Workflow States / Actions

Created by Joel Nave, last modified by Ralph Garcia on Sep 17, 2015

~~NPR-2~~ - Submit for approval TO DEMO/CLOSED | ~~NPR-1~~ - Reject a Request TO DEMO/CLOSED | ~~NPR-224~~ - Design Workflow visualization TO DEMO/CLOSED

Overview

There are 9 potential workflow states that a request can be in. States are entered or exited by either a system process or a user action. This page will define the workflow states, how they are entered and exited and how users will interact with and experience them.

Design Considerations

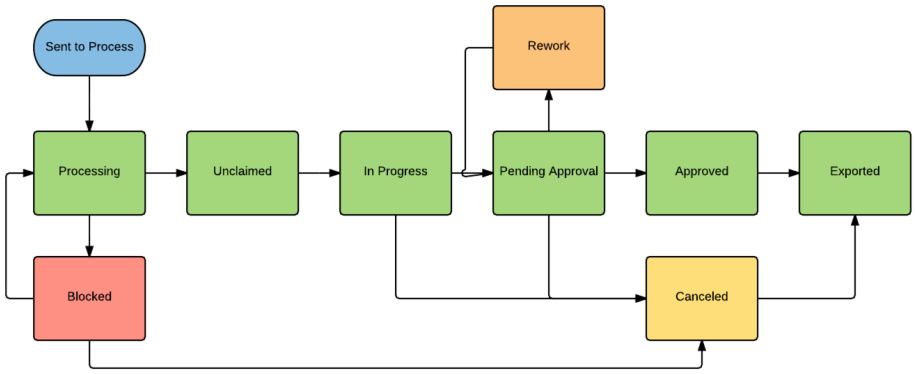
- The current status of a request must be clear to the user.
- The next available actions must be clear to the user.

Experience Design

Workflow State Summary

This section describes at a high level the available workflow states. Each workflow state will be documented in detail in following sections.

- There are 9 potential workflow states.
- The below diagram shows the movement between workflow states.
- The green states represent the "happy path" or ideal flow through the workflow states.



1. Processing
2. Blocked
3. Unclaimed
4. In Progress
5. Pending Approval
6. Rework
7. Approved
8. Cancelled (Rejected)
9. Exported

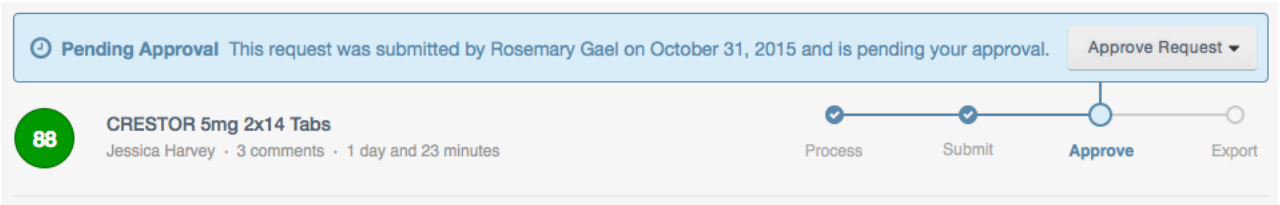
Workflow Visualization

We would like to provide a visualization of the workflow. There are two key goals with this:

1. Allow the user to quickly visualize where they are in the process.
2. Provide some visual interest to an otherwise boring page.

In this concept there are 4 steps:

1. Process
2. Submit
3. Approve
4. Export



Each step will map to one or more workflow states. The message area will list the *state*, while the visualization the *step*.

Step Visual States

The colors of the steps will vary and explained below.

- A current step is represented as an outlined filled circle that is a little larger than the other step circles.
- A completed step is represented as a non-outlined filled circle with a checkmark inside it.
- A blocked or canceled step is represented similar to a completed step, however, instead of the checkmark it will have an "X" icon in it.

Steps and States

Step	Step visual show as work in progress	Step visual shows as Completed	Notes
Process	• Processing • Blocked	After it has been successfully processed.	
Submit	• Unclaimed • In Progress • Validating • Rework	After the user has successfully submitted the request for approval.	
Approve	• Pending Approval • Validating Note that if a request has been sent for rework it will no longer show as work in progress for the Approve step.	After the the user has Approved or Canceled the request.	
Export	• Approved - not exported • Approved - exported • Canceled	After the user has exported an approved request.	

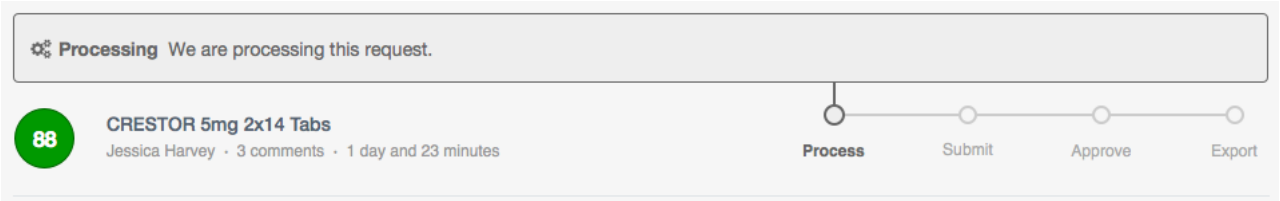
Process

There are four steps in the workflow Process, Submit, Approve and Export. Each step has more than one possible state as mentioned above. The text should remain static except for in certain cases which are highlighted below.

Processing State

Processing is the first state of the work flow.

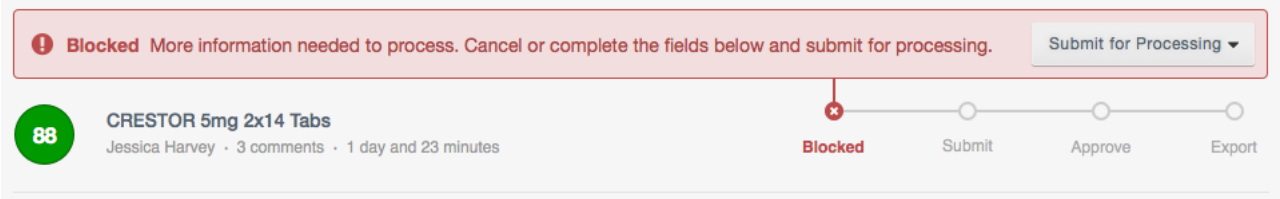
- Current step is represented as an outlined filled circle



Blocked State

The request is in a blocked state.

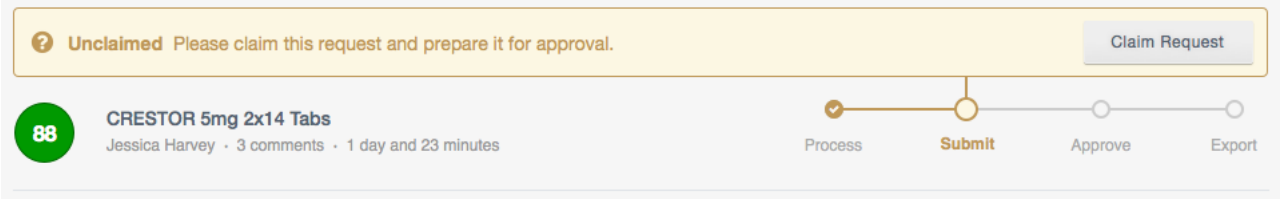
- Step 1 text is "Blocked" and red
- Current state dot is different than the normal current state look.
- An "X" icon is added to emphasize it has been blocked.



Submit

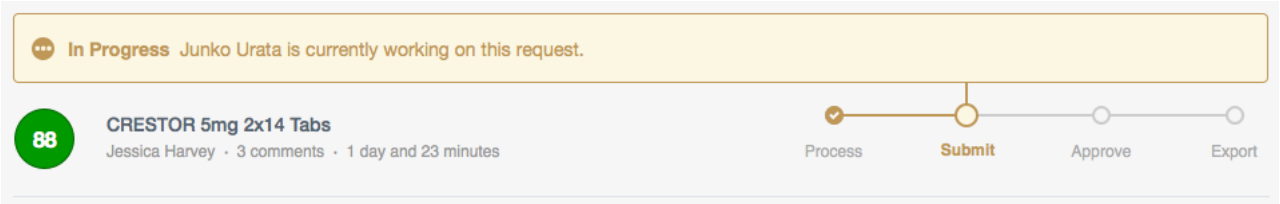
Unclaimed State

The request has processed and is not blocked. It now moved to the next step.



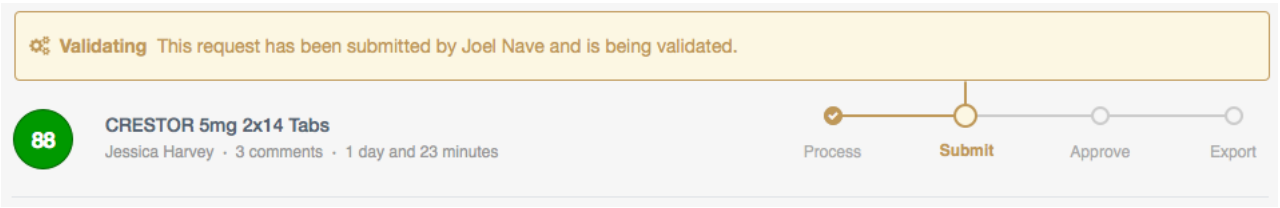
In Progress State

The request has been claimed and is now In Progress.



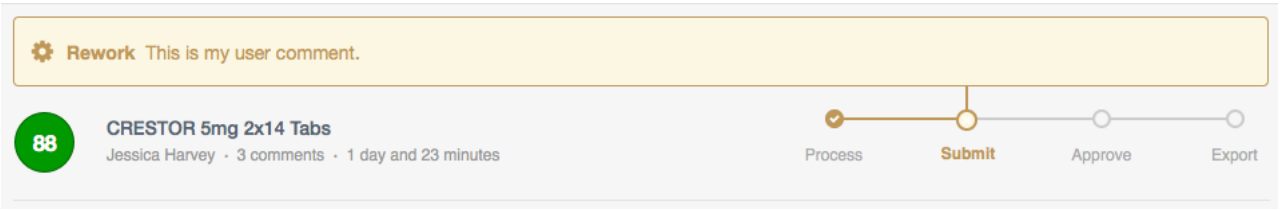
Validating State

The request has been claimed and submitted for approval. It is now in a validating state.



Rework State

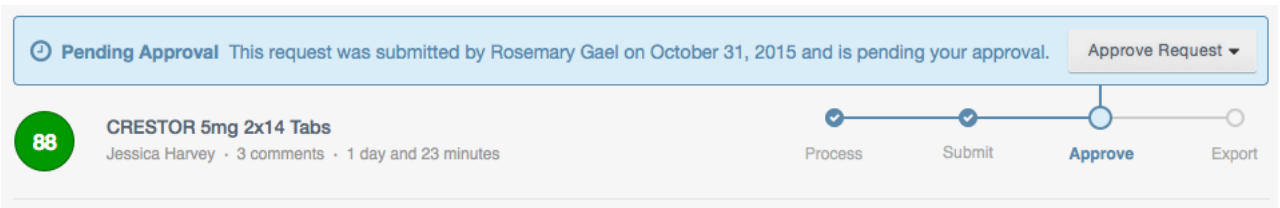
The request has been claimed and submitted for approval. The approver request a rework of this request. If a request has been sent for rework, it will no longer show as "In Progress" in the Approve step.



Approve

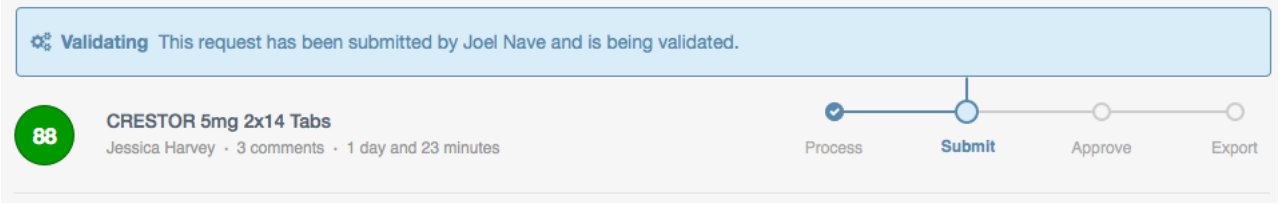
Pending Approval State

The request has processed, claimed and submitted for approval.



Validating State

The request has been claimed and submitted for approval. It is now in a validating state.

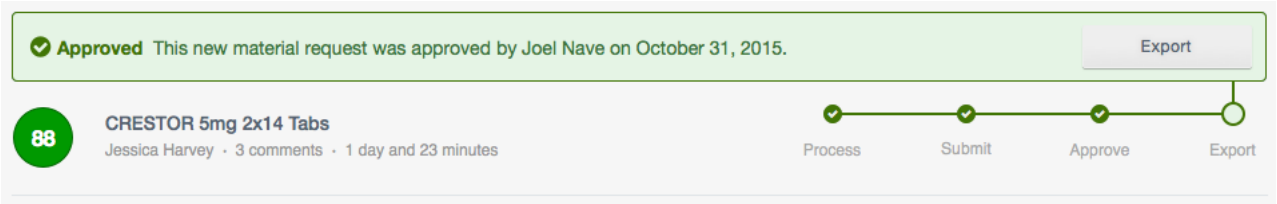


Export

Approved - Not Exported State

The request has been approved, but it has not been exported yet.

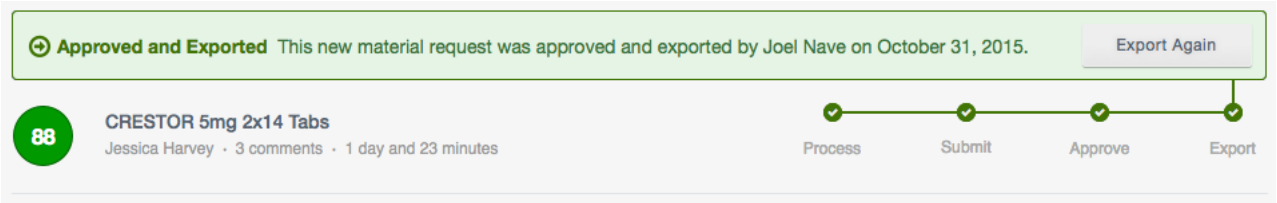
- Final step is still active because it has not yet been exported. (filled circle without checkmark)
- Button is "Export"



Approved - Exported State

The request has been approved and exported.

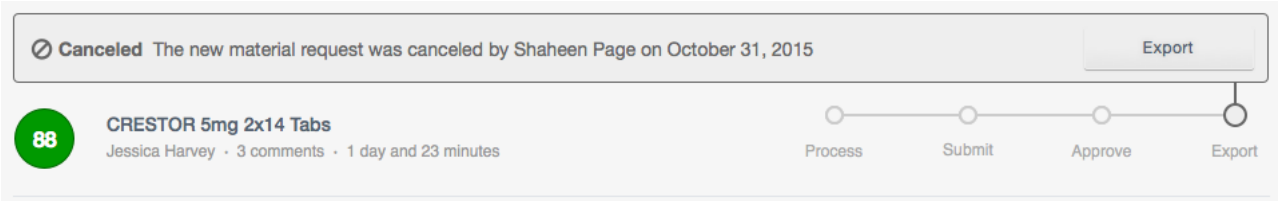
- Final step is completed so the step visual looks completed. (filled circle with checkmark).
- Button is "Export Again"



Canceled Not Exported State

The request has been canceled but not exported. A request can be canceled at any step in the workflow. The example below occurs in the last step.

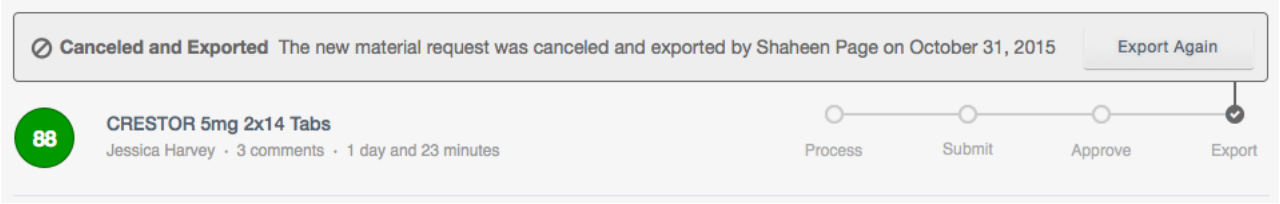
- Canceled state is gray.
- The current step is not completed so it is not the completed step visual.
- Any steps that were completed prior to the request being canceled will continue to show as completed.
- The entire workflow gets softened back to the inactive state appearance except the current step it is in.



Canceled Exported State

The request has been canceled and exported. A request can be canceled at any step in the workflow. The example below occurs in the last step.

- Canceled state is gray.
- The current step has been completed so it gets the completed step visual treatment.
- Message changes to include note that it has been exported.
- Button changes to "Export Again".
- The entire workflow gets softened back to the inactive state appearance except the current step it is in.



Reporting

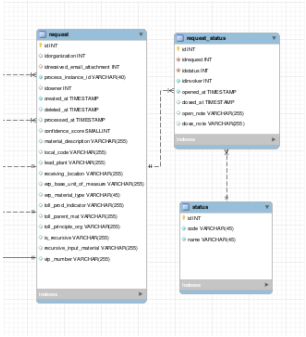
ⓘ This is for information purposes only for the technical design. Implementation of the analysis will be done as part of a separate, future story.

We will need to conduct analysis on each state and that should be thought of during the technical design. Things we will likely want to report on...

- How many requests are in each state.
- How many requests entered each state historically (e.g. how many requests were blocked, how many requests were sent back to rework, how many requests were canceled).
- What is the cycle time for each state (e.g. what is the min, max, average time a request is in the In Progress status)
- What is the overall cycle time for the entire workflow (min, max, average by time period (e.g. this week, last month, etc...)

Technical Design

States implementation in DB is done with M:N relationship:



Status table contains list of all possible statuses which are described above in experience design section, each status has its own unique id. Freshly imported requests are saved in **request** table and in application there is created new status for them which is stored in **request_status** association table. Status is joined with new request (*request.status.idrequest = request.id*) and *request_status.idstatus* is equal to 1 (*processing*). Attribute id_invoker is set to zero (meaning that author of this status change is system), *opened_at* is automatically set to NOW() and closed_at is NULL (which indicates that this status is active). When the status is changed, *closed_at* of last state is set to actual time and new row in **request_status** table is created (again with *closed_at* set to NULL). When creating new status for request, user can also add *opening_note* (for example some reason why the request is blocked, cancelled.. etc). Field closing_note is not used and can be removed in future.

Table request_status can be also used to reconstruct whole path through all statuses from processing state to exported state and

Few example SQL queries:

- **Select request with its status:**
  - SELECT r.id, s.idstatus, sc.name as status_name, s.open_note as status_note, s.opened_at as status_opened_at, su.id as status_invoker, su.name as status_invoker_name F
- **Select time taken to complete request (time between processing and approved state)**
  - SELECT stop - start as cycle_time, stop, start FROM (SELECT opened_at as start FROM request_status WHERE idrequest = 10 AND idstatus = 1 ORDER BY id ASC) f JOIN

List of state transitions:

- **Processing -> Blocked**: when TD returns processing error
- **Blocked -> Processing**: when user moves request to processing state (we will also assign request to that user)
- **Processing -> Unclaimed**: when TD returns successful processing result
- **Unclaimed -> In progress**: when user claims request
- **Processing -> In progress**: when processing ends and request was previously in "in progress" state (it has to be assigned to user)
- **Processing -> Rework**: when processing ends and request was previously in "rework" state
- **In progress/Rework -> Validating**: when user clicks on the validate button
- **Validating -> In progress/Rework**: when validation ends request will return to its previous state
- **In Progress -> Pending Approval**: when user moves request to pending approval
- **Pending Approval -> Approved/Rework**: When MDA moves request to approved or rework state
- **Approved -> Exported**: When MDA exports request

Implementation Notes

Every status has its unique ID:

Id	Code	Name
1	processing	Processing
2	blocked	Blocked
3	unclaimed	Unclaimed
4	in_progress	In Progress
5	pending_approval	Pending Approval
6	rework	Rework
7	approved	Approved
8	cancelled	Cancelled
9	exported	Exported
10	validating	Validating

Transition between states is done in file *models/requestStatus.js* and possible states request can transfer to is done using array list of possible states:

```
var statusTransitionMap = {

    1: [statsCodes.blocked, statsCodes.unclaimed, statsCodes.in_progress, statsCodes.pending_approval, statsCodes.rework],
    2: [statsCodes.processing, statsCodes.cancelled, statsCodes.validating],
    3: [statsCodes.in_progress],
    4: [statsCodes.pending_approval, statsCodes.cancelled, statsCodes.processing, statsCodes.validating],
    5: [statsCodes.rework, statsCodes.cancelled, statsCodes.approved, statsCodes.processing, statsCodes.validating],
    6: [statsCodes.pending_approval, statsCodes.cancelled, statsCodes.processing, statsCodes.validating],
    7: [statsCodes.exported],
    8: [statsCodes.exported],
    9: [],
    10: [statsCodes.in_progress, statsCodes.rework, statsCodes.pending_approval, statsCodes.blocked]

};
```

Resources

Impact Checklist

(Check to indicate that **there is an impact.** Based on the product being developed, addd to this checklist any additional impact)

Local Caching
Distributed Caching
Version Conflict
Run Time Support
Encryption
Backward Compatibility
Assess impact to authentication
Upgrade & impact to existing account
Duplicates?
Impact to retention policies
Assess impact to performance testing framework
Assess need for user interface
Assess impact to data warehouse
Do min / max validations apply
Assess impact to reports
Are documentation changes required
Does this impact audit logs
Does it impact disaster recovery

Open Tasks or Known Issues

Like Be the first to like this

No labels

4 Comments

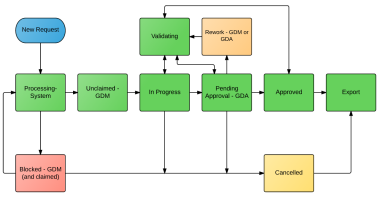
Jan Juna

@Kanda Kaliappan [Administrator] @Michele Yoshikawa
I wrote down some information about new workflow which needs to be reviewed..
Please look at that and let us know if this schema is right ok.
Initial status for freshly imported requests is processing and then:

1. If TD returns error as processing result, request will remain in processing state
2. If TD processing was successful but there are any blocker issues on this request, request will move into blocked state
3. If TD processing was successful and there are no blocker issues request will move to state from which it came (in selection it will skip processing validating and blocked statuses)
    a. If there is no possible state (not processing, blocked or validating) which we can use, and request has assigned owner it will be moved to in_progress
    b. If there is no possible state (not processing, blocked or validating) which we can use, and request has NOT assigned owner it will be moved to unclaimed
4. From unclaimed user can claim and move request to in_progress
5. From in progress user can validate request and it will be moved back after validation
6. From in progress user can move request to pending approval but it will first go to validating
    a. If validation was successful it will move request to pending approval
    b. If validation was unsuccessful it will move request back to in progress
    c. Here user can force change status to pending approval
7. From pending approval user can move request to approved and then exported
8. From pending approval user can move request to rework which acts same as in_progress

Also there are bound handlers:
- After processing or validating API will send an email notify to owner of this request if there is any
- If user moves request from blocked to processing, API will assign this request to him.

For this description I was using flow diagram which has Chris sent me earlier - see attachments.



Michele Yoshikawa

@Kanda Kaliappan [Administrator] @Jan Juna @Joel Nave Jan, #1 is fine. If there's a TD error, it won't be something that the user can address and when the error (like connecting to the DB) is resolved, the in process will be picked up and processed. Regarding #3, I like what Joel said -

JN: We will need a specific message / action for this situation. For example “3 new errors were found during validation. Correct the errors and resubmit, or you can submit for approval without correcting the errors. <Submit for Approval>. If the user does not fix the errors and submits then we will need a confirmation message “You are submitting with identified errors. <Cancel> <Submit with Errors>

Please let me know if there are other open issues or uncertainties.

Ralph Garcia

@Joel Nave @Michele Yoshikawa - Please review when you get the chance.

Michele Yoshikawa

@Ralph Garcia I added comments in line.