

Task Activity Types — Frontend Guide

Activities are returned in the `activities` array of `GET api/v1/projects/{projectId}/tasks/{taskId}` .

Each activity has:

- `user_id` — who performed the action (resolve name from your users store)
- `action_type` — determines which display sentence to show
- `old_value` / `new_value` — data to interpolate into the sentence
- `created_at` — epoch ms timestamp, format as relative time (e.g. "2 hours ago")

Display pattern: `{User Full Name} {sentence}` — `{relative time}`

1. TASK_CREATED

Display sentence: `"{User}" created this task`

Frontend usage: Show as the first item in the activity feed. No values from `new_value` need to be shown in the sentence itself, but you can optionally show the initial status as a badge.

```
{
  "action_type": "TASK_CREATED",
  "old_value": null,
  "new_value": { "status_key": "DRAFT", "priority": "HIGH" },
  "created_at": 1771660432193
}
```

Example: **John Smith** created this task — *2 hours ago*

2. TASK_UPDATED

Display sentence: `"{User}" updated the task details`

Frontend usage: Use `old_value` and `new_value` to show a diff of what changed. Only show fields that are present in `new_value` . For example if only `priority` changed, show "updated priority from LOW to HIGH".

```
{
  "action_type": "TASK_UPDATED",
  "old_value": { "status_key": "DRAFT", "priority": "LOW" },
  "new_value": { "status_key": "DRAFT", "priority": "HIGH" },
  "created_at": 1771660432193
}
```

```

old_value : {
  "title": "Old Title",
  "description": null,
  "priority": "LOW",
  "category_id": null,
  "subcategory_id": null,
  "suite_number": null,
  "plan_id": null,
  "x_coordinate": null,
  "y_coordinate": null,
  "start_date": null,
  "end_date": null,
  "assigned_to": null
},
"new_value": { "title": "New Title", "priority": "HIGH" },
"created_at": 1771660432193
}

```

Example: **John Smith** updated the task details — *1 hour ago (diff: title: "Old Title" → "New Title", priority: LOW → HIGH)*

3. STATUS_CHANGED

Display sentence: "{User}" moved the task from {old_value.status_key} to {new_value.status_key}

Frontend usage: Use `old_value.status_key` and `new_value.status_key` directly in the sentence. Optionally render each status key as a colored badge using the workflow status colors.

```

{
  "action_type": "STATUS_CHANGED",
  "old_value": { "status_key": "DEFICIENCY" },
  "new_value": {
    "status_key": "COMPLETED",
    "flag_value_ids": ["uuid-of-flag-value"]
  },
  "created_at": 1771660432193
}

```

Example: **John Smith** moved the task from `DEFICIENCY` → `COMPLETED` — *30 minutes ago*

4. COMMENT_ADDED

Display sentence: "{User}" added a comment

Frontend usage: Optionally show a preview of `new_value.content` (truncated to ~60 chars). If `new_value.attachments_count > 0`, append "with {n} attachment(s)".

```
{
  "action_type": "COMMENT_ADDED",
  "old_value": null,
  "new_value": { "content": "John - Looks good", "attachments_count": 2 },
  "created_at": 1771660432193
}
```

Example: **John Smith** added a comment: *"John - Looks good"* with 2 attachment(s) — 20 minutes ago

5. COMMENT_DELETED

Display sentence: "{User}" deleted a comment

Frontend usage: Optionally show a preview of `old_value.content` as strikethrough or greyed out text to indicate what was removed.

```
{
  "action_type": "COMMENT_DELETED",
  "old_value": { "content": "John - Looks good" },
  "new_value": null,
  "created_at": 1771660432193
}
```

Example: **John Smith** deleted a comment — 15 minutes ago

6. ATTACHMENT_ADDED

Display sentence: "{User}" uploaded an attachment

Frontend usage: Use `new_value.is_serial` to differentiate — if `true`, show "uploaded a serial number photo". You can render a thumbnail link using `new_value.file_url`.

```
{
  "action_type": "ATTACHMENT_ADDED",
  "old_value": null,
  "new_value": {
    "file_url": "/uploads/task-attachments/task-1771660432193-123456789.jpg",
    "is_serial": false
  }
}
```

```
is_serial : raise
},
"created_at": 1771660432193
}
```

Example (regular): **John Smith** uploaded an attachment — *10 minutes ago* Example (serial): **John Smith** uploaded a serial number photo — *10 minutes ago*

7. ATTACHMENT_DELETED

Display sentence: "{User}" removed an attachment

Frontend usage: No extra data needed in the sentence. `old_value.file_url` is available if you want to show which file was removed.

```
{
  "action_type": "ATTACHMENT_DELETED",
  "old_value": {
    "file_url": "/uploads/task-attachments/task-1771660432193-123456789.jpg"
  },
  "new_value": null,
  "created_at": 1771660432193
}
```

Example: **John Smith** removed an attachment — *5 minutes ago*

8. CHECKLIST_UPDATED

This `action_type` covers 3 sub-cases. Differentiate using `new_value.action`.

8a. Item Added (`new_value.action === "item_added"`)

Display sentence: "{User}" added checklist item "{new_value.label}"

```
{
  "action_type": "CHECKLIST_UPDATED",
  "old_value": null,
  "new_value": { "action": "item_added", "label": "Check wiring" },
  "created_at": 1771660432193
}
```

Example: **John Smith** added checklist item "Check wiring" — *3 minutes ago*

8b. Item Updated (no `action` field in `new_value`)

Display sentence: "{User}" updated checklist item "{old_value.label}"

```
{
  "action_type": "CHECKLIST_UPDATED",
  "old_value": { "label": "Check wiring", "sort_order": 1, "is_completed": false },
  "new_value": { "label": "Check wiring updated", "sort_order": 1 },
  "created_at": 1771660432193
}
```

Example: John Smith updated checklist item *"Check wiring"* — 2 minutes ago

8c. Item Deleted (`new_value.action === "item_deleted"`)

Display sentence: "{User}" removed checklist item "{old_value.label}"

```
{
  "action_type": "CHECKLIST_UPDATED",
  "old_value": { "label": "Check wiring" },
  "new_value": { "action": "item_deleted" },
  "created_at": 1771660432193
}
```

Example: John Smith removed checklist item *"Check wiring"* — 1 minute ago

9. CHECKLIST_ITEM_TOGGLED

Display sentence:

- If `new_value.is_completed === true`: "{User}" marked "{old_value.label}" as complete
- If `new_value.is_completed === false`: "{User}" marked "{old_value.label}" as incomplete

Frontend usage: Check `new_value.is_completed` to determine which sentence to show. Use `old_value.label` for the item name.

```
{
  "action_type": "CHECKLIST_ITEM_TOGGLED",
  "old_value": { "label": "Check wiring", "sort_order": 1, "is_completed": false },
  "new_value": { "is_completed": true },
  "created_at": 1771660432193
}
```

Example (completed): **John Smith** marked "Check wiring" as complete  — just now

Example (uncompleted): **John Smith** marked "Check wiring" as incomplete — just now

10. RELATED_TASKS_UPDATED

Display sentence:

- If `new_value.related_task_ids.length > 0`: "{User}" updated related tasks ({n} linked)
- If `new_value.related_task_ids.length === 0`: "{User}" removed all related tasks

Frontend usage: Use `new_value.related_task_ids.length` to determine the count. You can optionally resolve task titles from your tasks store using the IDs.

```
{
  "action_type": "RELATED_TASKS_UPDATED",
  "old_value": null,
  "new_value": { "related_task_ids": ["uuid1", "uuid2"] },
  "created_at": 1771660432193
}
```

Example: **John Smith** updated related tasks (2 linked) — just now

Summary Table

action_type	Display Sentence
TASK_CREATED	{User} created this task
TASK_UPDATED	{User} updated the task details
STATUS_CHANGED	{User} moved the task from {old status} → {new status}
COMMENT_ADDED	{User} added a comment
COMMENT_DELETED	{User} deleted a comment
ATTACHMENT_ADDED	{User} uploaded an attachment / uploaded a serial number photo
ATTACHMENT_DELETED	{User} removed an attachment
CHECKLIST_UPDATED	{User} added / updated / removed checklist item "{label}"
CHECKLIST_ITEM_TOGGLED	{User} marked "{label}" as complete / incomplete

RELATED_TASKS_UPDATED	{User} updated related tasks ({n} linked) / removed all related tasks
-----------------------	---