**API Documentation for Template-Based Release Emailer**

**1. HomePage Module**

* **Description**: Provides an introduction to the system.
* **Endpoint**: GET /api/home
* **Request Parameters**: None
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Welcome to the Template-Based Release Emailer"
* }
* }

**2. User Management Module**

**2.1 Get User List**

* **Description**: Retrieves a list of all users in the system.
* **Endpoint**: GET /api/users
* **Request Parameters**: None
* **Response Example**:
* [
* {
* "soeid": "zz57529",
* "name": "Zhang, Zhaofei Betty",
* "email": "zz57529@citi.com",
* "role": "admin"
* },
* {
* "soeid": "5g52032",
* "name": "Gu, Shuang",
* "email": "sg52032@citi.com",
* "role": "user"
* }
* ]

**2.2 Create New User**

* **Description**: Creates a new user (admin-only access).
* **Endpoint**: POST /api/users
* **Request Example**:
* {
* "soeid": "tc57089",
* "name": "Chen, Tianqi",
* "email": "tc57089@citi.com",
* "role": "user"
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "User created successfully"
* }
* }

**3. Template Management Module**

This module uses a unified approach for template creation, updating, and retrieval. The templateType field indicates whether the template is a standard HTML template ("html") or created using a drag-and-drop interface ("dragdrop"). If using drag-and-drop, additional layout data can be included in an optional layout field.

**3.1 Create Template**

* **Endpoint**: POST /api/templates
* **Request Example**:
* {
* "templateName": "Release Announcement",
* "templateContent": "<html><body>Release content goes here.</body></html>",
* "templateType": "html", // "html" for standard or "dragdrop" for drag-and-drop generated templates
* "layout": {
* "components": [
* {
* "id": "comp1",
* "type": "text",
* "position": { "x": 10, "y": 20 },
* "settings": { "fontSize": 14 }
* },
* {
* "id": "comp2",
* "type": "image",
* "position": { "x": 150, "y": 20 },
* "settings": { "width": 200 }
* }
* ]
* }
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Template created successfully"
* }
* }

**3.2 Update Template**

* **Endpoint**: PUT /api/templates/{templateId}
* **Request Example**:
* {
* "templateName": "Updated Release Template",
* "templateContent": "<html><body>Updated release content.</body></html>",
* "templateType": "html", // "html" or "dragdrop"
* "layout": { /\* Optional layout data for drag-and-drop templates \*/ }
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Template updated successfully"
* }
* }

**3.3 Get Template Content**

* **Endpoint**: GET /api/templates/{templateId}
* **Response Example**:
* {
* "status": "success",
* "data": {
* "templateName": "Updated Release Template",
* "templateContent": "<html><body>Retrieve release content.</body></html>",
* "templateType": "html",
* "layout": { /\* If applicable, the layout data for drag-and-drop templates \*/ }
* }
* }

**3.4 Delete Template**

* **Endpoint**: DELETE /api/templates/{templateId}
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Template deleted successfully"
* }
* }

**4. ROD Integration Module**

This module handles integration with the ROD system for retrieving release data. It allows querying release information with optional filtering parameters.

**4.1 Get ROD Releases**

* **Description**: Retrieves release information from the ROD system based on provided filters.
* **Endpoint**: GET /api/rod/releases
* **Request Parameters** (as query parameters):
  + startDate: Start date in ISO8601 format (e.g., "2025-02-01T00:00:00").
  + endDate: End date in ISO8601 format (e.g., "2025-02-28T23:59:59").
  + releaseType: (Optional) Type of release (e.g., "software").
* **Request Example**:
* GET /api/rod/releases?startDate=2025-02-01T00:00:00&endDate=2025-02-28T23:59:59&releaseType=software
* **Response Example**:
* {
* "status": "success",
* "data": [
* {
* "releaseId": "r123",
* "title": "Release v1.2",
* "description": "Details about release v1.2",
* "releaseDate": "2025-02-20T10:00:00"
* },
* {
* "releaseId": "r124",
* "title": "Release v1.3",
* "description": "Details about release v1.3",
* "releaseDate": "2025-02-25T10:00:00"
* }
* ]
* }

**5. Email Management Module**

This module supports draft creation, preview, sending, scheduling, and deletion of emails.

**5.1 Get Sent Email List**

* **Description**: Retrieves a list of sent emails for a user (with optional filtering).
* **Endpoint**: GET /api/emails/sent
* **Request Example**:
* {
* "userSoeid": "zz57529",
* "startDate": "2025-02-19T12:34:56",
* "endDate": "2025-02-25T10:01:01"
* }
* **Response Example**:
* [
* {
* "emailId": "2",
* "userSoeid": "zz57529",
* "title": "New Release",
* "recipient": "user@example.com",
* "templateId": "1",
* "sendTime": "2025-02-20T10:00:00",
* "status": "success"
* },
* {
* "emailId": "3",
* "userSoeid": "zz57529",
* "title": "Release Update",
* "recipient": "user@example.com",
* "templateId": "2",
* "sendTime": "2025-02-24T10:00:00",
* "status": "success"
* }
* ]

**5.2 Create Email Draft**

* **Endpoint**: POST /api/emails
* **Request Example**:
* {
* "userSoeid": "zz57529",
* "title": "New Release Information",
* "recipient": "user@example.com",
* "templateId": "1",
* "sendTime": "2025-02-24T12:30:00",
* "content": "<html>...</html>"
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Email created successfully as draft",
* "emailId": "1"
* }
* }

**5.3 Save Email Draft**

* **Endpoint**: POST /api/emails/{emailId}/save
* **Request Example**:
* {
* "userSoeid": "zz57529",
* "draftStatus": true
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Email saved as draft successfully"
* }
* }

**5.4 Update Email Draft**

* **Endpoint**: PUT /api/emails/{emailId}
* **Request Example**:
* {
* "userSoeid": "zz57529",
* "title": "Updated Release Information",
* "recipient": "user@example.com",
* "templateId": "1",
* "sendTime": "2025-02-24T12:30:00",
* "content": "<html>...</html>"
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Email draft updated successfully"
* }
* }

**5.5 Preview Email Draft**

* **Endpoint**: GET /api/emails/{emailId}/preview
* **Response Example**:
* {
* "status": "success",
* "data": {
* "emailId": "3",
* "userSoeid": "zz57529",
* "title": "New Release",
* "recipient": "user@example.com",
* "templateId": "1",
* "sendTime": "2025-02-20T10:00:00",
* "content": "<html>...</html>"
* }
* }

**5.6 Publish (Send) Email**

* **Endpoint**: POST /api/emails/{emailId}/publish
* **Request Example**:
* {
* "userSoeid": "zz57529"
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Email sent successfully"
* }
* }

**5.7 Schedule Email**

* **Endpoint**: POST /api/emails/{emailId}/schedule
* **Request Example**:
* {
* "userSoeid": "zz57529",
* "sendTime": "2025-02-24T12:30:00"
* }
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Email scheduled successfully"
* }
* }

**5.8 Delete Email**

* **Endpoint**: DELETE /api/emails/{emailId}
* **Response Example**:
* {
* "status": "success",
* "data": {
* "message": "Email deleted successfully"
* }
* }

**6. Security and Error Handling**

* **Authorization**: All endpoints require an authorization token in the header:
* Authorization: Bearer {token}
* **Error Format**: On error (e.g., authentication failure, bad request), the API returns:
* {
* "status": "error",
* "errorCode": "SPECIFIC\_ERROR\_CODE",
* "message": "Detailed error description"
* }

**7. Additional Notes**

* **Unified Template Management**: By using the templateType field, both standard HTML and drag-and-drop templates are managed through the same endpoints. Drag-and-drop templates can include additional layout metadata, while HTML templates simply ignore this field.
* **ROD Integration**: The ROD Integration Module is designed to be robust, with caching and error handling on the backend to manage high traffic or delays from the ROD system.
* **Time Format**: All dates and times use the ISO8601 standard format.

You can now save this content into a Word document for further editing or distribution.