Endpoint Authorization Matrix - Complete API Reference

Date: 2025-11-14 **Version:** 3.2 **Total Endpoints:** 16 Files, 113 Endpoints

---

Authorization Levels

Level	Description	Implementation
Authenticated	Any logged-in user	`.RequireAuthorization()`
HasAccess	Users with system access (Reader, Publisher, SuperUser)	`.RequireAuthorization("HasAccess")`
SuperUser	Administrative users only	`.RequireAuthorization("SuperUser")`

Note: All endpoints require Windows Authentication (or TestAuth in dev). “access” means the role can call the endpoint; “-” means no access.

---

1. Document Endpoints (`/api/documents`)

Base Authorization: `RequireAuthorization()` (All authenticated users)

HTTP Method	Endpoint	Description	Reader	Publisher
GET	`/api/documents`	Get all documents	access	access
GET	`/api/documents/{id}`	Get document by ID	access	access
GET	`/api/documents/barcode/{barCode}`	Get document by barcode	access	access
POST	`/api/documents`	Create new document	-	access
PUT	`/api/documents/{id}`	Update document	-	access
DELETE	`/api/documents/{id}`	Delete document	-	-
POST	`/api/documents/search`	Search documents	access	access
GET	`/api/documents/{id}/stream`	Stream document file (inline display)	access	access
GET	`/api/documents/{id}/download`	Download document file	access	access

Notes: - All GET operations return data filtered by user permissions (document type, country, counter party) - SuperUser sees all documents regardless of permissions - POST/PUT operations require Publisher or SuperUser role (enforced in service layer) - DELETE requires SuperUser role (enforced in service layer)

2. Counter Party Endpoints (`/api/counterparties`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	S
GET	`/api/counterparties`	Get all counter parties	access	access	a
GET	`/api/counterparties/search?searchTerm={term}`	Search counter parties	access	access	a
GET	`/api/counterparties/{id}`	Get counter party by ID	access	access	a
POST	`/api/counterparties`	Create counter party	-	-	a
PUT	`/api/counterparties/{id}`	Update counter party	-	-	a
DELETE	`/api/counterparties/{id}`	Delete counter party	-	-	a
GET	`/api/counterparties/{id}/usage`	Get counter party usage count	access	access	a

Notes: - All roles can view counter parties - Only SuperUser can create, update, or delete counter parties - Usage endpoint shows how many documents and user permissions reference this counter party

3. Country Endpoints (`/api/countries`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	Supe
GET	`/api/countries`	Get all countries	access	access	acces
GET	`/api/countries/{code}`	Get country by code	access	access	acces
POST	`/api/countries`	Create country	-	-	acces
PUT	`/api/countries/{code}`	Update country	-	-	acces
DELETE	`/api/countries/{code}`	Delete country	-	-	acces
GET	`/api/countries/{code}/usage`	Get country usage count	access	access	acces

Notes: - All roles can view countries - Only SuperUser can create, update, or delete countries - Usage endpoint shows counter parties and user permissions for this country

4. Currency Endpoints (`/api/currencies`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	Sup
GET	`/api/currencies`	Get all currencies	access	access	acc
GET	`/api/currencies/{code}`	Get currency by code	access	access	acc
POST	`/api/currencies`	Create currency	-	-	acc
PUT	`/api/currencies/{code}`	Update currency	-	-	acc
DELETE	`/api/currencies/{code}`	Delete currency	-	-	acc
GET	`/api/currencies/{code}/usage`	Get currency usage count	access	access	acc

Notes: - All roles can view currencies - Only SuperUser can create, update, or delete currencies - Usage endpoint shows how many documents use this currency

5. Document Type Endpoints (`/api/documenttypes`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	Su
GET	`/api/documenttypes`	Get all enabled document types	access	access	acc
GET	`/api/documenttypes/all`	Get all document types (including disabled)	access	access	acc
GET	`/api/documenttypes/{id}`	Get document type by ID	access	access	acc
POST	`/api/documenttypes`	Create document type	-	-	acc
PUT	`/api/documenttypes/{id}`	Update document type	-	-	acc
DELETE	`/api/documenttypes/{id}`	Delete document type	-	-	acc
GET	`/api/documenttypes/{id}/usage`	Get document type usage count	access	access	acc

Notes: - All roles can view document types - Only SuperUser can create, update, or delete document types - Usage endpoint shows documents, document names, and user permissions for this type

---

6. Document Name Endpoints (`/api/documentnames`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	
GET	`/api/documentnames`	Get all document names	access	a
GET	`/api/documentnames/bytype/{documentTypeId}`	Get document names by type	access	a
GET	`/api/documentnames/{id}`	Get document name by ID	access	a

Notes: - Read-only API (no create/update/delete endpoints) - All roles can view document names - Document names are filtered by document type

---

7. Scanned File Endpoints (`/api/scannedfiles`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publis
GET	`/api/scannedfiles`	Get all scanned files from directory	access	access
GET	`/api/scannedfiles/{fileName}`	Get scanned file metadata	access	access
GET	`/api/scannedfiles/{fileName}/content`	Get scanned file content (download)	access	access
GET	`/api/scannedfiles/{fileName}/exists`	Check if scanned file exists	access	access
GET	`/api/scannedfiles/{fileName}/stream`	Stream scanned file (inline display)	access	access
DELETE	`/api/scannedfiles/{fileName}`	Delete scanned file	-	-

Notes: - All roles can view and download scanned files - Only SuperUser can delete scanned files from the network directory - Files are served from the configured scan directory path

8. User Permission Endpoints (`/api/userpermissions`)

Base Authorization: `RequireAuthorization()` (All authenticated users)

HTTP Method	Endpoint	Description	Reader	Publishe
GET	`/api/userpermissions?accountNameFilter={filter}`	Get all user permissions	-	-
GET	`/api/userpermissions/users?accountNameFilter={filter}`	Get all DocuScan users	-	-
GET	`/api/userpermissions/{id}`	Get user permission by ID	-	-
GET	`/api/userpermissions/user/{userId}`	Get permissions by user ID	-	-
POST	`/api/userpermissions`	Create user permission	-	-
PUT	`/api/userpermissions/{id}`	Update user permission	-	-
DELETE	`/api/userpermissions/{id}`	Delete user permission	-	-
DELETE	`/api/userpermissions/user/{userId}`	Delete DocuScan user	-	-
POST	`/api/userpermissions/user`	Create DocuScan user	-	-
PUT	`/api/userpermissions/user/{userId}`	Update DocuScan user	-	-

Notes: - **SuperUser only** - All endpoints require SuperUser authorization - Used for user and permission management - Allows creating users, assigning permissions, and managing access

9. Audit Trail Endpoints (`/api/audittrail`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publis
POST	`/api/audittrail`	Log audit entry by barcode	access	access
POST	`/api/audittrail/document/{documentId}`	Log audit entry by document ID	access	access
POST	`/api/audittrail/batch`	Log batch audit entries	access	access
GET	`/api/audittrail/barcode/{barCode}?limit={limit}`	Get audit entries by barcode	access	access
GET	`/api/audittrail/user/{username}?limit={limit}`	Get audit entries by user	access	access
GET	`/api/audittrail/recent?limit={limit}`	Get recent audit entries	access	access
GET	`/api/audittrail/daterange?startDate={start}&endDate={end}&action={action}`	Get audit entries by date range	access	access

Notes: - All roles can log and view audit trail entries - Audit entries are filtered by document access permissions - SuperUser can view all audit entries - Common actions: Create, Read, Update, Delete, CheckIn, Export, AccessRequest, Login, Logout

10. Excel Export Endpoints (`/api/excel`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	Sup
POST	`/api/excel/export/documents`	Export documents to Excel	access	access	acc
POST	`/api/excel/validate/documents`	Validate export size	access	access	acc
GET	`/api/excel/metadata/documents`	Get document export metadata	access	access	acc

Notes: - All roles can export documents to Excel - Export results are filtered by user permissions - Maximum export size enforced (configured in appsettings) - Returns `.xlsx` file with search results

11. Action Reminder Endpoints (`/api/action-reminders`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	Super
GET	`/api/action-reminders?dateFrom={from}&dateTo={to}&documentTypeIds={ids}&counterPartyIds={ids}&counterPartySearch={term}&searchString={str}&includeFutureActions={bool}&includeOverdueOnly={bool}`	Get due actions with filters	access	access	access
GET	`/api/action-reminders/count`	Get due actions count	access	access	access
GET	`/api/action-reminders/date/{date}`	Get actions due on specific date	access	access	access

Notes: - All roles can view action reminders - Results filtered by document access permissions - Readers see reminders for documents they can access - SuperUser sees all reminders regardless of permissions

12. Configuration Endpoints (`/api/configuration`)

Base Authorization: `RequireAuthorization("SuperUser")` (SuperUser only)

HTTP Method	Endpoint	Description
Email Recipients		
GET	`/api/configuration/email-recipients`	Get all email recipient groups
GET	`/api/configuration/email-recipients/{groupKey}`	Get specific email recipient group
POST	`/api/configuration/email-recipients/{groupKey}`	Update email recipient group
Email Templates		
GET	`/api/configuration/email-templates`	Get all email templates
GET	`/api/configuration/email-templates/{key}`	Get email template by key
POST	`/api/configuration/email-templates`	Create email template
PUT	`/api/configuration/email-templates/{id}`	Update email template
DELETE	`/api/configuration/email-templates/{id}`	Deactivate email template (soft delete)
POST	`/api/configuration/email-templates/preview`	Preview email template with sample data
GET	`/api/configuration/email-templates/placeholders`	Get available placeholders for templates
GET	`/api/configuration/email-templates/diagnostic/DocumentAttachment`	Diagnostic for DocumentAttachment template
Configuration CRUD		
GET	`/api/configuration/sections`	List all configuration sections
GET	`/api/configuration/{section}/{key}`	Get specific configuration value
POST	`/api/configuration/{section}/{key}`	Set configuration value
SMTP & Testing		
POST	`/api/configuration/smtp?skipTest={bool}`	Update all SMTP settings atomically
POST	`/api/configuration/test-smtp`	Test SMTP connection
POST	`/api/configuration/reload`	Reload configuration cache
POST	`/api/configuration/migrate?overwriteExisting={bool}`	Migrate configuration from appsettings to database

Notes: - **SuperUser only** - All configuration endpoints require SuperUser authorization - Used for system administration - Includes email templates, SMTP settings, and application configuration - SMTP changes can be tested before saving - Configuration changes are logged to audit trail

---

13. Email Endpoints (`/api/email`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reader	Publisher	SuperUser
POST	`/api/email/send`	Send custom email with HTML body	-	access	access
POST	`/api/email/send-with-attachments`	Send email with document attachments	-	access	access
POST	`/api/email/send-with-links`	Send email with document links	-	access	access

Notes: - Only Publisher and SuperUser can send emails (enforced in UI/service layer) - Readers cannot send emails (view-only access) - Email templates loaded from database configuration - Document attachments require user to have access to the documents

---

14. Report Endpoints (`/api/reports`)

Base Authorization: `RequireAuthorization("HasAccess")` (Reader, Publisher, SuperUser)

HTTP Method	Endpoint	Description	Reade
Report Data (JSON)			
GET	`/api/reports/barcode-gaps`	Get barcode gaps report	access
GET	`/api/reports/duplicate-documents`	Get duplicate documents report	access
GET	`/api/reports/unlinked-registrations`	Get unlinked registrations report	access
GET	`/api/reports/scan-copies`	Get scan copies report	access
GET	`/api/reports/suppliers`	Get suppliers/counterparty statistics	access
GET	`/api/reports/all-documents`	Get all documents report	access
Excel Exports			
GET	`/api/reports/barcode-gaps/excel`	Export barcode gaps to Excel	access
GET	`/api/reports/duplicate-documents/excel`	Export duplicate documents to Excel	access
GET	`/api/reports/unlinked-registrations/excel`	Export unlinked registrations to Excel	access
GET	`/api/reports/scan-copies/excel`	Export scan copies to Excel	access
GET	`/api/reports/suppliers/excel`	Export suppliers to Excel	access
GET	`/api/reports/all-documents/excel`	Export all documents to Excel	access
POST	`/api/reports/documents/search/excel`	Export document search results to Excel	access
POST	`/api/reports/documents/selected/excel`	Export selected documents to Excel	access

Notes: - All roles can view and export reports - Report data is filtered by user permissions (documents) - SuperUser sees unfiltered data - Excel exports return `.xlsx` files - Includes specialized reports: barcode gaps, duplicates, unlinked registrations, scan copies, suppliers

15. User Identity Endpoints (`/api/user`)

Base Authorization: `RequireAuthorization()` (All authenticated users)

HTTP Method	Endpoint	Description	Reader	Publisher	SuperUser
GET	`/api/user/identity`	Get current user identity and claims	access	access	access

Notes: - Available to all authenticated users - Returns user name, authentication type, and all claims - Used for debugging authentication and displaying user information - No sensitive operations - read-only endpoint

16. Test Identity Endpoints (`/api/test-identity`) - DEBUG ONLY

Base Authorization: None (No authorization required)

HTTP Method	Endpoint	Description	Reader	Publisher	Sup
GET	`/api/test-identity/profiles`	Get available test identity profiles	access	access	acc
GET	`/api/test-identity/status`	Get current test identity status	access	access	acc
POST	`/api/test-identity/activate/{profileId}`	Activate a test identity profile	access	access	acc
POST	`/api/test-identity/reset`	Reset to real identity	access	access	acc

Notes: - **DEVELOPMENT ONLY** - These endpoints only exist when compiled in DEBUG mode - Removed in production builds - Used for testing different user roles and permissions - Allows switching between test user profiles without AD authentication - No authorization required to facilitate testing - **DO NOT** deploy to production with DEBUG enabled

---

Summary by Role

Reader Access Summary

- **Total Endpoints:** 60 accessible (excluding DEBUG-only endpoints: 56)
- **Primary Use Cases:**
  - View documents (filtered by permissions)
  - Search documents
  - View reference data (countries, currencies, document types, counter parties)
  - View scanned files
  - Export to Excel (filtered results)
  - View action reminders
  - View audit trail
  - View specialized reports (barcode gaps, duplicates, etc.)
  - Export reports to Excel
- **Cannot Access:**
  - Create/Update/Delete any data
  - Send emails
  - Manage users or permissions
  - System configuration
  - Delete scanned files

Publisher Access Summary

- **Total Endpoints:** 69 accessible (excluding DEBUG-only endpoints: 65)
- **Primary Use Cases:**
  - All Reader capabilities
  - Create and update documents
  - Send emails with attachments/links
  - Log audit entries
- **Cannot Access:**
  - Delete documents

- Create/Update/Delete reference data (countries, currencies, etc.)
- Manage users or permissions
- System configuration
- Delete scanned files

SuperUser Access Summary

- **Total Endpoints:** 108 accessible (all endpoints, excluding DEBUG-only endpoints: 104)
- **Primary Use Cases:**
  - All Reader and Publisher capabilities
  - Delete documents
  - Manage all reference data
  - Manage users and permissions
  - System configuration and administration
  - Delete scanned files
  - View all data without permission filters

Endpoint Count by Category

Category	Total Endpoints	Reader	Publisher	SuperUser
Documents	9	6 read	8 (read + create/update)	9 (all)
Counter Parties	7	4 read	4 read	7 (all)
Countries	6	3 read	3 read	6 (all)
Currencies	6	3 read	3 read	6 (all)
Document Types	7	4 read	4 read	7 (all)
Document Names	3	3 read	3 read	3 read
Scanned Files	6	5 read	5 read	6 (all)
User Permissions	10	0	0	10 (all)
Audit Trail	7	7	7	7
Excel Export	3	3	3	3
Action Reminders	3	3	3	3
Configuration	19	0	0	19 (all)
Email	3	0	3	3
Reports	14	14	14	14
User Identity	1	1	1	1
Test Identity (DEBUG)	4	4	4	4
TOTAL	**108**	**60**	**69**	**108**
TOTAL (Production)	**104**	**56**	**65**	**104**

Authorization Patterns

Pattern 1: Read-Only Reference Data

Example: Countries, Currencies, Document Types, Document Names

- **GET** - All roles (HasAccess)
- **POST/PUT/DELETE** - SuperUser only

Reason: Reference data should be stable and managed centrally by administrators.

Pattern 2: Full CRUD with Role Restrictions

Example: Documents

- **GET** - All roles (HasAccess), filtered by permissions
- **POST/PUT** - Publisher, SuperUser
- **DELETE** - SuperUser only

Reason: Documents can be created/edited by Publishers, but deletion requires admin privileges.

Pattern 3: Admin-Only Management

Example: User Permissions, Configuration

- **All Operations** - SuperUser only

Reason: Security-sensitive operations that affect system behavior or user access.

Pattern 4: Operational Endpoints

Example: Email, Audit Trail, Excel Export

- **Operations** - Based on functional need (e.g., Publishers can send emails, all can export)

Reason: Functional capabilities tied to role responsibilities.

Permission Filtering

Client-Side vs Server-Side Filtering

Server-Side Filtering (Recommended): - Documents: Filtered by `DocumentTypeId`, `CountryCode`, `CounterPartyId` - Action Reminders: Filtered by document access - Audit Trail: Filtered by document access - Excel Export: Results filtered before export

SuperUser Bypass: - SuperUser role bypasses all permission filters - Sees all data regardless of assigned permissions - No `UserPermission` records needed for SuperUser

Implementation Pattern

```
// In service layer
var currentUser = await _currentUserService.GetCurrentUserAsync();

if (currentUser.IsSuperUser)
{
    // Return all data
    return await _dbContext.Documents.ToListAsync();
}
else
{
    // Filter by permissions
    return await _dbContext.Documents
        .Where(d => currentUser.CanAccessDocument(d.DocumentTypeId, d.Coun
        .ToListAsync();
}
```

Testing Authorization

Test Reader Access

```
# Login as user in "Reader" AD group
# Should be able to GET but not POST/PUT/DELETE

curl -X GET https://localhost:44101/api/documents \
  -u "DOMAIN\reader"

curl -X POST https://localhost:44101/api/documents \
  -u "DOMAIN\reader" \
  -H "Content-Type: application/json" \
  -d '{...}'
# Expected: 403 Forbidden or filtered by service layer
```

Test Publisher Access

```
# Login as user in "Publisher" AD group
# Should be able to GET, POST, PUT but not DELETE

curl -X DELETE https://localhost:44101/api/documents/123 \
  -u "DOMAIN\publisher"
# Expected: 403 Forbidden or filtered by service layer
```

Test SuperUser Access

```
# Login as user in "SuperUser" AD group or database IsSuperUser=true
# Should be able to perform all operations

curl -X DELETE https://localhost:44101/api/documents/123 \
  -u "DOMAIN\superuser"
# Expected: 204 No Content (success)

curl -X GET https://localhost:44101/api/userpermissions \
  -u "DOMAIN\superuser"
# Expected: 200 OK with all users
```

Related Documentation

- **AUTHORIZATION_GUIDE.md** - Complete authorization implementation guide
- **AUTHORIZATION_MATRIX.md** - High-level authorization by feature
- **AD_GROUPS_QUICK_REFERENCE.md** - Active Directory group reference
- **CLAUDE.md** - Development guidelines

Status: □ Current as of 2025-11-07 **Maintained by:** IkeaDocuScan Development Team **Review Frequency:** On API changes or authorization policy updates

Recent Changes: - **2025-11-07:** Added Report Endpoints (14), User Identity Endpoints (1), Test Identity Endpoints (4 - DEBUG only) - Updated total endpoint count from 89 to 108 (104 in production)