# **Report System Documentation**

## **System Overview**

The Report System is a microservice-based application deployed in OpenShift that handles HTML report generation and storage. It interfaces with MagicDraw clients and provides flexible storage options including MongoDB and GitHub.

## **Components**

### **1. Client Layer**

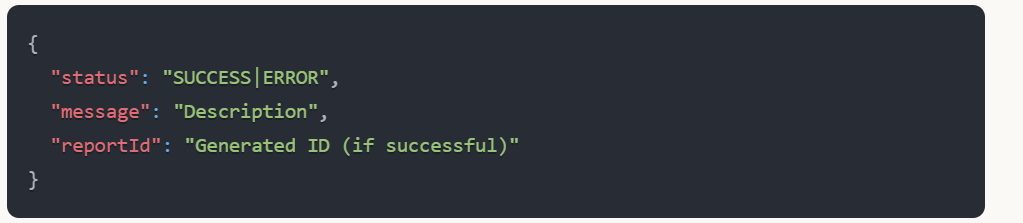
**MagicDraw Client**

* Initiates report generation requests
* Handles user interactions and error displays
* Manages retry logic for failed requests
* Displays success/error messages to users

### **2. API Layer (OpenShift)**

**Report API Endpoint: /report/Reports**

* Method: POST
* Deployment: OpenShift Container Platform
* Input Parameters:
  + htmlContent (String): HTML payload containing report content
  + type (String): Report type identifier
* Response Format: JSON



### **3. Service Layer**

**Validation Service**

* Performs input validation:
  + HTML content null/blank check
  + Report type validation
  + Content format verification
* Returns detailed validation errors

**Report Service**

* Orchestrates the report generation process
* Manages storage decisions
* Handles error scenarios
* Coordinates with ID generator

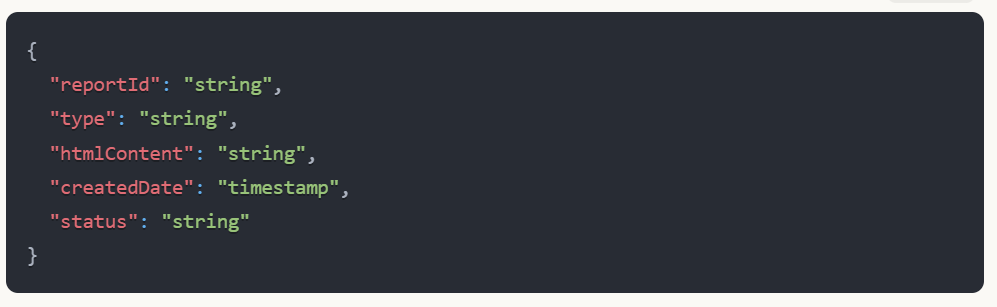
**ID Generator**

* Generates auto-incremental IDs
* Ensures uniqueness across the system
* Format: RPT-YYYYMMDD-NNNN

### **4. Storage Layer**

**Primary Storage: MongoDB**

* Stores all report data
* Schema:



**Secondary Storage: GitHub (Optional)**

* Configured through environment variables
* Stores HTML files in structured repositories
* Path format: /<report-type>/<YYYY>/<MM>/<reportId>.html

## **Process Flow**

### **1. Request Initiation**

1. MagicDraw client prepares HTML content and report type
2. Sends POST request to /report/Reports

### **2. Validation Phase**

1. API receives request and forwards to Validation Service
2. Validates:
   * HTML content presence
   * Report type validity
   * Content format
3. Returns validation status

### **3. Error Handling**

1. If validation fails:
   * Returns detailed error message
   * MagicDraw displays error
   * Allows user to retry
2. Common error scenarios:
   * Empty content
   * Invalid report type
   * Malformed HTML
   * Service unavailability

### **4. Success Flow**

1. Generates unique report ID
2. Stores data in MongoDB
3. If configured, stores in GitHub
4. Returns success response with report ID

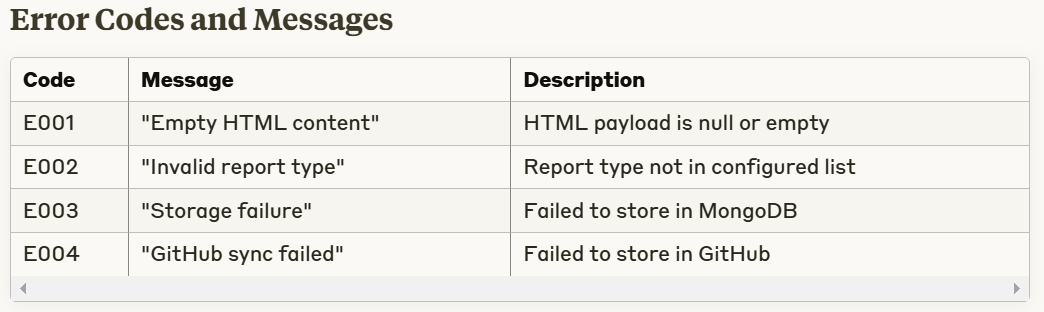
### **5. Storage Process**

1. MongoDB Storage (Required):
   * Stores complete report data
   * Maintains audit trail
   * Handles versioning
2. GitHub Storage (Optional):
   * Stores HTML files
   * Maintains file history
   * Enables version control

## **Configuration Parameters**

### **OpenShift Environment Variables**





## **Retry Mechanism**

* MagicDraw implements exponential backoff
* Maximum 3 retry attempts
* User notification after final retry
* Retry only for specific error codes (E003, E004)

## **Security Considerations**

1. OpenShift Security:
   * Pod Security Policies
   * Network Policies
   * Resource Quotas
2. Data Security:
   * MongoDB Authentication
   * GitHub Token Security
   * Data Encryption at Rest
   * TLS for Data in Transit

Diagram

