select any project and workspace within the IMS organization. Then click OAuth Server-to-Server in the side-navigation menu.
- NOTE: These credentials are automatically populated in Configure OAuth Server-to-Server Credential.
    - OAUTH_CLIENT_ID=<client id> (Found in Developer Console → your Project → Credentials → **Client ID**.)
    - OAUTH_CLIENT_SECRETS=<client secrets> (Found in Developer Console → your Project → Credentials → **Client Secrets**.)
    - OAUTH_TECHNICAL_ACCOUNT_ID=<technical account id> (Found in Developer Console → your Project → Credentials → **Technical Account ID**.)
    - OAUTH_TECHNICAL_ACCOUNT_EMAIL=<technical account email> (Found in Developer Console → your Project → Credentials → **Technical Account Email**.)
    - OAUTH_SCOPES=<scopes>
    - OAUTH_IMS_ORG_ID=<ims org> (Found in Developer Console → your Project → Credentials → **Organization ID**.)


CREDENTIALS

OAuth Server-to-Server

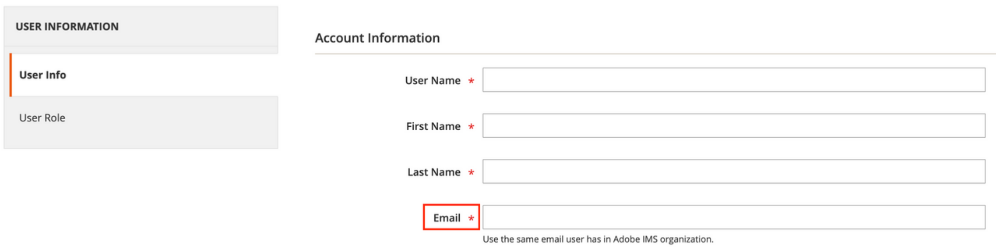
APIS

Adobe Commerce as a Cloud Service

Client ID	[redacted]
Client Secret	Retrieve client secret
Scopes	openid, AdobeID, org.read, email, profile, additional_info.projectedProductContext, additional_info.roles, commerce.accs View scopes per service
Credential Name	Credential in configurationmanager - Stage
Technical Account ID	[redacted]
Technical Account Email	[redacted]
Organization ID	[redacted]

- Provide the technical account with access to the Commerce instance:
    - **SaaS** only The technical account is automatically created and associated with the Commerce instance once the first request is made using the OAuth credentials.
    - **PaaS** only Add a technical account with server-to-server credentials to the Commerce Admin with the appropriate permissions using the Admin User Creation Guide.

    When associating the user with the account, find your Technical Account email as a part of generated IMS credentials with following pattern: ***<technical-account>@techacct.adobe.com*** and use that value in the Email field during user creation:


USER INFORMATION

User Info

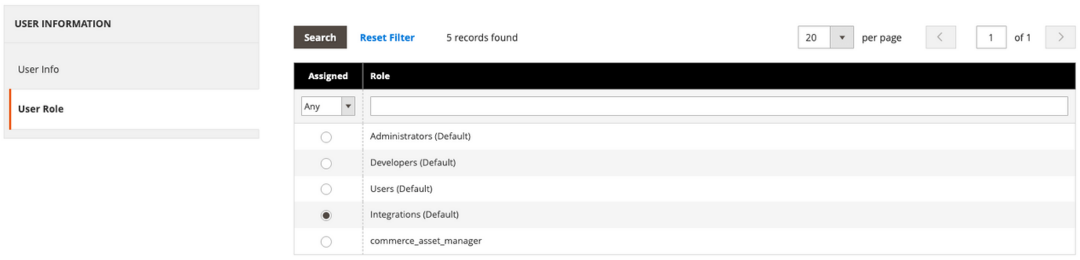
User Role

Account Information

User Name *	
First Name *	
Last Name *	
Email *	

Use the same email user has in Adobe IMS organization.

On the User Role tab, select the role that provides all necessary permissions for API integrations.


USER INFORMATION

User Info

User Role

Search | Reset Filter | 5 records found | 20 per page | 1 of 1

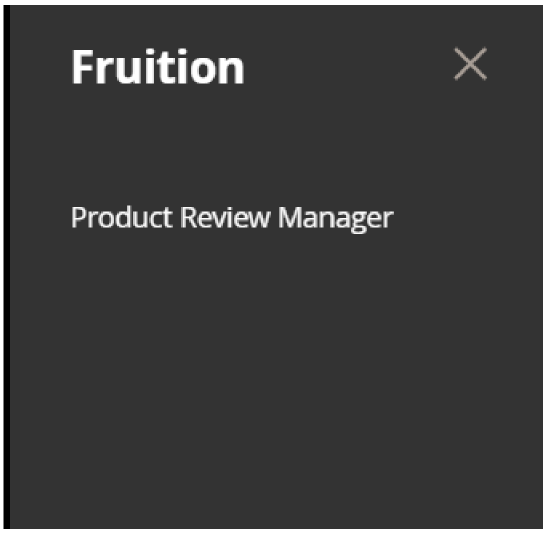
Assigned	Role
Any	
○	Administrators (Default)
○	Developers (Default)
○	Users (Default)
●	Integrations (Default)
○	commerce_asset_manager

Locate Configurations in Admin Panel

Configuration:

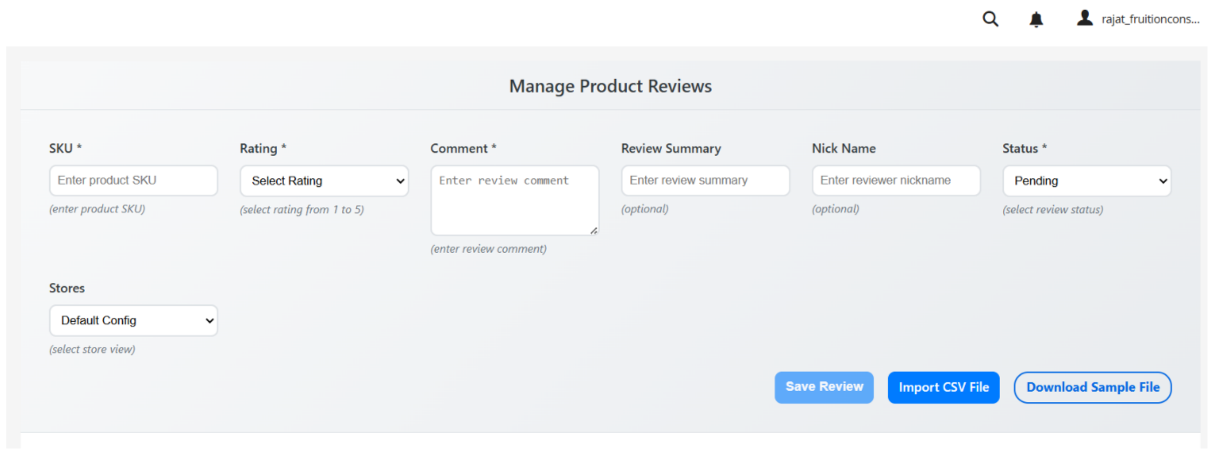
Please follow the instructions Admin configuration and testing

Once the App is installed and configured via Admin SDK UI, you will be able to see the Menu with name "Fruition", when you click on it, a sub menu will appear "Product Review Manager".



Landing on Product Review Manager

When you click on **Fruition Product Review Manager**, the **Configuration Manager screen** opens. From here, you can **add, edit, delete, or view** existing reviews.



Manage Product Reviews

SKU *	Rating *	Comment *	Review Summary	Nick Name	Status *
Enter product SKU	Select Rating	Enter review comment	Enter review summary	Enter reviewer nickname	Pending
(enter product SKU)	(select rating from 1 to 5)	(enter review comment)	(optional)	(optional)	(select review status)

Stores
Default Config
(select store view)

Add a Review

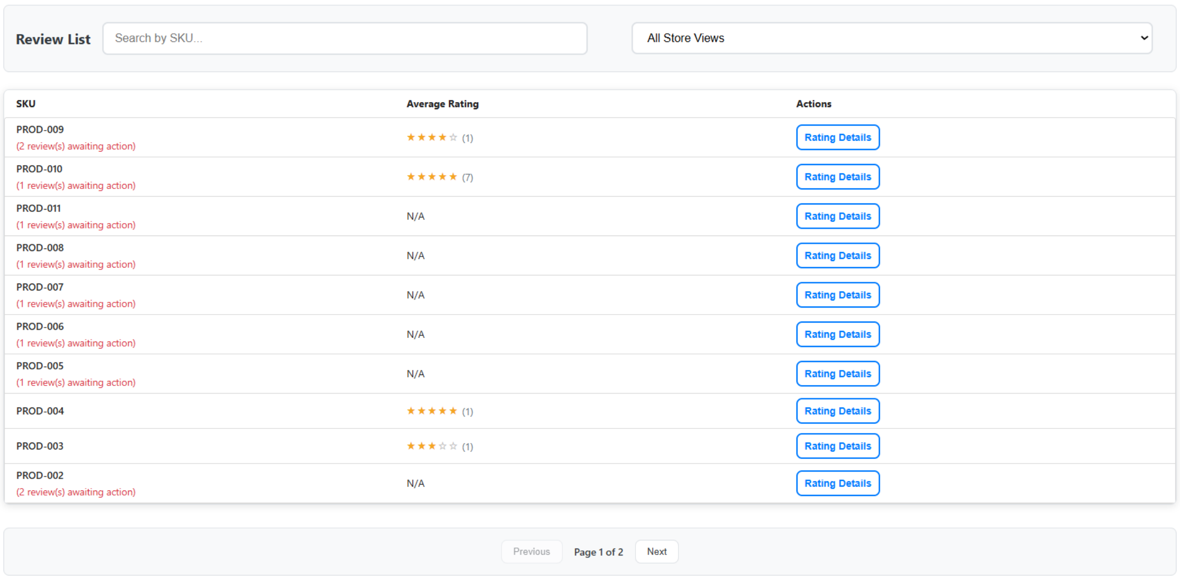
To add a new configuration:

- Enter the **SKU, Rating, Comment as mandatory fields.**
- Specify Review Summary, Nick Name and Status as optional fields.
- Select the appropriate **Store** where this configuration should apply.
- Save the review.

Search and Listing of Review

User can search existing configurations using filters such as **SKU** or **Store ID**.

- If no filters are applied, the system will display **all available reviews**.
- User can view Rating Details against Product
- Average Rating is the average of rating of Approved status.
- Notification for Admin user to take action for Pending Reviews.



Review List | Search by SKU... | All Store Views

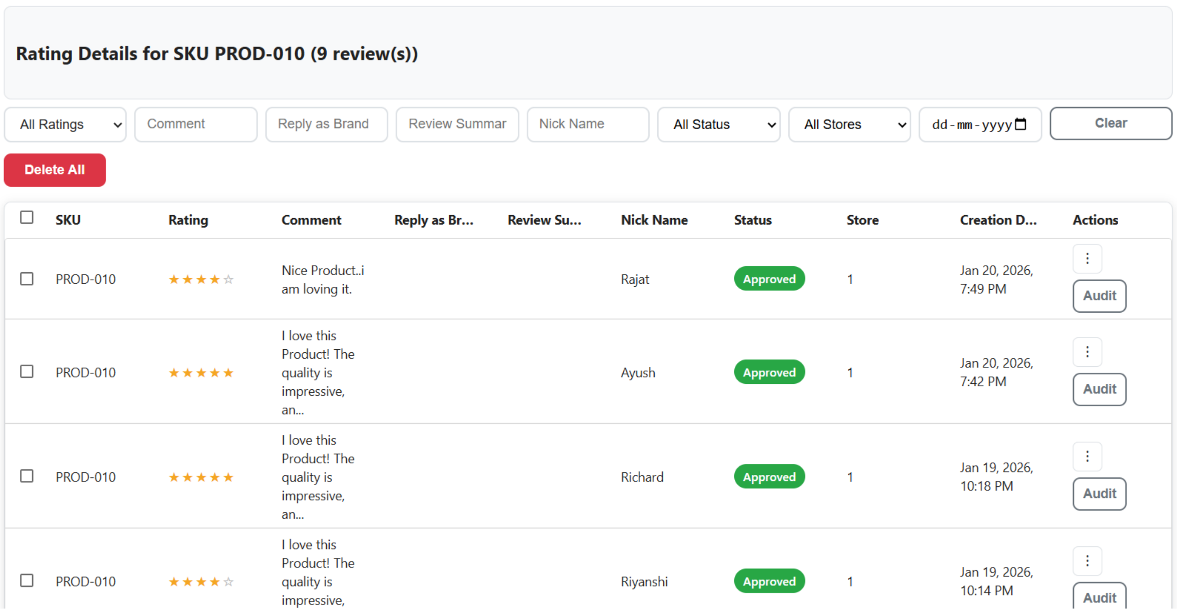
SKU	Average Rating	Actions
PROD-009 (2 review(s) awaiting action)	★★★★☆ (1)	Rating Details
PROD-010 (1 review(s) awaiting action)	★★★★★ (7)	Rating Details
PROD-011 (1 review(s) awaiting action)	N/A	Rating Details
PROD-008 (1 review(s) awaiting action)	N/A	Rating Details
PROD-007 (1 review(s) awaiting action)	N/A	Rating Details
PROD-006 (1 review(s) awaiting action)	N/A	Rating Details
PROD-005 (1 review(s) awaiting action)	N/A	Rating Details
PROD-004	★★★★★ (1)	Rating Details
PROD-003	★★★☆☆ (1)	Rating Details
PROD-002 (2 review(s) awaiting action)	N/A	Rating Details

Previous | Page 1 of 2 | Next

Product Level Review Listing & Actions

User can view and search all the ratings received against the product selected.

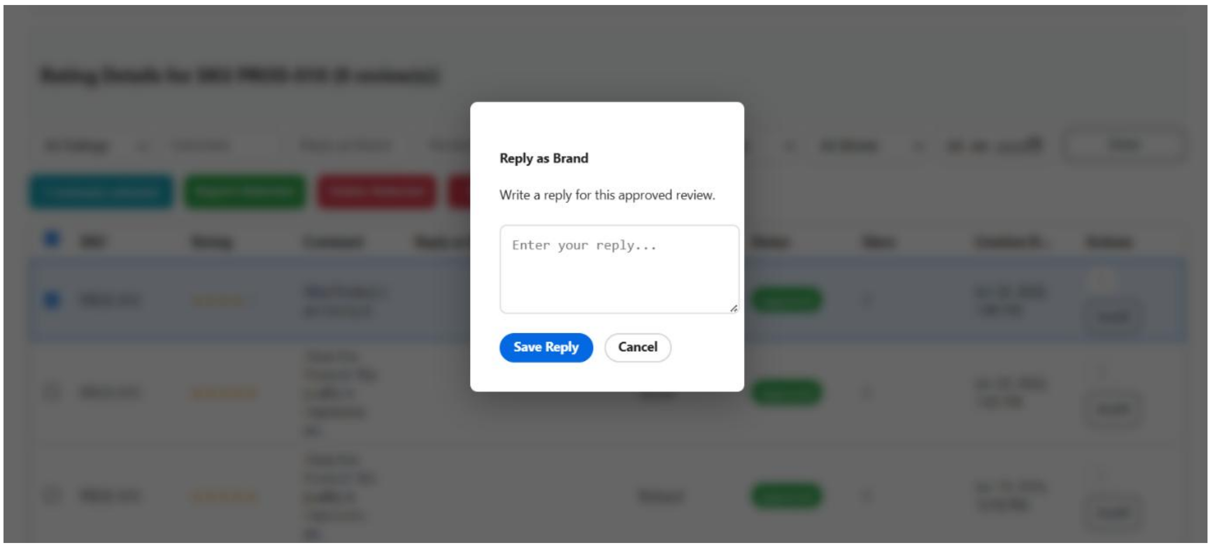
1. User can search with various options like Ratings, Comment, Reply as Brand, Review Summary, Nick Name, Status, Stores, Creation Date
2. User can edit the review and mark it Approve / Reject



Rating Details for SKU PROD-010 (9 review(s))

	SKU	Rating	Comment	Reply as Br...	Review Su...	Nick Name	Status	Store	Creation D...	Actions
☐	PROD-010	★★★★☆	Nice Product..i am loving it.			Rajat	Approved	1	Jan 20, 2026, 7:49 PM	⋮ Audit
☐	PROD-010	★★★★★	I love this Product! The quality is impressive, an...			Ayush	Approved	1	Jan 20, 2026, 7:42 PM	⋮ Audit
☐	PROD-010	★★★★★	I love this Product! The quality is impressive, an...			Richard	Approved	1	Jan 19, 2026, 10:18 PM	⋮ Audit
☐	PROD-010	★★★★☆	I love this Product! The quality is impressive,			Riyanshi	Approved	1	Jan 19, 2026, 10:14 PM	⋮ Audit

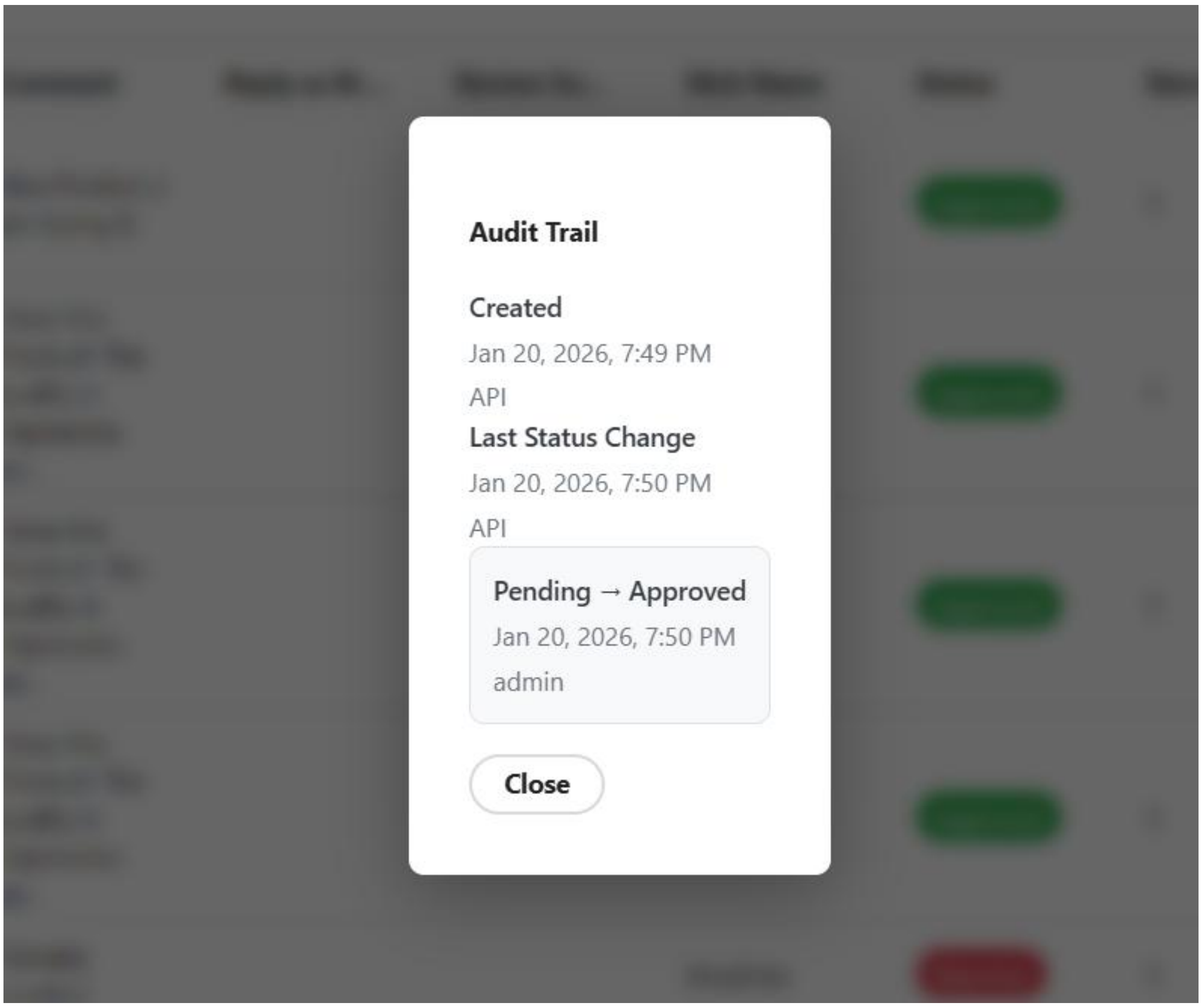
3. Review can be moderated as well by clicking Edit button
4. Once Review is rejected, then user can't make any changes to it.
5. Once Review is accepted, user can reply as Brand, so that it can be visible in storefront.



Reply as Brand

Write a reply for this approved review.

6. User can check Audit trail to see who has created or updated the review / rating.


Audit Trail

Created

Jan 20, 2026, 7:49 PM

API

Last Status Change

Jan 20, 2026, 7:50 PM

API

Pending → Approved

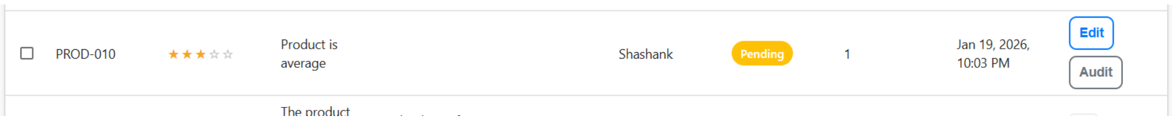
Jan 20, 2026, 7:50 PM

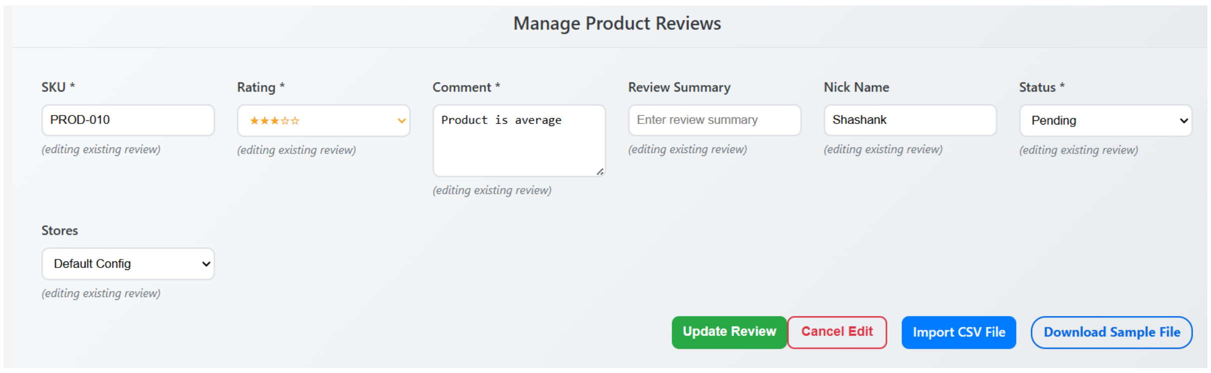
admin

Moderate Review

To edit an existing review with Pending status:

1. Select the review you want to modify with pending status
2. Click the **Edit** link.
3. The current values will be displayed with editable fields.
4. Make the required changes.
5. Click **Update Review** to save your changes.


☐	PROD-010	★★★☆☆	Product is average	Shashank	Pending	1	Jan 19, 2026, 10:03 PM	Edit Audit


Manage Product Reviews

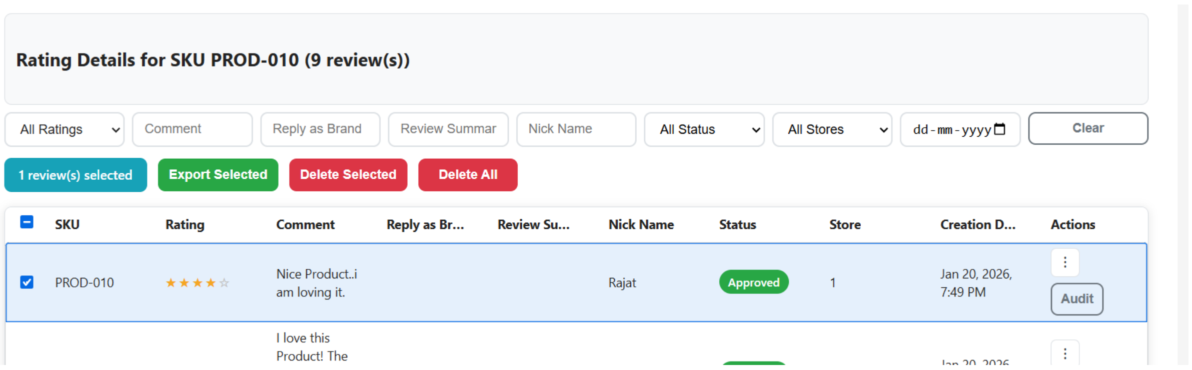
SKU *	Rating *	Comment *	Review Summary	Nick Name	Status *
PROD-010	★★★☆☆	Product is average	Enter review summary	Shashank	Pending
(editing existing review)	(editing existing review)	(editing existing review)	(editing existing review)	(editing existing review)	(editing existing review)

Stores: Default Config (editing existing review)

Delete Review

To delete a review:

1. Select the checkbox next to the review you want to remove.
2. You can select multiple review(s) if needed.
3. Click the **Delete Selected** button.
4. Confirm the deletion when prompted.


Rating Details for SKU PROD-010 (9 review(s))

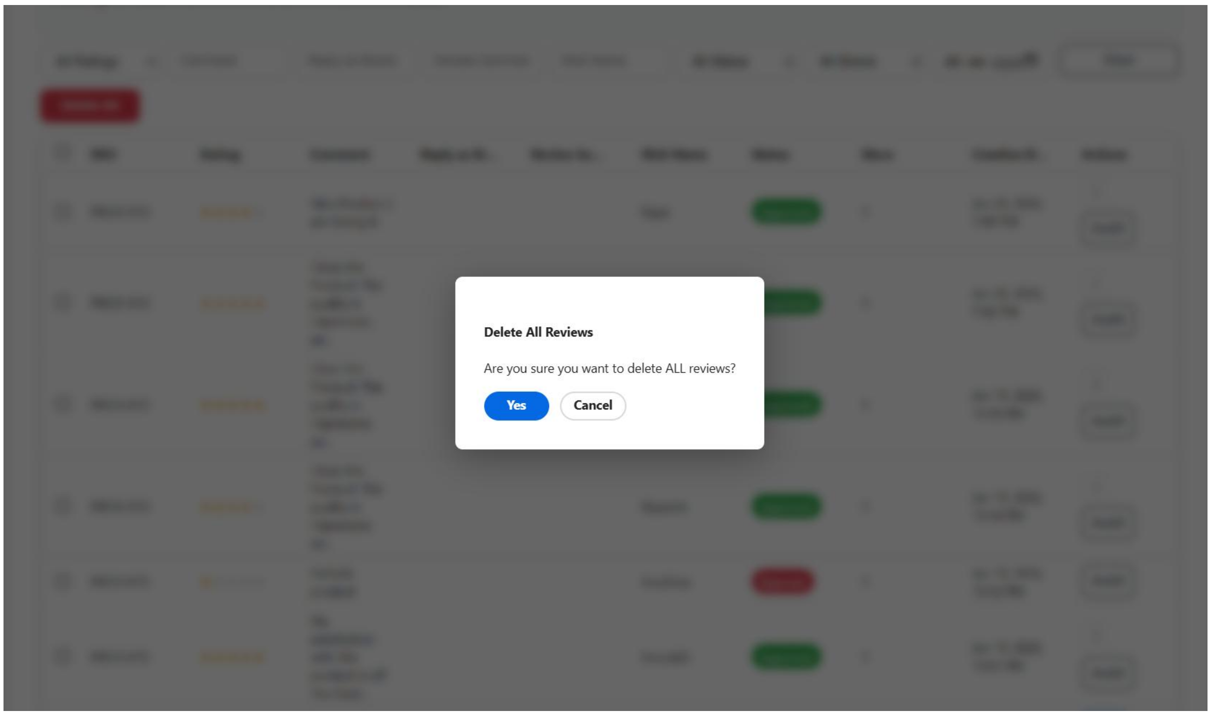
	SKU	Rating	Comment	Reply as Br...	Review Su...	Nick Name	Status	Store	Creation D...	Actions
☑	PROD-010	★★★★☆	Nice Product..i am loving it.			Rajat	Approved	1	Jan 20, 2026, 7:49 PM	⋮ Audit
			I love this Product! The							⋮

Delete All Reviews

To remove all existing reviews in one step:

1. Click the **Delete All** button.
2. Confirm the deletion when prompted.

⚠️ **Note:** This action is irreversible and will permanently delete all reviews from the system.


Delete All Reviews

Are you sure you want to delete ALL reviews?

API Endpoints

Base URL

https://<ACTION_URL>.adobeio-static.net/api/v1/web/productreview_manager/

<ACTION_URL> should be replaced with the actual deployed URL.

Submit Review

This endpoint post review data to the Product Review Manager. It is a simple HTTP POST request that returns the data posted to system. Create a new product review entry.

HEADER

x-gw-ims-org-id: <OrganizationID>

Authorization: Bearer <Auth Token>

Request

Endpoint: https://<ACTION_URL>.adobeio-static.net/api/v1/web/productreview_manager/submitReview

Method: POST

Response

- **Status Code**: 200 OK
- **Content-Type**: application/json

Request Body

The request contains the following structure:

- **JSON**

```
{
  "sku": "PROD-001",
  "rating": "5",
  "comment": "Excellent product!",
  "reviewSummary": "Great quality product",
  "nickName": "John Doe",
  "storeid": "0"
}
```

- **Required Fields**
  - `sku` (string): Product SKU
  - `rating` (string): Rating value (must be "1", "2", "3", "4", or "5")
  - `comment` (string): Review comment text
- **Optional Fields**
  - `reviewSummary` (string): Brief summary of the review. Default: `""` (empty string)
  - `nickName` (string): Reviewer's nickname. Default: `""` (empty string)
  - `storeid` (string): Store identifier. Default: `"0"`

Response Body

The response contains the following structure:

- **JSON**

```
{
    "message": "Review submitted successfully",
    "review": {
        "comment": "Excellent product!",
        "created_at": "2026-01-21T03:35:46.371Z",
        "id": 31,
        "nickName": "John Doe",
        "rating": "4",
        "reviewSummary": "Great quality product",
        "sku": "PROD-001",
        "status": "Pending",
        "storeid": "0"
    },
    "success": true
}
```

Get Product Review

This endpoint retrieves product review based on specified filters. It allows users to query reviews by providing a filter object that includes the desired parameters.

HEADER

x-gw-ims-org-id: <OrganizationID>

Authorization: Bearer <Auth Token>

Request

- **Method**: POST or GET
- **Endpoint:** https://<ACTION_URL>.adobeio-static.net/api/v1/web/productreview_manager/getProductReview
- **Request Body** (JSON): (for POST)
    - filter: An object containing the following keys:
        - `sku` (string): Filter by product SKU
        - `rating` (string): Filter by rating (1-5)
        - `comment` (string): Filter by comment text (partial match)
        - `reviewSummary` (string): Filter by review summary (partial match)
        - `nickName` (string): Filter by reviewer nickname
        - `status` (string): Filter by status (`Pending`, `Approved`, `Rejected`)
        - `storeid` (string): Filter by store identifier
        - `reply` (string): Filter by brand reply text

Example Request Body

- **JSON**

```
{
  "filter": {
    "sku": "PROD-001",
    "rating": "5",
    "comment": "excellent",
    "reviewSummary": "great",
    "nickName": "John",
    "status": "Approved",
    "storeid": "0",
    "reply": "thank you"
  }
}
```

Response

- **Status Code**: 200 OK
- **Content-Type**: application/json

- **Response Body** (JSON):

Example Response Body

- **JSON**

```json
{
    "data": [
        {
            "id": 2,
            "sku": "PROD-001",
            "rating": "5",
            "comment": "Excellent product!",
            "reviewSummary": "Great quality product",
            "nickName": "John Doe",
            "reply": "",
            "status": "Approved",
            "storeid": "0",
            "created_at": "2026-01-17T09:05:49.348Z",
            "updated_at": "2026-01-17T10:23:10.571Z"
        },
        {
            "id": 3,
            "sku": "PROD-001",
            "rating": "4",
            "comment": "Excellent product!",
            "reviewSummary": "Great quality product",
            "nickName": "John Doe1",
            "reply": "",
            "status": "Approved",
            "storeid": "0",
            "created_at": "2026-01-17T10:07:17.828Z",
            "updated_at": "2026-01-17T12:08:25.942Z"
        },
        {
            "id": 7,
            "sku": "PROD-001",
            "rating": "5",
            "comment": "Excellent product!",
            "reviewSummary": "Great quality product",
            "nickName": "Rajat",
            "reply": "",
            "status": "Approved",
            "storeid": "0",
            "created_at": "2026-01-17T10:20:51.403Z",
            "updated_at": "2026-01-17T10:50:19.046Z"
        },
        {
            "id": 25,
            "sku": "PROD-001",
            "rating": "5",
            "comment": "Excellent product! Very satisfied with the quality and
delivery.",
            "reviewSummary": "Great quality product",
            "nickName": "JohnDoe",
            "reply": "thanks1",
            "status": "Approved",
            "storeid": "0",
            "created_at": "2026-01-17T14:37:25.304Z",
            "updated_at": "2026-01-17T15:13:25.370Z"
        }
    ],
    "total": 4,
    "summary": [
        {
            "sku": "PROD-001",
            "storeid": "0",
            "approvedCount": 4,
            "averageRating": 4.75
        }
    ],
    "filters": {
        "sku": "PROD-001",
        "status": "Approved"
    }
}
```

- **Data:** An array of product review objects that match the filter criteria, with the following properties:
    - id (integer): The unique identifier of the review.
    - sku (string): The product SKU associated with the review.
    - rating (string): The rating given by the customer for the product.
    - comment (string): The detailed review comment provided by the customer.
    - reviewSummary (string): A short summary of the review.
    - nickName (string): The display name of the customer who submitted the review.
    - reply (string): The reply added by the store/admin to the review, if any.
    - status (string): The approval status of the review (e.g., Approved).
    - storeid (string): The store identifier to which the review belongs.
    - created_at (string): The timestamp when the review was created.
    - updated_at (string): The timestamp when the review was last updated.

- **Summary:** An array containing aggregated review information per product and store, with the following properties:
  - sku (string): The product SKU for which the summary is calculated.
  - storeid (string): The store identifier associated with the summary.
  - approvedCount (integer): The total number of approved reviews for the product.
  - averageRating (number): The average rating calculated from approved reviews.
- **Filters:** An object reflecting the filters applied in the request, such as product SKU and review status.

Notes

- Ensure that the key parameter is provided to filter the product reviews effectively.
- The response will include an array of product reviews that match the provided filter criteria, along with metadata about the total number of reviews found.

Get Product Review Summary

This endpoint retrieves product review summary with status Approved only based on specified filters. It allows users to query reviews by providing a filter object that includes the desired parameters.

HEADER

x-gw-ims-org-id: <OrganizationID>

Authorization: Bearer <Auth Token>

Request

- **Method**: POST or GET
- **Endpoint**: https://<ACTION_URL>.adobeio-static.net/api/v1/web/productreview_manager/getProductReviewSummary
- **Request Body** (JSON): (For POST)
  - filter: An object containing the following keys:
    - `sku` (string): Filter by product SKU
    - `rating` (string): Filter by rating (1-5)
    - `comment` (string): Filter by comment text (partial match)
    - `reviewSummary` (string): Filter by review summary (partial match)
    - `nickName` (string): Filter by reviewer nickname
    - `storeid` (string): Filter by store identifier
    - `reply` (string): Filter by brand reply text

Example Request Body

- **JSON**

```
{
  "filter": {
    "sku": "PROD-001",
    "rating": "5",
    "comment": "excellent",
    "reviewSummary": "great",
    "nickName": "John",
    "storeid": "0",
    "reply": "thank you"
  }
}
```

Response

- **Status Code**: 200 OK
- **Content-Type**: application/json
- **Response Body** (JSON):

Example Response Body

- **JSON**

```
{
    "summaries": [
        {
            "sku": "PROD-001",
            "storeid": "0",
            "approvedCount": 4,
            "averageRating": 4.75
        }
    ],
    "total": 1,
    "filters": {
        "sku": "PROD-001",
        "status": "Approved"
    }
}
```

- **summaries**: An array of review summary objects that match the filter criteria, with the following properties:
  - **sku** (string): The product SKU for which the review summary is generated.
  - **storeid** (string): The store identifier associated with the product reviews.
  - **approvedCount** (integer): The total number of approved reviews for the product.
  - **averageRating** (number): The average rating calculated from the approved reviews.
- **total** (integer): The total number of summary records returned.

- **filters**: An object reflecting the filters applied in the request, such as product SKU and review status.

Instructions for Integrating an EDS / Headless Storefront with app

Below are steps to integrate this app with an EDS Storefront or Headless App:

1. **Setting up EDS storefront –** Please follow the steps mentioned in the link https://experienceleague.adobe.com/developer/commerce/storefront/get-started/create-storefront/
2. **API Exposure from the App** – The Product Review Manager exposes product review and rating data via APIs. – This is already there in App Builder App. Refer to section API Endpoints in this document.
3. **Consumption in the EDS / Headless Storefront** – An EDS / Headless storefront consumes these APIs and could render the product review and rating.
    a. Consume the API
    b. Call API in the block like product-details.js
        a. const reviewData = await fetchProductReviews(sku);

```
async function fetchProductReviews(sku) {
 try {
  const response = await
fetch(`${REVIEWS_API_BASE}/getProductReview`, {
   method: 'POST',
   headers: {
    'Content-Type': 'application/json',
   },
   body: JSON.stringify({
    filter: {
     sku,
    },
   }),
  });
  if (!response.ok) {
   throw new Error('Failed to fetch product reviews.');
  }
  return await response.json();
 } catch (error) {
  console.error('Review fetch failed.', error);
  return { data: [], summary: [] };
 }
}
```

c. Render the product Rating at desired place

Note:

- The steps under "Consumption in the EDS / Headless Storefront" are just one example of how to use the app's configurations.
- the request schemas, response schemas and mesh.json needs to be created in the api mesh app builder project before deploying the app to the workspace

Configuring a Mesh

The getProductReviewSummary, submitReview and getProductReview endpoint can be added as a source to a mesh, the mesh can securely store the auth credentials needed for requests to the getProductReviewSummary, submitReview and getProductReview endpoint. Please refer to the link for more information Command Reference

Sample mesh configuration snippet for the endpoint:

```
{
 "meshConfig": {
  "sources": [
   {
    "name": "productReviewManager",
    "handler": {
     "JsonSchema": {
      "baseUrl": "<ACTION_URL>/api/v1/web/productreview_manager/ ",
      "operations": [
       {
        "type": "Query",
        "field": "getProductReview",
        "path": "/getProductReview",
        "method": "GET",
        "responseSchema": "./responseSchema/get-product-review.json"
       },
       {
        "type": "Query",
        "field": "getProductReviewByFilters",
        "path": "/getProductReview",
        "method": "POST",
        "requestSchema": "./requestSchema/get-product-review.json",
        "responseSchema": "./responseSchema/get-product-review.json"
       },
       {
        "type": "Query",
        "field": "getReviewSummary",
        "path": "/getProductReviewSummary",
        "method": "GET",
        "responseSchema": "./responseSchema/get-review-summary.json"
       },
       {
        "type": "Query",
        "field": "getReviewSummaryByFilters",
        "path": "/getProductReviewSummary",
        "method": "POST",
        "requestSchema": "./requestSchema/get-review-summary.json",
        "responseSchema": "./responseSchema/get-review-summary.json"
       },
       {
        "type": "Mutation",
        "field": "submitReview",
        "path": "/submitReview",
        "method": "POST",
          "requestSchema": "./requestSchema/submit-review.json",
          "responseSchema": "./responseSchema/submit-review.json"
         },
         {
          "type": "Query",
          "field": "getStores",
          "path": "/getStores",
          "method": "GET",
          "responseSchema": "./responseSchema/get-stores.json"
         },
         {
          "type": "Query",
          "field": "getStoresByPost",
          "path": "/getStores",
          "method": "POST",
          "responseSchema": "./responseSchema/get-stores.json"
         }
        ],
        "operationHeaders": {
         "Authorization": "{context.headers['authorization']}",
         "x-gw-ims-org-id": "{context.headers['x-gw-ims-org-id']}",
         "Content-Type": "application/json"
        }
       }
      }
     }
   ]
  }
}
```

< ACTION_URL> - Replace with your URL https://.adobeio-static.net/api/v1/web/productreview_manager

requestSchema/get-product-review.json

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "title": "Get Product Review Request",
 "type": "object",
 "properties": {
  "filter": {
   "type": "object",
   "properties": {
    "sku": { "type": "string" },
    "rating": { "type": "string" },
    "comment": { "type": "string" },
    "reviewSummary": { "type": "string" },
    "nickName": { "type": "string" },
    "status": { "type": "string" },
    "storeid": { "type": "string" },
    "reply": { "type": "string" }
   },
   "additionalProperties": false
  }
 },
 "additionalProperties": false
}
```

requestSchema/submit-review.json

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "title": "Submit Review Request",
 "type": "object",
 "properties": {
  "sku": { "type": "string" },
  "rating": { "type": "string" },
  "comment": { "type": "string" },
  "reviewSummary": { "type": "string" },
  "nickName": { "type": "string" },
  "status": { "type": "string" },
  "storeid": { "type": "string" }
 },
 "required": ["sku", "rating", "comment"],
 "additionalProperties": false
}
```

requestSchema/get-review-summary.json

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "title": "Get Review Summary Request",
 "type": "object",
 "properties": {
  "filter": {
   "type": "object",
   "properties": {
    "sku": { "type": "string" },
    "rating": { "type": "string" },
    "comment": { "type": "string" },
    "reviewSummary": { "type": "string" },
    "nickName": { "type": "string" },
    "status": { "type": "string" },
    "storeid": { "type": "string" },
    "reply": { "type": "string" }
   },
   "additionalProperties": false
  }
 },
  "additionalProperties": false
}
```

requestSchema/get-stores.json

```
{
"$schema":"http://json-schema.org/draft-07/schema#",
"title":"Get Stores Request",
"type":"object",
"properties":{},
"additionalProperties":false
}
```

responseSchema/get-stores.json

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Get Stores Response",
  "type": "object",
  "properties": {
    "data": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": { "type": "string" },
          "code": { "type": "string" },
          "name": { "type": "string" }
        },
        "required": ["id", "code", "name"],
        "additionalProperties": false
      }
    }
```

```
 },
 "required": ["data"],
 "additionalProperties": false
}
```

responseSchema/get-product-review.json

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "title": "Get Product Review Response",
 "type": "object",
 "properties": {
  "data": {
   "type": "array",
   "items": {
    "type": "object",
    "properties": {
     "id": { "type": "integer" },
     "sku": { "type": "string" },
     "rating": { "type": "string" },
     "comment": { "type": "string" },
     "reviewSummary": { "type": "string" },
     "nickName": { "type": "string" },
     "status": { "type": "string" },
     "storeid": { "type": "string" },
     "reply": { "type": "string" },
     "created_at": { "type": "string" },
     "updated_at": { "type": "string" }
    },
    "required": ["id", "sku", "rating", "comment", "status", "storeid", "created_at"],
    "additionalProperties": true
   }
  },
  "total": { "type": "integer" },
  "filters": { "type": ["object", "null"] },
  "summary": {
   "type": "array",
   "items": {
    "type": "object",
    "properties": {
     "sku": { "type": "string" },
     "storeid": { "type": "string" },
     "approvedCount": { "type": "integer" },
     "averageRating": { "type": "number" }
    },
    "required": ["sku", "storeid", "approvedCount", "averageRating"],
    "additionalProperties": false
   }
  }
 },
 "required": ["data", "total"],
 "additionalProperties": false
}
```

responseSchema/submit-review.json

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Submit Review Response",
  "type": "object",
  "properties": {
   "success": { "type": "boolean" },
   "message": { "type": "string" },
   "review": {
    "type": "object",
    "properties": {
      "id": { "type": "integer" },
      "sku": { "type": "string" },
      "rating": { "type": "string" },
      "comment": { "type": "string" },
      "reviewSummary": { "type": "string" },
      "nickName": { "type": "string" },
      "status": { "type": "string" },
      "storeid": { "type": "string" },
      "created_at": { "type": "string" }
    },
    "required": ["id", "sku", "rating", "comment", "status", "storeid", "created_at"],
    "additionalProperties": true
   }
  },
  "required": ["success"],
  "additionalProperties": false
}
```

responseSchema/get-review-summary.json

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Get Product Review Summary Response",
  "type": "object",
  "properties": {
   "summaries": {
    "type": "array",
    "items": {
     "type": "object",
     "properties": {
      "sku": {
        "type": "string"
      },
      "storeid": {
        "type": "string"
      },
      "approvedCount": {
        "type": "integer"
      },
      "averageRating": {
        "type": "number"
      }
     },
     "required": ["sku", "storeid", "approvedCount", "averageRating"],
     "additionalProperties": false
    }
   },
   "total": {
    "type": "integer"
   },
   "filters": {
    "type": ["object", "null"]
   }
  },
  "required": ["summaries", "total"],
  "additionalProperties": false
}
```

Examples of Creating Secret File

Create a YAML file, such as secrets.yaml, to define your secrets. The file name must end with the yaml or yml file extension. Each line in the files defines a different secret.

Example:

```
authorization: ${authorization_token}
```

x-gw-ims-org-id: ${x-gw-ims-org-id}

Create or update your mesh secrets

When you create or update a mesh that you want to include secrets in, add the --secrets flag followed by the path to your secrets file. If you do not provide the secrets file when updating a mesh that has secrets, the secrets` values are overwritten by their literal references.

Create:

```
aio api-mesh create mesh.json --secrets secrets.yaml
```

Update:

```
aio api-mesh update mesh.json --secrets secrets.yaml
```

For more information about mesh secret, please visit Secrets management

Support

In case of any questions, please contact support@fruitionconsultancies.com