

 Add icon  Add cover

9.15 Approval Required for Public Dataset Comments

As a Dataset Owner

I want to approve comments on Public Datasets before they are shown on the Dataset or Resource page

So that I can check comments are appropriate before they are visible

Acceptance Criteria:

- 1. When a (logged-in) user posts a comment on a Public Dataset, it must be approved by the Dataset owner
- 2. When a user posts a comment, it should be clear to them that the comment is pending approval (e.g. on-screen message or posted comment is greyed out with 'pending' flag
- 3. Dataset owner gets a notification and can either approve or delete the comment
- 4. There should be a toggle in the 'create dataset' form which allows the Dataset owner to toggle the 'Approval required for public comments' feature for the dataset.

Overview

Enable comment moderation workflow where comments on public datasets require dataset owner approval before becoming visible to other users.

Backend Components

1. Database Schema Extension

	≡ Field	≡ Type	≡ Description
1	<u>status</u>	ENUM	<u>pending</u> , <u>approved</u> , <u>rejected</u>
2	<u>moderated_by</u>	UUID	User who approved/rejected
3	<u>moderated_at</u>	TIMESTAMP	When moderation occurred

New Table: `trusted_commenter`

	≡ Field	≡ Type	≡ Description
1	<u>id</u>	UUID	Primary key
2	<u>dataset_id</u>	UUID	FK to dataset
3	<u>user_id</u>	UUID	FK to trusted user

4	<u>trusted_by</u>	UUID	Dataset owner who granted trust
5	<u>created_at</u>	TIMESTAMP	When trust was granted

2. API Endpoints

	Endpoint	Method	Description
1	<u>/api/comment/create</u>	POST	Create comment (status=pending unless user trusted)
2	<u>/api/comment/approve/{id}</u>	POST	Approve pending comment (owner only)
3	<u>/api/comment/approve-trust/{id}</u>	POST	Approve comment + trust user for dataset
4	<u>/api/comment/reject/{id}</u>	DELETE	Reject/delete pending comment (owner only)
5	<u>/api/comment/pending</u>	GET	List pending comments for owner's datasets
6	<u>/api/dataset/{id}/trusted-users</u>	GET	List trusted commenters for dataset
7	<u>/api/dataset/{id}/trust-user/{user_id}</u>	POST	Add trusted commenter
8	<u>/api/dataset/{id}/untrust-user/{user_id}</u>	DELETE	Remove trusted commenter

3. Business Logic

- Determine if dataset is public → check if user is trusted → set comment status
- If user is trusted for dataset → auto-approve (status = `approved`)
- If user is NOT trusted → set status to `pending`
- Send notification to dataset owner on new pending comment
- Filter comments by status when displaying (only show `approved` to non-owners)
- Allow comment author to see their own pending comments
- Trust cascades to approval - approving with trust also approves the comment

4. Authorization Rules

	≡ Action	≡ Allowed Users
1	View pending	Comment author, Dataset owner, Admin
2	Approve/Reject	Dataset owner, Admin
3	Approve & Trust	Dataset owner, Admin
4	Manage trusted users	Dataset owner, Admin
5	View approved	All authenticated users

Frontend Components

1. Comment Form Enhancement

- Add info notice: "Comments require approval before becoming visible"
- Show success message post-submission: "Awaiting approval"

2. Pending Comment Display

- Greyed-out card with dashed border
- "Pending Approval" badge
- Yellow notice box for commenter

3. Moderation Actions (Owner View)

- **Approve dropdown button** with two options:
 - "Approve Comment" - approve this comment only
 - "Approve & Trust User" - approve + auto-approve future comments from this user
- Delete button

4. Notification Integration

- New notification type: `comment_pending_approval`
- **Approval dropdown** in notification with:
 - "Approve this comment"
 - "Approve & trust this user"

5. Dataset create/update modification

'Approval required for public comments' this needs to be implemented in the dataset to create screen through the **ckanext-scheming** as a checkbox, it would automatically get added and saved while creating and updating. This field should appear conditionally i.e. when in Visibility field "Public" is selected then only it should be shown, Also the placement of this checkbox should only be below the Visibility field.

1. The addition of this field can be achieved via the ckanext-scheming but to show conditionally Javascript needs to be used. Here is a sample JavaScript module to achieve the same.

Example Requirement

You have:

Mermaid ▾

```

1 dataset_fields:
2   - field_name: visibility
3     label: Visibility
4     preset: select
5     choices:
6       - value: public
7         label: Public
8       - value: private
9         label: Private
10
11   - field_name: approval_comments
12     label: Approval required for public comments
13     preset: checkbox

```

Requirement:

- When `visibility = public` → show `approval_comments`
- When `visibility = private` → hide `approval_comments`

Custom JavaScript

This is the most common approach.

Create a CKAN fanstatic/webassets JS module:

JavaScript ▾

```

1 ckan.module('conditional_fields', function ($) {
2   return {
3     initialize: function () {
4
5       function toggleCheckbox() {
6         var visibility = $('[name="visibility"]').val();
7
8         if (visibility === 'public') {
9           $('[data-field="approval_comments"]').show();
10        } else {
11          $('[data-field="approval_comments"]').hide();
12        }
13      }
14
15      $('[name="visibility"]').on('change', toggleCheckbox);
16
17      toggleCheckbox();
18    }
19  };
20 });

```

Then attach it in your dataset form template:

Mermaid ▾

```

1 {% asset 'my_extension/conditional_fields' %}

```

This works with Scheming forms because the generated fields are standard HTML fields.

Effort Estimation

Backend Development

	☰ Task	☰ Hours
1	Database schema migration (comment + trusted_commenter)	3
2	Comment model updates	2
3	Create comment API (with pending/trusted logic)	3
4	Approve/Reject API endpoints	2
5	Approve & Trust API endpoint	2
6	Trusted user CRUD APIs	2
7	Authorization/permission checks	3
8	Notification trigger on new comment	3
9	List pending comments for owner	3
10	Unit tests	2
11	Backend Subtotal	25

Frontend Development

	☰ Task	☰ Hours
1	Comment form - pending notice UI	2
2	Success message after submission	1
3	Pending comment card styling	2
4	Approval dropdown (Approve / Approve & Trust)	2
5	Notification panel - approval dropdown	2
6	API integration (AJAX calls)	2
7	Loading states & error handling	2

8	Frontend tests	3
9	Ckanext-scheming changes	4
10	Frontend Subtotal	23

Integration & QA

	≡ Task	≡ Hours
1	End-to-end testing	4
2	Bug fixes & polish	4
3	Integration Subtotal	8

Summary

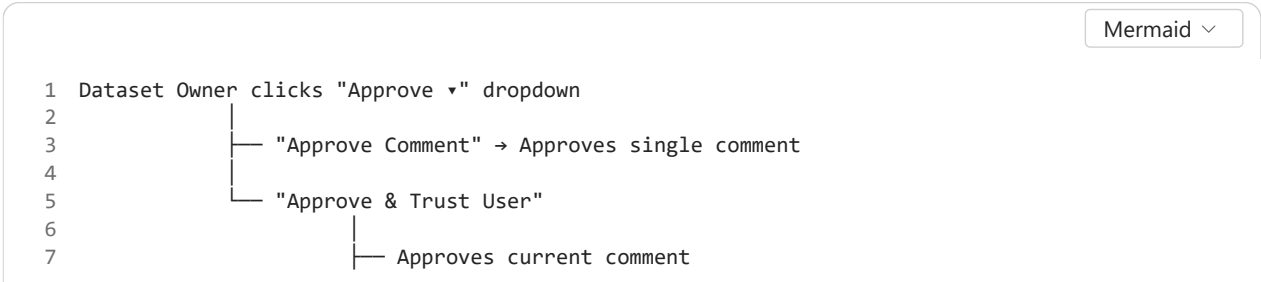
	≡ Category	≡ Hours
1	Backend	25
2	Frontend	23
3	Integration & QA	8
4	Total	56 hours

Estimated Duration: ~7 working days (1 developer)

Dependencies

- Existing comment extension infrastructure
- Notification system
- Dataset ownership/permission model

Trust User Flow



8
9

- └─ Adds user to trusted_commenter table for this dataset
- └─ Future comments from user → Auto-approved (skip pending)