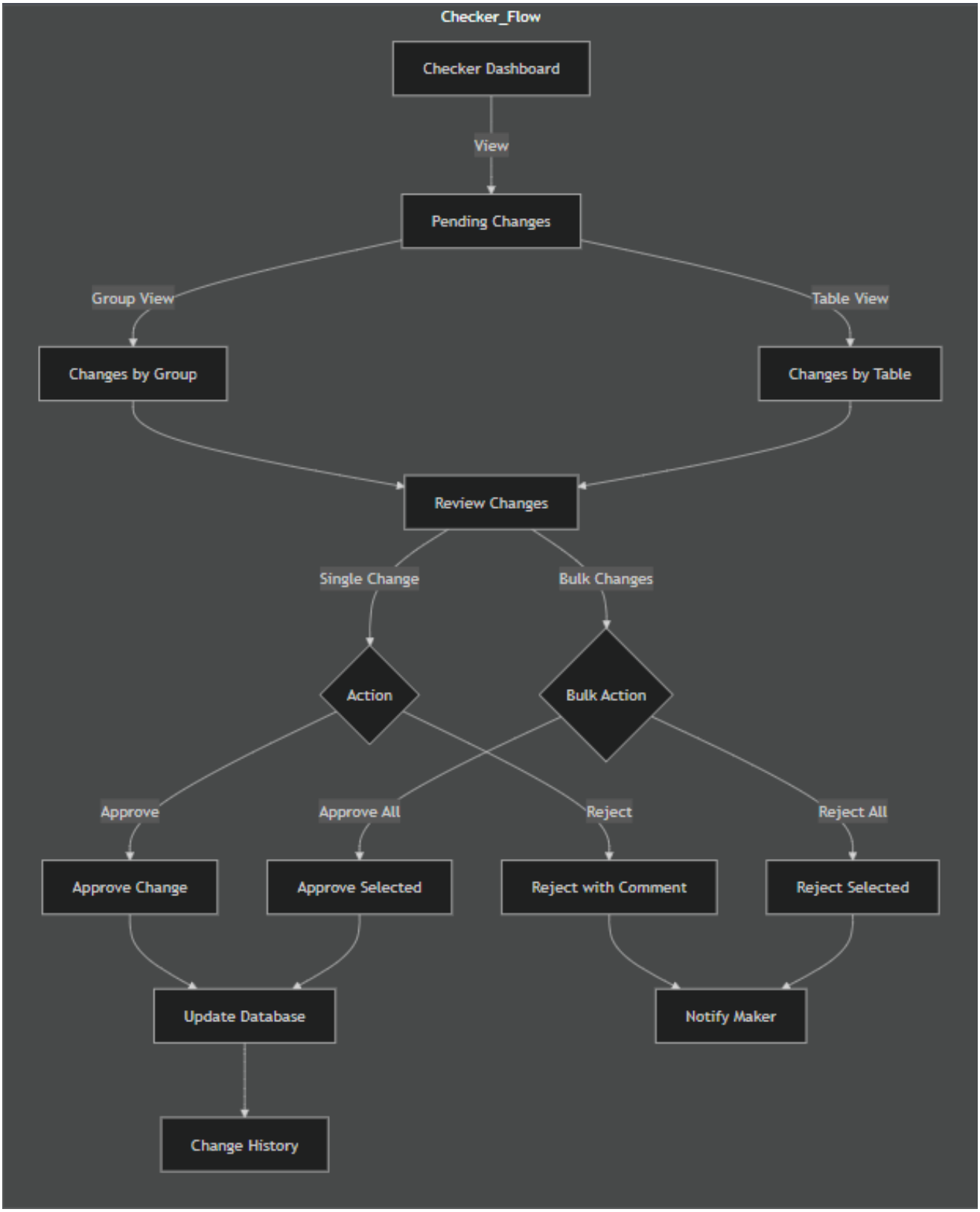
Sundaram Portal – Checker Documentation



Overview

The Checker role is a critical component of the maker-checker system, responsible for reviewing, approving, or rejecting changes made by makers. This role ensures data integrity through a four-eyes principle, where no data modification can occur without proper review and approval.

Core Features

1. Dashboard Interface

Table Organization

Grouped View

- Tables organized by admin-configured groups
- Each group displays:
    - Group name
    - Number of pending requests
    - Last update timestamp
    - List of tables with pending changes
- Groups are collapsible/expandable

Ungrouped View

- Shows tables not assigned to any group
- Displays pending request count
- Direct access to table data

Request Overview

- Total pending requests count
- Recent activity timeline
- Priority indicators for urgent requests
- Quick filters for:
  - Pending requests
  - Approved requests
  - Rejected requests

2. Change Request Management

Request Details

Each change request contains:

```
{
   "request_id": "uuid",
   "table_name": "string",
   "old_data": {
      "column1": "value1",
      "column2": "value2"
   },
   "new_data": {
      "column1": "new_value1",
      "column2": "new_value2"
   },
   "maker": "string",
   "created_at": "timestamp",
   "table_id": "string",
   "row_id": "string",
   "status": "pending"
}
```

Review Process

1. Request Access

- View pending requests via `/fetchCheckerRequest`
- Filter by:
  - Table name
  - Date range
  - Status
  - Maker

2. Data Comparison

- Side-by-side view of old and new data
- Highlighted changes
- Column-level comparison
- Original data preservation

3. Decision Making

- Approve individual requests
- Reject with mandatory comments
- Bulk approval/rejection options
- Transaction management

3. Bulk Operations

Mass Approval

```
-- Example of bulk approval process

BEGIN;
UPDATE app.change_tracker
SET status = 'approved',
    checker = :checker_id,
    updated_at = CURRENT_TIMESTAMP
WHERE request_id IN (:request_ids)
AND status = 'pending';
COMMIT;
```

Mass Rejection

- Requires comments for each rejection
- Batch processing capability
- Error handling for partial failures
- Transaction rollback on failure

4. Notification System

Real-time Alerts

- New request notifications
- Urgent request indicators
- System updates
- Maker responses

Notification Management

```
-- Notification tracking
UPDATE app.change_tracker
SET checkerseen = true
WHERE request_id = :request_id
AND checker = :checker_id;
```

5. History and Audit Trail

Change History

- Complete record of all actions
- Filterable by:
  - Date range
  - Table
  - Action type
  - Status

Audit Log Structure

```
{
   "action_id": "uuid",
   "request_id": "uuid",
   "action_type": "approve/reject",
   "checker_id": "uuid",
   "timestamp": "timestamp",
   "comments": "string",
```

```
  "table_name": "string",
  "affected_rows": "integer"
}
```

6. Security and Access Control

Session Management

- Single device login enforcement
- Session timeout handling
- Secure token management

Access Restrictions

- Read-only access to original data
- Limited to assigned tables/groups
- Action logging for all operations

7. Error Handling

Common Scenarios

- Concurrent modification conflicts
- Network failures
- Invalid data formats
- Transaction rollbacks

Recovery Procedures

- Automatic retry mechanisms
- Manual intervention options
- Error logging and reporting

8. Performance Considerations

Optimization Features

- Pagination for large datasets
- Caching of frequently accessed data
- Batch processing for bulk operations
- Indexed queries for quick retrieval

API Endpoints

Request Management

GET /api/checker/requests

GET /api/checker/groups

POST /api/checker/approve

POST /api/checker/reject

POST /api/checker/bulk-approve

POST /api/checker/bulk-reject

GET /api/checker/history

Notification Management

GET /api/checker/notifications

PUT /api/checker/notifications/seen

GET /api/checker/notifications/count

Best Practices

1. Review Process

- Always verify changes thoroughly
- Document reasons for rejections
- Use bulk operations carefully
- Maintain audit trail integrity

2. Security

- Log out after sessions
- Report suspicious activities
- Follow access control guidelines
- Maintain data confidentiality

3. Performance

- Use appropriate filters
- Leverage bulk operations
- Monitor system resources
- Report performance issues