**API Endpoints Documentation**

**A document covering all API endpoints (with what is sent and received) is mandatory.**

**Table of Contents**

1. [**Authentication Endpoints**](#authentication-endpoints)
2. [**Email Management Endpoints**](#email-management-endpoints)
3. [**Role Management Endpoints**](#role-management-endpoints)
4. [**User Management Endpoints**](#user-management-endpoints)
5. [**Company Management Endpoints**](#company-management-endpoints)
6. [**GDPR Request Management Endpoints**](#gdpr-request-management-endpoints)

**Authentication Endpoints**

**Base URL: /api/auth**

**1. Register User**

* **Endpoint: POST /api/auth/register**
* **Description: Create a new user account**
* **Authentication: None required**
* **Request Body:**

**{**

**"firstname": "string",**

**"lastname": "string",**

**"email": "string",**

**"password": "string",**

**"roleId": "integer"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "User registered successfully!"**
* **}**
  + **Error (400):**
* **{**
* **"message": "Error: Email is already in use!"**
* **}**

**2. Login User**

* **Endpoint: POST /api/auth/login**
* **Description: Authenticate user and return JWT token**
* **Authentication: None required**
* **Request Body:**

**{**

**"email": "string",**

**"password": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"token": "string",**
* **"type": "Bearer",**
* **"id": "integer",**
* **"email": "string",**
* **"firstname": "string",**
* **"lastname": "string",**
* **"role": "string"**
* **}**
  + **Error (401):**
* **{**
* **"message": "Error: Invalid email or password!"**
* **}**

**3. Refresh Token**

* **Endpoint: POST /api/auth/refresh**
* **Description: Refresh JWT token**
* **Authentication: None required**
* **Request Body:**

**{**

**"token": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"token": "string",**
* **"type": "Bearer"**
* **}**
  + **Error (401):**
* **{**
* **"message": "Invalid refresh token!"**
* **}**

**4. Logout User**

* **Endpoint: POST /api/auth/logout**
* **Description: Logout user (frontend clears token)**
* **Authentication: None required**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "User logged out successfully!"**
* **}**

**5. Validate Token**

* **Endpoint: POST /api/auth/validate**
* **Description: Validate JWT token and return user info**
* **Authentication: None required**
* **Request Body:**

**{**

**"token": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"id": "integer",**
* **"email": "string",**
* **"firstname": "string",**
* **"lastname": "string",**
* **"role": "string",**
* **"active": "boolean"**
* **}**
  + **Error (401):**
* **{**
* **"message": "Invalid token!"**
* **}**

**6. Change Password**

* **Endpoint: POST /api/auth/change-password**
* **Description: Change the current user's password**
* **Authentication: Bearer Token required**
* **Request Body:**

**{**

**"oldPassword": "string",**

**"newPassword": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Password changed successfully!"**
* **}**
  + **Error (400/401):**
* **{**
* **"message": "Error changing password: [error details]"**
* **}**

**7. Forgot Password**

* **Endpoint: POST /api/auth/forgot-password**
* **Description: Request password reset**
* **Authentication: None required**
* **Request Body:**

**{**

**"email": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Password reset instructions sent to your email!"**
* **}**

**Email Management Endpoints**

**Base URL: /api/emails**

**1. Send Test Email**

* **Endpoint: POST /api/emails/test**
* **Description: Send a test email to verify email configuration**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"email": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Test email sent successfully to [email]",**
* **"status": "success"**
* **}**

**2. Send Simple Email**

* **Endpoint: POST /api/emails/simple**
* **Description: Send a simple text email**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"to": "string",**

**"subject": "string",**

**"text": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Simple email sent successfully",**
* **"status": "success"**
* **}**

**3. Send Custom Email**

* **Endpoint: POST /api/emails/custom**
* **Description: Send a custom email using template**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"to": "string",**

**"subject": "string",**

**"templateName": "string",**

**"variables": {**

**"key1": "value1",**

**"key2": "value2"**

**}**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Custom email sent successfully",**
* **"status": "success"**
* **}**

**4. Send Bulk Email**

* **Endpoint: POST /api/emails/bulk**
* **Description: Send bulk emails to multiple recipients**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"recipients": ["string", "string"],**

**"subject": "string",**

**"templateName": "string",**

**"variables": {**

**"key1": "value1",**

**"key2": "value2"**

**}**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Bulk emails sent successfully to [count] recipients",**
* **"status": "success"**
* **}**

**5. Send Admin Notification**

* **Endpoint: POST /api/emails/admin-notification**
* **Description: Send notification to all administrators**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"subject": "string",**

**"message": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Admin notification sent successfully",**
* **"status": "success"**
* **}**

**6. Get Email Statistics**

* **Endpoint: GET /api/emails/statistics**
* **Description: Get email sending statistics**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"totalEmailsSent": "long",**
* **"totalEmailsFailed": "long",**
* **"totalEmailsToday": "long"**
* **}**

**7. Resend Welcome Email**

* **Endpoint: POST /api/emails/resend-welcome/{userId}**
* **Description: Resend welcome email to specific user**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: userId (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Welcome email resent successfully",**
* **"status": "success"**
* **}**

**Role Management Endpoints**

**Base URL: /api/roles**

**1. Get All Roles**

* **Endpoint: GET /api/roles**
* **Description: Get all roles**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **[**
* **{**
* **"idRole": "integer",**
* **"role": "string"**
* **}**
* **]**

**2. Get Role by ID**

* **Endpoint: GET /api/roles/{id}**
* **Description: Get role by ID**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"idRole": "integer",**
* **"role": "string"**
* **}**
  + **Not Found (404): Empty response**

**3. Get Role by Name**

* **Endpoint: GET /api/roles/name/{roleName}**
* **Description: Get role by name**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: roleName (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"idRole": "integer",**
* **"role": "string"**
* **}**

**4. Create Role**

* **Endpoint: POST /api/roles**
* **Description: Create new role**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"role": "string"**

**}**

* **Response:** 
  + **Success (201):**
* **{**
* **"idRole": "integer",**
* **"role": "string"**
* **}**
  + **Error (400):**
* **{**
* **"message": "Error: Role name already exists!",**
* **"status": "error"**
* **}**

**5. Update Role**

* **Endpoint: PUT /api/roles/{id}**
* **Description: Update role**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body:**

**{**

**"role": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **{**
* **"idRole": "integer",**
* **"role": "string"**
* **}**

**6. Delete Role**

* **Endpoint: DELETE /api/roles/{id}**
* **Description: Delete role**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Role deleted successfully",**
* **"status": "success"**
* **}**
  + **Error (400):**
* **{**
* **"message": "Cannot delete role: It has associated users",**
* **"status": "error"**
* **}**

**7. Get Users Count by Role**

* **Endpoint: GET /api/roles/{id}/users/count**
* **Description: Get users count for a specific role**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **5**

**8. Get Role Statistics**

* **Endpoint: GET /api/roles/statistics**
* **Description: Get role statistics**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"totalRoles": "long",**
* **"rolesWithUsers": "long",**
* **"emptyRoles": "long"**
* **}**

**9. Get All Role Names**

* **Endpoint: GET /api/roles/names**
* **Description: Get all role names (for dropdowns)**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **["ADMIN", "CLIENT", "GERANT"]**

**10. Create Default Roles**

* **Endpoint: POST /api/roles/init-defaults**
* **Description: Create default roles**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Default roles created successfully",**
* **"status": "success"**
* **}**

**11. Validate Role Name**

* **Endpoint: GET /api/roles/validate/{roleName}**
* **Description: Check if role name is valid**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: roleName (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **true**

**User Management Endpoints**

**Base URL: /api/users**

**1. Get All Users**

* **Endpoint: GET /api/users**
* **Description: Get all users**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **[**
* **{**
* **"idUser": "integer",**
* **"firstname": "string",**
* **"lastname": "string",**
* **"email": "string",**
* **"active": "boolean",**
* **"role": {**
* **"idRole": "integer",**
* **"role": "string"**
* **}**
* **}**
* **]**

**2. Get User by ID**

* **Endpoint: GET /api/users/{id}**
* **Description: Get user by ID**
* **Authentication: Bearer Token required (ADMIN role or current user)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"idUser": "integer",**
* **"firstname": "string",**
* **"lastname": "string",**
* **"email": "string",**
* **"active": "boolean",**
* **"role": {**
* **"idRole": "integer",**
* **"role": "string"**
* **}**
* **}**

**3. Get User by Email**

* **Endpoint: GET /api/users/email/{email}**
* **Description: Get user by email**
* **Authentication: Bearer Token required (ADMIN role or authenticated user's own email)**
* **Path Parameters: email (string)**
* **Request Body: None**
* **Response: Same as Get User by ID**

**4. Create User**

* **Endpoint: POST /api/users**
* **Description: Create new user**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"firstname": "string",**

**"lastname": "string",**

**"email": "string",**

**"password": "string",**

**"role": {**

**"idRole": "integer"**

**}**

**}**

* **Response:** 
  + **Success (201):**
* **{**
* **"idUser": "integer",**
* **"firstname": "string",**
* **"lastname": "string",**
* **"email": "string",**
* **"active": "boolean",**
* **"role": {**
* **"idRole": "integer",**
* **"role": "string"**
* **}**
* **}**

**5. Update User**

* **Endpoint: PUT /api/users/{id}**
* **Description: Update user**
* **Authentication: Bearer Token required (ADMIN role or current user)**
* **Path Parameters: id (integer)**
* **Request Body: Same as Create User**
* **Response: Same as Create User**

**6. Deactivate User**

* **Endpoint: PUT /api/users/{id}/deactivate**
* **Description: Deactivate user (soft delete)**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response: Same as Get User by ID**

**7. Activate User**

* **Endpoint: PUT /api/users/{id}/activate**
* **Description: Activate user**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response: Same as Get User by ID**

**8. Delete User**

* **Endpoint: DELETE /api/users/{id}**
* **Description: Delete user permanently**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **"User deleted successfully"**

**9. Get Users by Role**

* **Endpoint: GET /api/users/role/{roleId}**
* **Description: Get users by role**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: roleId (integer)**
* **Request Body: None**
* **Response: Same as Get All Users**

**10. Get Active Users**

* **Endpoint: GET /api/users/active**
* **Description: Get active users only**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response: Same as Get All Users**

**11. Change User Password**

* **Endpoint: PUT /api/users/{id}/password**
* **Description: Change user password**
* **Authentication: Bearer Token required (ADMIN role or current user)**
* **Path Parameters: id (integer)**
* **Request Body:**

**{**

**"oldPassword": "string",**

**"newPassword": "string"**

**}**

* **Response:** 
  + **Success (200):**
* **"Password changed successfully"**

**Company Management Endpoints**

**Base URL: /api/companies**

**1. Get All Companies**

* **Endpoint: GET /api/companies**
* **Description: Retrieve all companies**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **[**
* **{**
* **"idCompany": "integer",**
* **"companyName": "string",**
* **"email": "string"**
* **}**
* **]**

**2. Get Company by ID**

* **Endpoint: GET /api/companies/{id}**
* **Description: Retrieve a company by its ID**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"idCompany": "integer",**
* **"companyName": "string",**
* **"email": "string"**
* **}**
  + **Not Found (404): Empty response**

**3. Get Company by Email**

* **Endpoint: GET /api/companies/email/{email}**
* **Description: Retrieve a company by its email**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: email (string)**
* **Request Body: None**
* **Response: Same as Get Company by ID**

**4. Get Company by Name**

* **Endpoint: GET /api/companies/name/{companyName}**
* **Description: Retrieve a company by its name**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: companyName (string)**
* **Request Body: None**
* **Response: Same as Get Company by ID**

**5. Create Company**

* **Endpoint: POST /api/companies**
* **Description: Create a new company**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body:**

**{**

**"companyName": "string",**

**"email": "string"**

**}**

* **Response:** 
  + **Success (201):**
* **{**
* **"idCompany": "integer",**
* **"companyName": "string",**
* **"email": "string"**
* **}**
  + **Error (400):**
* **{**
* **"message": "Error: Company with this email already exists!",**
* **"status": "error"**
* **}**

**6. Update Company**

* **Endpoint: PUT /api/companies/{id}**
* **Description: Update an existing company**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: Same as Create Company**
* **Response: Same as Create Company**

**7. Delete Company**

* **Endpoint: DELETE /api/companies/{id}**
* **Description: Delete a company**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Company deleted successfully",**
* **"status": "success"**
* **}**
  + **Error (400):**
* **{**
* **"message": "Cannot delete company: It has associated GDPR requests",**
* **"status": "error"**
* **}**

**8. Search Companies by Name**

* **Endpoint: GET /api/companies/search/name**
* **Description: Search companies by name (case-insensitive)**
* **Authentication: Bearer Token required (ADMIN role)**
* **Query Parameters: name (string)**
* **Request Body: None**
* **Response: Same as Get All Companies**

**9. Search Companies by Email**

* **Endpoint: GET /api/companies/search/email**
* **Description: Search companies by email (case-insensitive)**
* **Authentication: Bearer Token required (ADMIN role)**
* **Query Parameters: email (string)**
* **Request Body: None**
* **Response: Same as Get All Companies**

**10. Get Companies with Pagination**

* **Endpoint: GET /api/companies/paginated**
* **Description: Get companies with pagination support**
* **Authentication: Bearer Token required (ADMIN role)**
* **Query Parameters:** 
  + **page (integer, default: 0)**
  + **size (integer, default: 10)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"content": [**
* **{**
* **"idCompany": "integer",**
* **"companyName": "string",**
* **"email": "string"**
* **}**
* **],**
* **"totalElements": "long",**
* **"totalPages": "integer",**
* **"size": "integer",**
* **"number": "integer"**
* **}**

**11. Get Companies Count**

* **Endpoint: GET /api/companies/count**
* **Description: Get total number of companies**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **10**

**12. Get Company Statistics**

* **Endpoint: GET /api/companies/statistics**
* **Description: Get company statistics**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"totalCompanies": "long"**
* **}**

**13. Get All Company Names**

* **Endpoint: GET /api/companies/names**
* **Description: Get all company names for dropdowns**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **["Company A", "Company B", "Company C"]**

**14. Get All Company Emails**

* **Endpoint: GET /api/companies/emails**
* **Description: Get all company emails for dropdowns**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **["companya@example.com", "companyb@example.com"]**

**15. Validate Company Email**

* **Endpoint: GET /api/companies/validate/email/{email}**
* **Description: Check if company email format is valid**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: email (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **true**

**16. Validate Company Name**

* **Endpoint: GET /api/companies/validate/name/{companyName}**
* **Description: Check if company name is valid**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: companyName (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **true**

**17. Check if Company Exists by Email**

* **Endpoint: GET /api/companies/exists/email/{email}**
* **Description: Check if a company with this email exists**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: email (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **false**

**18. Check if Company Exists by Name**

* **Endpoint: GET /api/companies/exists/name/{companyName}**
* **Description: Check if a company with this name exists**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: companyName (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **false**

**19. Create Default Companies**

* **Endpoint: POST /api/companies/init-defaults**
* **Description: Create default companies for testing**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "Created 5 default companies",**
* **"status": "success"**
* **}**

**GDPR Request Management Endpoints**

**Base URL: /api/gdpr-requests**

**1. Get All GDPR Requests**

* **Endpoint: GET /api/gdpr-requests**
* **Description: Retrieve all GDPR requests (Admin only)**
* **Authentication: Bearer Token required (ADMIN role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **[**
* **{**
* **"idRequest": "integer",**
* **"requestType": "string",**
* **"status": "string",**
* **"requestDate": "datetime",**
* **"requestContent": "string",**
* **"user": {**
* **"idUser": "integer",**
* **"firstname": "string",**
* **"lastname": "string",**
* **"email": "string"**
* **},**
* **"company": {**
* **"idCompany": "integer",**
* **"companyName": "string",**
* **"email": "string"**
* **}**
* **}**
* **]**

**2. Get GDPR Request by ID**

* **Endpoint: GET /api/gdpr-requests/{id}**
* **Description: Retrieve a GDPR request by its ID**
* **Authentication: Bearer Token required (ADMIN role or request owner)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response: Same as individual request in Get All GDPR Requests**

**3. Get Current User's GDPR Requests (UC1)**

* **Endpoint: GET /api/gdpr-requests/my-requests**
* **Description: UC1 - Client can view their own GDPR requests**
* **Authentication: Bearer Token required (CLIENT or ADMIN role)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**4. Get User's GDPR Requests by User ID**

* **Endpoint: GET /api/gdpr-requests/user/{userId}**
* **Description: Get GDPR requests for a specific user (Admin only)**
* **Authentication: Bearer Token required (ADMIN role)**
* **Path Parameters: userId (integer)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**5. Get Company's GDPR Requests (UC2)**

* **Endpoint: GET /api/gdpr-requests/company/{companyId}**
* **Description: UC2 - Manager can view requests for their company**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: companyId (integer)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**6. Get Company's Pending Requests (UC2)**

* **Endpoint: GET /api/gdpr-requests/company/{companyId}/pending**
* **Description: UC2 - Manager can view pending requests for their company**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: companyId (integer)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**7. Create New GDPR Request**

* **Endpoint: POST /api/gdpr-requests**
* **Description: Create a new GDPR request**
* **Authentication: Bearer Token required (CLIENT or ADMIN role)**
* **Request Body:**

**{**

**"requestType": "string",**

**"requestContent": "string",**

**"userId": "integer",**

**"companyId": "integer"**

**}**

* **Response:** 
  + **Success (201): Same as individual request format**
  + **Error (400):**
* **{**
* **"message": "Error creating GDPR request: [error details]",**
* **"status": "error"**
* **}**

**8. Update GDPR Request Status**

* **Endpoint: PUT /api/gdpr-requests/{id}/status**
* **Description: Update the status of a GDPR request (Manager/Admin only)**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: id (integer)**
* **Request Body:**

**{**

**"status": "string"**

**}**

* **Response: Same as individual request format**

**9. Update GDPR Request Content**

* **Endpoint: PUT /api/gdpr-requests/{id}/content**
* **Description: Update the content of a GDPR request**
* **Authentication: Bearer Token required (ADMIN role or request owner)**
* **Path Parameters: id (integer)**
* **Request Body:**

**{**

**"content": "string"**

**}**

* **Response: Same as individual request format**

**10. Delete GDPR Request**

* **Endpoint: DELETE /api/gdpr-requests/{id}**
* **Description: Delete a GDPR request**
* **Authentication: Bearer Token required (ADMIN role or request owner)**
* **Path Parameters: id (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"message": "GDPR request deleted successfully",**
* **"status": "success"**
* **}**

**11. Get Requests by Status**

* **Endpoint: GET /api/gdpr-requests/status/{status}**
* **Description: Get GDPR requests filtered by status**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: status (string - "PENDING" or "PROCESSED")**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**12. Get Requests by Type**

* **Endpoint: GET /api/gdpr-requests/type/{requestType}**
* **Description: Get GDPR requests filtered by type**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: requestType (string - "MODIFICATION" or "DELETION")**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**13. Get Current User's Requests by Status**

* **Endpoint: GET /api/gdpr-requests/my-requests/status/{status}**
* **Description: Get current user's GDPR requests filtered by status**
* **Authentication: Bearer Token required (CLIENT or ADMIN role)**
* **Path Parameters: status (string)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**14. Get Requests Between Dates**

* **Endpoint: GET /api/gdpr-requests/date-range**
* **Description: Get GDPR requests within a date range**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Query Parameters:** 
  + **startDate (datetime ISO format)**
  + **endDate (datetime ISO format)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**15. Get Recent Requests**

* **Endpoint: GET /api/gdpr-requests/recent**
* **Description: Get GDPR requests from the last 30 days**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Request Body: None**
* **Response: Same as Get All GDPR Requests**

**16. Count Requests by Status**

* **Endpoint: GET /api/gdpr-requests/count/status/{status}**
* **Description: Get the count of requests for a specific status**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: status (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **15**

**17. Count Requests by Company**

* **Endpoint: GET /api/gdpr-requests/count/company/{companyId}**
* **Description: Get the count of requests for a specific company**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: companyId (integer)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **8**

**18. Get GDPR Request Statistics**

* **Endpoint: GET /api/gdpr-requests/statistics**
* **Description: Get comprehensive statistics about GDPR requests**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **{**
* **"totalRequests": "long",**
* **"pendingRequests": "long",**
* **"processedRequests": "long",**
* **"modificationRequests": "long",**
* **"deletionRequests": "long"**
* **}**

**19. Get Valid Request Types**

* **Endpoint: GET /api/gdpr-requests/valid-types**
* **Description: Get all valid GDPR request types**
* **Authentication: Bearer Token required (CLIENT, ADMIN or GERANT role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **["MODIFICATION", "DELETION"]**

**20. Get Valid Statuses**

* **Endpoint: GET /api/gdpr-requests/valid-statuses**
* **Description: Get all valid GDPR request statuses**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **["PENDING", "PROCESSED"]**

**21. Validate Request Type**

* **Endpoint: GET /api/gdpr-requests/validate/type/{requestType}**
* **Description: Check if a request type is valid**
* **Authentication: Bearer Token required (CLIENT, ADMIN or GERANT role)**
* **Path Parameters: requestType (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **true**

**22. Validate Status**

* **Endpoint: GET /api/gdpr-requests/validate/status/{status}**
* **Description: Check if a status is valid**
* **Authentication: Bearer Token required (ADMIN or GERANT role)**
* **Path Parameters: status (string)**
* **Request Body: None**
* **Response:** 
  + **Success (200):**
* **true**