

FileCap

Public API Reference

Integration & mail-plugin endpoints

1. Introduction

This document describes the public HTTP API exposed by the FileCap server. These endpoints are intended for mail plugins, third-party integrations and other external systems that need to upload files, request invites, validate transfer metadata or read server configuration. Internal endpoints used by the admin panel, the OWA add-in and partner / license-server integrations are out of scope.

The API has grown in two distinct generations:

1. **Legacy API** – a set of `.jsp` URLs (e.g. `process_upload.jsp`) that historically powered the Outlook classic plugin and similar integrations. Each JSP performs a server-side forward to a modern REST endpoint; both the JSP and the modern path keep working.
2. **Modern API** – JSON / multipart endpoints under `/FileCap/api/...` that return a structured `ApiResponse<T>` envelope. New integrations should target this surface.

1.1 Base URL & context root

All endpoints are served from the FileCap web application at the context root `/FileCap`. The Postman collection that ships with this document uses a `baseUrl` variable that should contain the scheme + host (no trailing slash), e.g.:

```
https://filecap.example.com
```

Resulting URLs are then constructed as `{{baseUrl}}/FileCap/<path>`.

1.2 Authentication

Most endpoints accept an `APIKey` in the request body (form field, urlencoded parameter or JSON property — the modern variants accept `apiKey` as well). The key is configured per FileCap tenant and is required for every integration call unless the endpoint is explicitly public (e.g. `password-rules` and the `validate/*` endpoints).

A subset of the modern endpoints additionally accepts a `deviceApiKey`, returned by the device registration flow (section 9). Endpoints that require web-portal authentication use either the portal session cookie or a JWT Bearer token (scope `transfer:free`).

1.3 Response envelope (modern API)

Modern JSON endpoints wrap their result in:

```
{
  "success": true,
  "value": { ... },
  "errorMessage": null
}
```

On failure, `success` is false, `value` is null and `errorMessage` contains the reason.

1.4 Response style (legacy API)

The legacy `/*.jsp` endpoints return plain text:

- `true` — request succeeded.

- `false|<ERROR_CODE>` — request failed; the code identifies the cause.

1.5 Rate limiting

Several endpoints are rate-limited per session, per JWT or globally. Limits are shown in the endpoint reference where relevant. When a limit is exceeded the server responds with HTTP 429.

2. Legacy API (JSP)

These are the original public endpoints. They are kept for backward compatibility with deployed mail plugins and third-party scripts. New integrations should prefer the modern equivalents in the next chapters; the table below shows which legacy URL forwards to which modern path.

Legacy URL	Method	Forwards to	Notes
process_upload.jsp	POST	/api/transfer/v1/send-outlook-transfer	multipart/form-data, text/plain response
invite.jsp	POST	/api/invite/old	application/x-www-form-urlencoded
mailPluginSettings.jsp	POST	/api/settings/outlook-windows	application/x-www-form-urlencoded → XML
checkFile.jsp	POST	/api/validate/file	application/x-www-form-urlencoded
checkTransfer.jsp	POST	/api/validate/transfer	application/x-www-form-urlencoded
domainCheck.jsp	POST	/api/validate/domain	application/x-www-form-urlencoded

Each legacy endpoint is documented in detail below.

2.1 Upload files — process_upload.jsp

POST {{baseUrl}}/FileCap/process_upload.jsp

Content-Type: multipart/form-data

Upload one or more files in a single transfer. Forwards to /api/transfer/v1/send-outlook-transfer. Returns plain text: true on success, false|<ERROR_CODE> on failure.

Field ordering: the server streams the multipart body and processes parts in the order they arrive. The fileMetadata field MUST be sent BEFORE the file parts (file01, file02, ...). When a file part is read the server looks up its expected size in the metadata list; on S3 storage the upload is rejected with "File metadata required" when the metadata has not yet been seen, and on FREE-tier portals a NullPointerException is raised. A typical multipart body therefore starts with all text fields (APIKey, ids, from, ..., fileMetadata) and ends with the binary file01 / file02 / ... parts.

Parameters

Name	Type	Req.	Description
APIKey	string	yes	FileCap server API key.
deviceApiKey	string	no	Per-device API key when the integration is registered as a device.
ids	string	yes	Transfer ID. Must be unique per transfer.

from	<i>string</i>	yes	Sender e-mail address.
fromName	<i>string</i>	no	Display name of the sender.
subject	<i>string</i>	yes	Subject of the transfer.
comment	<i>string</i>	yes	Message body.
encryptMessage	<i>boolean</i>	no	When true the message body is encrypted (plain text only). Requires a password.
rec0, rec1, ...	<i>string</i>	yes	TO recipients. The first is rec0, the next rec1 etc.
cc0, cc1, ...	<i>string</i>	no	CC recipients.
bcc0, bcc1, ...	<i>string</i>	no	BCC recipients.
nameRec0, ...	<i>string</i>	no	Display names for the matching recN recipient.
storeDays_value	<i>integer</i>	no	How long the transfer stays on the FileCap server.
downloadLoadable_value	<i>integer</i>	no	Maximum number of downloads.
password	<i>string</i>	no	Mandatory when encryptMessage = true or when policy requires a password.
useSmsTextAuthentication	<i>boolean</i>	no	Enable SMS verification for the recipient (requires mobileNr).
mobileNr	<i>string</i>	no	Recipient mobile number when SMS verification is used.
dontNotify	<i>boolean</i>	no	When true, FileCap does not send a notification mail to the recipient.
nrOfFilesInTransfer	<i>integer</i>	no	Explicit number of files in the transfer.
userAgent	<i>string</i>	no	Identifier of the calling client.
version	<i>string</i>	no	Version of the calling client.
fileMetadata	<i>JSON (text)</i>	yes	JSON array describing every file part. One entry per file with filename, mimetype and size (bytes). The filename MUST exactly match the filename of the corresponding file part. CRITICAL: this field MUST appear in the multipart body BEFORE the file parts (see note below).
file01, file02, ...	<i>file</i>	yes	File parts. The first file is file01, the next file02 etc. MUST appear AFTER the fileMetadata field.

Response

Plain text.

```
true
false|VIRUS_FOUND
```

Error codes

- **VIRUS_FOUND** — A virus was detected in one of the files. The transfer was cancelled.
- **NO_RECIPIENT_GIVEN** — No recipients were supplied for the transfer.
- **FILETYPE_NOT_ALLOW** — File type forbidden by MIME or extension policy.
- **PASSWORD_MANDATORY_BUT_NOT_GIVEN** — A password is required by policy but missing.
- **CANNOT_SEND_MAIL** — FileCap could not send the notification mail (mailserver / relay misconfigured).

- **API_KEY_INVALID** — The supplied APIKey is not valid.
- **GENERAL_ERROR** — Unknown error; check the FileCap server log for the stack trace. Often a network / proxy issue.

2.2 Send invite — invite.jsp

POST {{baseUrl}}/FileCap/invite.jsp

Content-Type: application/x-www-form-urlencoded

Send a FileCap invite to an external user, asking them to upload files for the inviter. Forwards to /api/invite/old. Returns plain text true / false | <ERROR_CODE>.

Parameters

Name	Type	Req.	Description
APIKey	string	yes	FileCap server API key.
deviceApiKey	string	no	Per-device API key.
id	string	yes	Invite ID. MUST start with the prefix iiiii.
senderEmail	string	yes	E-mail address of the person who needs to receive the invite (the inviter).
receiverEmail	string	yes	E-mail address of the person who needs to receive the files (normally your own e-mail address).
receiverMobileNr	string	no	Mobile number of the receiver when SMS verification is required.
message	string	no	Free-form message attached to the invite.
notify	boolean	no	When false, FileCap will not send a notification mail (the calling system must handle it).

Response

Plain text.

```
true
false|CANNOT_SEND_MAIL
```

Error codes

- **CANNOT_SEND_MAIL** — The mailserver could not process the invite mail.
- **CANNOT_ADD_INVITE_TO_DB** — FileCap could not persist the invite ID.
- **UNKNOWN_ERROR** — An exception was raised. Check the server log.

2.3 Mail plugin settings — mailPluginSettings.jsp

POST {{baseUrl}}/FileCap/mailPluginSettings.jsp

Content-Type: application/x-www-form-urlencoded

Request the FileCap server settings used by mail plugins (Outlook classic etc.). Forwards to `/api/settings/outlook-windows`. Returns the plug-in configuration as XML.

Parameters

Name	Type	Req.	Description
APIKey	<i>string</i>	yes	FileCap server API key.
userAgent	<i>string</i>	no	Identifier of the calling client.
version	<i>string</i>	no	Version of the calling client.
lang	<i>string</i>	no	Locale (e.g. NL, EN).

Response

XML document containing maximum file size, allowed extensions, password policy, verification methods, branding, etc.

2.4 Check file — `checkFile.jsp`

POST `{{baseUrl}}/FileCap/checkFile.jsp`

Content-Type: `application/x-www-form-urlencoded`

Verify – before uploading – whether a single file is allowed by the server policy (extension and MIME blocklists, sender restrictions). Forwards to `/api/validate/file`.

Parameters

Name	Type	Req.	Description
APIKey	<i>string</i>	yes	FileCap server API key.
filename	<i>string</i>	yes	Name of the file.
sender	<i>string</i>	yes	Sender e-mail address.
mime	<i>string</i>	yes	MIME type of the file.
fileID	<i>string</i>	no	Identifier of the file.

Response

Plain text true when allowed, false|<ERROR_CODE> when blocked.

2.5 Inspect transfer — `checkTransfer.jsp`

POST `{{baseUrl}}/FileCap/checkTransfer.jsp`

Content-Type: `application/x-www-form-urlencoded`

Inspect a previously created transfer (used by mail plugins to check status / contents). Forwards to `/api/validate/transfer`.

Parameters

Name	Type	Req.	Description
APIKey	<i>string</i>	yes	FileCap server API key.
deviceApiKey	<i>string</i>	no	Per-device API key.
id	<i>string</i>	yes	Transfer ID.
email	<i>string</i>	no	Recipient e-mail; when supplied the response is scoped to this recipient.

Response

Plain text response with the transfer status / file list.

2.6 Check domain — domainCheck.jsp

POST `{{baseUrl}}/FileCap/domainCheck.jsp`

Content-Type: `application/x-www-form-urlencoded`

Verify whether the sender domain is allowed to send to the recipient domain (whitelist / IP rules). Forwards to `/api/validate/domain`.

Parameters

Name	Type	Req.	Description
email	<i>string</i>	yes	Recipient e-mail address whose domain has to be validated.
sender	<i>string</i>	yes	Sender e-mail address.

Response

Plain text `true / false|<ERROR_CODE>`.

3. Transfer

Modern transfer endpoints under `/FileCap/api/transfer`. The `send-outlook-transfer` endpoint replaces the legacy `process_upload.jsp` flow but returns a structured JSON `ApiResponse` instead of plain text. The `/transfer/v1/send-outlook-transfer` endpoint is the path the legacy JSP forwards to (text response) and is documented for completeness.

*All multipart upload endpoints in this chapter (`send-outlook-transfer`, `v1/send-outlook-transfer`, `send-transfer`, `send-reply`) share a single multipart parser. They all accept — and require — a `fileMetadata` form field that **MUST** be sent **BEFORE** the file parts. See the note in section 3.1 for details and consequences when the order is reversed.*

3.1 Send Outlook transfer (modern, JSON response)

POST `{{baseUrl}}/FileCap/api/transfer/send-outlook-transfer`

Content-Type: `multipart/form-data`

Modern Outlook plugin upload endpoint. Same form fields as `process_upload.jsp`; the response is a JSON `ApiResponse<ApiUploadValue>`. Rate-limited per session (`SendOutlookTransfer`).

Field ordering: like the legacy endpoint, the `fileMetadata` form field is required and **MUST** be sent **BEFORE** the file parts. The server processes the multipart body in stream order; for each file part it looks up the expected size in the metadata list, and rejects the upload ("File metadata required" on S3 storage; `NullPointerException` on FREE-tier portals) when the metadata has not yet been seen.

Example `fileMetadata` value (sent as a single text form field)

```
[
  { "filename": "report.pdf", "mimetype": "application/pdf", "size": 102400 },
  { "filename": "photo.jpg", "mimetype": "image/jpeg", "size": 51200 }
]
```

Parameters

Name	Type	Req.	Description
APIKey	<i>string</i>	yes	FileCap server API key (alias: <code>apiKey</code>).
deviceApiKey	<i>string</i>	no	Per-device API key.
ids	<i>string</i>	yes	Transfer ID (alias: <code>transferId</code>).
from	<i>string</i>	yes	Sender e-mail (aliases: <code>fromEmail</code> , <code>senderEmail</code> , <code>senderEmailAddress</code>).
fromName	<i>string</i>	no	Sender display name (alias: <code>senderName</code>).
subject	<i>string</i>	yes	Transfer subject.
comment	<i>string</i>	yes	Message body.
encryptMessage	<i>boolean</i>	no	Encrypt the message body (alias: <code>useEncryption</code>).
rec0, rec1, ...	<i>string</i>	yes	TO recipients (alias: <code>recipient</code>).
cc0, cc1, ...	<i>string</i>	no	CC recipients.
bcc0, bcc1, ...	<i>string</i>	no	BCC recipients.

maxDownloads	<i>integer</i>	no	Maximum number of downloads (alias: downloadLoadable_value).
maxDaysToStore	<i>integer</i>	no	Retention period in days (alias: storeDays_value).
password	<i>string</i>	no	Transfer password.
useSmsTextAuthentication	<i>boolean</i>	no	Enable SMS verification (alias: enableToken).
verificationMethod1	<i>string</i>	no	Primary verification method (e.g. EMAIL_ACCESS_CODE, SMS, PASSWORD, NONE).
verificationMethod2	<i>string</i>	no	Secondary verification method.
dontNotify	<i>boolean</i>	no	When true, FileCap does not send a notification mail.
nrOfFilesInTransfer	<i>integer</i>	no	Explicit file count.
userAgent / version	<i>string</i>	no	Identification of the calling client.
fileMetadata	<i>JSON (text)</i>	yes	JSON array describing every file part. One entry per file with filename, mimetype and size (bytes). The filename MUST exactly match the corresponding file part. CRITICAL: must appear BEFORE the file parts in the multipart body.
file01, file02, ...	<i>file</i>	yes	File parts. MUST appear AFTER the fileMetadata field.

Response

JSON. Example:

```
{
  "success": true,
  "value": {
    "transferId": "263478632784632748237513",
    "generatedPassword": "Ax9!kp2Qm"
  },
  "errorMessage": null
}
```

Notes

- Rate limit: SendOutlookTransfer (per session).

3.2 Send Outlook transfer (legacy v1, text response)

POST {{baseUrl}}/FileCap/api/transfer/v1/send-outlook-transfer

Content-Type: multipart/form-data

Plain-text variant of the upload (returns true / false|<ERROR_CODE>). This is the endpoint that process_upload.jsp forwards to. Use it when the calling client requires the legacy text response shape.

Parameters

Name	Type	Req.	Description
APIKey	<i>string</i>	yes	FileCap server API key.
deviceApiKey	<i>string</i>	no	Per-device API key when the integration is registered as a device.
ids	<i>string</i>	yes	Transfer ID. Must be unique per transfer.

from	<i>string</i>	yes	Sender e-mail address.
fromName	<i>string</i>	no	Display name of the sender.
subject	<i>string</i>	yes	Subject of the transfer.
comment	<i>string</i>	yes	Message body.
encryptMessage	<i>boolean</i>	no	When true the message body is encrypted (plain text only). Requires a password.
rec0, rec1, ...	<i>string</i>	yes	TO recipients. The first is rec0, the next rec1 etc.
cc0, cc1, ...	<i>string</i>	no	CC recipients.
bcc0, bcc1, ...	<i>string</i>	no	BCC recipients.
nameRec0, ...	<i>string</i>	no	Display names for the matching recN recipient.
storeDays_value	<i>integer</i>	no	How long the transfer stays on the FileCap server.
downloadLoadable_value	<i>integer</i>	no	Maximum number of downloads.
password	<i>string</i>	no	Mandatory when encryptMessage = true or when policy requires a password.
useSmsTextAuthentication	<i>boolean</i>	no	Enable SMS verification for the recipient (requires mobileNr).
mobileNr	<i>string</i>	no	Recipient mobile number when SMS verification is used.
dontNotify	<i>boolean</i>	no	When true, FileCap does not send a notification mail to the recipient.
nrOfFilesInTransfer	<i>integer</i>	no	Explicit number of files in the transfer.
userAgent	<i>string</i>	no	Identifier of the calling client.
version	<i>string</i>	no	Version of the calling client.
fileMetadata	<i>JSON (text)</i>	yes	JSON array describing every file part. One entry per file with filename, mimetype and size (bytes). The filename MUST exactly match the filename of the corresponding file part. CRITICAL: this field MUST appear in the multipart body BEFORE the file parts (see note below).
file01, file02, ...	<i>file</i>	yes	File parts. The first file is file01, the next file02 etc. MUST appear AFTER the fileMetadata field.

Response

Plain text.

```
true
false|API_KEY_INVALID
```

3.3 Validate upload (pre-flight)

POST {{baseUrl}}/FileCap/api/transfer/validate-upload

Content-Type: application/json

Validate the metadata of an upload before sending bytes. Checks server quota, allowed extensions / MIME, max size etc. The aliased path /transfer/validateUpload works as well.

Request body

```
{
  "lang": "NL",
  "fileMetadataList": [
    {
      "fileName": "report.pdf",
      "fileSize": 102400,
      "mimeType": "application/pdf"
    }
  ]
}
```

Response

ApiResponse<Void> — only success and errorMessage are populated.

3.4 Validate files

POST {{baseUrl}}/FileCap/api/transfer/validate-files

Content-Type: application/json

Per-file validation. Returns one entry per file indicating whether it is allowed and, if not, which rule was violated.

Request body

```
{
  "lang": "NL",
  "fileMetadataList": [
    { "fileName": "clean.pdf", "fileSize": 1024, "mimeType": "application/pdf" },
    { "fileName": "forbidden.exe", "fileSize": 2048, "mimeType": "application/x-msdownload" }
  ]
}
```

Response

ApiResponse<ValidateFilesResultModel>.

3.5 Validate recipient

POST {{baseUrl}}/FileCap/api/transfer/validate-recipient

Content-Type: application/json

Validate the sender / recipient combination against domain whitelists, IP allow-lists and the emailMaySend / emailMayReceive rules.

Request body

```
{
  "senderEmailAddress": "kees@example.com",
```

```
"recipientEmailAddress": "piet@receiver.com"
}
```

Response

ApiResponse<Void>.

Error codes

- **VALIDATION_ERROR_DOMAIN_INVALID** — Recipient or sender domain is not allowed.
- **VALIDATION_ERROR_EMAIL_ADDRESS** — One of the e-mail addresses is malformed.

3.6 Send reply

POST {{baseUrl}}/FileCap/api/transfer/send-reply

Content-Type: multipart/form-data

Send a secure reply to an existing transfer. The original transfer is identified by its transfer ID. Rate-limited (SendReply). Same multipart parser as section 3.1, so when file parts are attached the fileMetadata field MUST appear before them.

Parameters

Name	Type	Req.	Description
ids	string	yes	ID of the original transfer.
from	string	yes	Sender e-mail.
subject	string	yes	Reply subject.
comment	string	no	Reply message.
replyToAll	boolean	no	Reply to every recipient of the original transfer.
fileMetadata	JSON (text)	yes*	JSON array describing every file part (filename, mimetype, size). Required when file parts are attached. MUST appear BEFORE the file parts.
file01, ...	file	no	Optional file parts. Place AFTER fileMetadata.

Response

ApiResponse<SentTransferInfo>.

3.7 Check connection

POST {{baseUrl}}/FileCap/api/transfer/checkConnection

Content-Type: application/json

Lightweight liveness probe. Always returns ApiResponse.success("connected") when the FileCap portal is reachable.

Response

Always returns:

```
{ "success": true, "value": "connected", "errorMessage": null }
```

4. Invite

Endpoints for sending FileCap invites. The whole controller is rate-limited at 10 requests / minute.

4.1 Send invite (modern JSON)

POST {{baseUrl}}/FileCap/api/invite

Content-Type: application/json

Modern JSON variant of the legacy invite.jsp flow. Accepts a list of invitees in a single call.

Request body

```
{
  "apiKey": "HKADASD68768768ASDASDASDAD",
  "deviceApiKey": "",
  "sender": {
    "emailAddress": "kees@example.com",
    "name": "Kees Janssen"
  },
  "invitees": [
    "piet@company.com",
    "jan@partner.com"
  ],
  "message": "Please upload the requested documents."
}
```

Response

ApiResult<Void>.

Notes

- Rate limit: 10 requests / minute (InviteService).

4.2 Send invite (legacy form, /invite/old)

POST {{baseUrl}}/FileCap/api/invite/old

Content-Type: application/x-www-form-urlencoded

Direct call to the endpoint that invite.jsp forwards to. Returns plain text true / false | <ERROR_CODE>.

Parameters

Name	Type	Req.	Description
APIKey	string	yes	FileCap server API key.
deviceApiKey	string	no	Per-device API key.
id	string	yes	Invite ID, must start with iiii.
senderEmail	string	yes	Inviter e-mail.
receiverEmail	string	yes	Receiver e-mail.
receiverMobileNr	string	no	Receiver mobile number.

message	<i>string</i>	no	Optional free-form message.
notify	<i>boolean</i>	no	When false, FileCap will not send a notification mail.

Response

Plain text true / false|<ERROR_CODE>.

4.3 Generate invite ID

POST {{baseUrl}}/FileCap/api/invite/generate-id

Content-Type: application/json

Pre-generate the invite ID used in the id field of the legacy invite call (the value starting with iiii). Aliased path: /invite/generateInvitelId.

Request body

```
{
  "apiKey": "HKADASD68768768ASDASDASDAD",
  "deviceApiKey": "",
  "sender": {
    "emailAddress": "kees@example.com"
  }
}
```

Response

ApiResponse<String> — the generated invite ID.

5. Settings

Server settings used by clients and plugins: password policy, recipient verification options, branding, plugin settings.

5.1 Password rules (detailed)

POST `{{baseUrl}}/FileCap/api/settings/getPasswordRules`

Content-Type: application/json

Detailed description of the password requirements.

Request body

```
{
  "apiKey": "HKADASD68768768ASDASDASDAD",
  "lang": "NL"
}
```

Response

ApiResponse<PasswordRules>:

```
{
  "success": true,
  "value": {
    "enabled": true,
    "title": "Password requirements",
    "rules": [
      { "description": "At least 8 characters", "requirement": ".{8,}" },
      { "description": "At least one digit", "requirement": "(?=.*\\d)" }
    ]
  }
}
```

Error codes

- **UNEXPECTED_ERROR** — Internal server error; check the server log.

5.2 Password rules (GET, public)

GET `{{baseUrl}}/FileCap/api/settings/password-rules?lang=NL`

Public, unauthenticated GET version. Returns the same ApiResponse<PasswordRules> body.

Parameters

Name	Type	Req.	Description
lang	string	no	Locale code (e.g. NL, EN).

5.3 Password policy (short)

POST `{{baseUrl}}/FileCap/api/settings/getPasswordPolicy`

Content-Type: application/json

Short summary of the password requirements: minimum length, character classes, expiration, reuse.

Request body

```
{
  "apiKey": "HKADASD68768768ASDASDASDAD"
}
```

Response

ApiResponse<PasswordPolicy>.

Error codes

- **UNEXPECTED_ERROR** — Internal server error; check the server log.

5.4 Mail plugin settings (XML)

POST {{baseUrl}}/FileCap/api/settings/outlook-windows

Content-Type: application/x-www-form-urlencoded

Endpoint that mailPluginSettings.jsp forwards to. Returns the plug-in configuration as XML.

Parameters

Name	Type	Req.	Description
APIKey	string	yes	FileCap server API key.
userAgent	string	no	Identifier of the calling client.
version	string	no	Client version.
lang	string	no	Locale code.

Response

XML document. Also exposed as a JSON variant on the same path (POST application/json) which returns HTTP 202 + ApiResponse<OutlookClassicSettings>, or HTTP 401 when authentication fails.

5.5 Recipient verification options

POST {{baseUrl}}/FileCap/api/settings/recipient-verification-options

Content-Type: application/json

Lists the verification methods that can be applied to a transfer (e.g. NONE, EMAIL_ACCESS_CODE, SMS, PASSWORD) and which ones are mandatory by current policy.

Request body

```
{ "apiKey": "HKADASD68768768ASDASDASDAD" }
```

Response

ApiResponse<RecipientVerificationOptions>.

Error codes

- **UNEXPECTED_ERROR** — Internal server error; check the server log.

5.6 App settings

POST {{baseUrl}}/FileCap/api/settings/app-settings

Content-Type: application/json

Generic settings bundle for client apps: enabled features, max upload size, branding, supported languages, etc.

Request body

```
{
  "apiKey": "HKADASD68768768ASDASDASDAD",
  "lang": "NL"
}
```

Response

ApiResponse<AppSettings>.

Error codes

- **UNEXPECTED_ERROR** — Internal server error; check the server log.

5.7 Web portal settings

GET {{baseUrl}}/FileCap/api/settings?lang=NL

Returns ApiResponse<WebPortalSettings> for the web portal (theme, locale options, feature flags). Allowed for portal types MAIN, SUB and FREE.

Parameters

Name	Type	Req.	Description
lang	string	no	Locale code.

5.8 Is shared access code available

POST {{baseUrl}}/FileCap/api/settings/is-shared-access-code-available

Content-Type: application/json

Returns a raw boolean (true / false) — not wrapped in ApiResponse — indicating whether a shared access code has been configured for the given transfer.

Request body

```
{ "transferId": "263478632784632748237513" }
```

Response

Raw boolean: true or false. Returns false on any internal error.

6. Validation

Lightweight validators that mail plugins call before composing or sending a message. All endpoints under /api/validate return plain text (true / false|<ERROR_CODE>).

6.1 Check file (form)

POST {{baseUrl}}/FileCap/api/validate/file

Content-Type: application/x-www-form-urlencoded

Validate a single file by name + MIME against the policy engine. Path that checkFile.jsp forwards to.

Parameters

Name	Type	Req.	Description
APIKey	string	yes	FileCap server API key.
filename	string	yes	File name.
sender	string	yes	Sender e-mail.
mime	string	yes	MIME type.
fileID	string	no	File identifier.

Response

Plain text true / false|<ERROR_CODE>.

6.2 Check transfer

POST {{baseUrl}}/FileCap/api/validate/transfer

Content-Type: application/x-www-form-urlencoded

Path that checkTransfer.jsp forwards to.

Parameters

Name	Type	Req.	Description
APIKey	string	yes	FileCap server API key.
deviceApiKey	string	no	Per-device API key.
id	string	yes	Transfer ID.
email	string	no	Optional recipient e-mail.

Response

Plain text true / false|<ERROR_CODE>.

6.3 Check domain

POST {{baseUrl}}/FileCap/api/validate/domain

Content-Type: application/x-www-form-urlencoded

Path that domainCheck.jsp forwards to. Validates the sender domain against the recipient domain.

Parameters

Name	Type	Req.	Description
email	<i>string</i>	yes	Recipient e-mail.
sender	<i>string</i>	yes	Sender e-mail.

Response

Plain text true / false|<ERROR_CODE>.

6.4 Check sender may send

POST {{baseUrl}}/FileCap/api/validate/email-may-send

Content-Type: text/plain

Returns a raw boolean for the question "is this sender allowed to send via FileCap?". The body is the bare e-mail address as text/plain.

Response

Raw boolean: true / false.

7. Business Rules (DLP)

Endpoints for the policy / DLP engine. Both endpoints require either DEVICE or API_KEY authentication and are rate-limited at 60 requests / minute.

7.1 Scan attachment

POST `{{baseUrl}}/FileCap/api/business-rules/scan-attachment`

Content-Type: multipart/form-data

Run a DLP / business-rules scan against a single attachment.

Parameters

Name	Type	Req.	Description
APIKey	<i>string</i>	yes	FileCap server API key.
file	<i>file</i>	yes	File part to scan.

Response

ApiResponse<BusinessRulesScanResult> with the scan verdict and any policy violations.

Notes

- Rate limit: 60 requests / minute (ScanAttachment).

7.2 Scan text

POST `{{baseUrl}}/FileCap/api/business-rules/scan-text`

Content-Type: application/json

Run a DLP scan against arbitrary text.

Request body

```
{
  "text": "Hello, this is the message body that needs to be scanned for sensitive data."
}
```

Response

ApiResponse<BusinessRulesScanResult>.

Notes

- Rate limit: 60 requests / minute (ScanText).

8. Block transfer

Sender-facing endpoints to inspect and revoke (block) a transfer that has already been sent. When a transfer is created, FileCap stores a hashed blockPassword together with the transfer and e-mails the SENDER a notification that contains a block link of the form `https://<host>/FileCap/blockTransfer.jsp?id=<transferId>&blockId=<blockId>&email=<senderEmail>`. The JSP sanitizes the parameters and redirects to the portal “block” page, which loads transfer metadata via `GET /api/block/info` and — on confirmation — calls `POST /api/block` to actually block the transfer. The blockId is the secret: anyone holding transferId + blockId + sender e-mail can block the transfer; no API key or session is required. Every validation failure (wrong blockId, wrong sender, unknown transferId) deliberately returns the same generic error `PORTAL_BLOCK_TRANSFER_NOT_FOUND` so the caller cannot tell which field was wrong. Integrations that suppress FileCap’s own notification mails (`dontNotify=true` on upload) must capture and forward the blockId to the sender themselves — without it the transfer cannot be blocked through the API.

8.1 Get block info

GET `{{baseUrl}}/FileCap/api/block/info?transferId=...&blockId=...&emailAddress=...`

Return the metadata that the FileCap web UI shows on the block confirmation page: the TO recipients of the transfer, the file names and a flag that indicates whether the message body was encrypted. The call is read-only — it does not modify the transfer, send any mail or write to the audit log. Validation order (every failure collapses to `PORTAL_BLOCK_TRANSFER_NOT_FOUND` unless noted): (1) all three parameters must be non-blank; (2) the transfer must exist; (3) SHA-hash of blockId must match the stored blockPassword; (4) emailAddress must be a known sender of the transfer; (5) the transfer must not be `DISABLED` — otherwise `PORTAL_DOWNLOAD_TRANSFER_EXPIRED`; (6) the transfer must not already be `BLOCKED` — otherwise `PORTAL_BLOCK_TRANSFER_ALREADY_BLOCKED`.

Parameters

Name	Type	Req.	Description
transferId	string	yes	The transfer’s unique ID (the “ids” value supplied at upload time). Used to look up the transfer record.
blockId	string	yes	The block secret embedded in the FileCap notification mail to the sender (URL parameter blockId of blockTransfer.jsp). The server hashes this value and compares it to the blockPassword stored on the transfer.
emailAddress	string	yes	The e-mail address of the SENDER of the transfer. Validated against the transfer’s sender record (SenderDb.isSender). Not the recipient’s address.

Response

`ApiResult<TransferBlockInfo>` with three fields: “recipients” (array of e-mail addresses of the TO recipients), “files” (set of filenames inside the transfer, no path) and “hasEncryptedMessage” (true when the message body was encrypted at upload time). Example success body: `{ "success": true, "value": { "recipients": ["piet@company.com", "jan@partner.com"], "files": ["contract.pdf", "appendix.docx"], "hasEncryptedMessage": false }, "errorMessage": null }`.

8.2 Block transfer

POST `{{baseUrl}}/FileCap/api/block`

Content-Type: `application/json`

Mark the transfer as blocked. After a successful call: (1) the transfer status is set to BLOCKED — subsequent download attempts (web portal or API) are refused; (2) an NTA audit-log entry is written (“user <sender> blocked transfer with id <transferId>”); (3) a “message withdrawn” notification mail is sent to all TO and CC recipients of the original transfer. Request body fields: `transferId` (string, required) — the transfer’s unique ID (the “ids” value from the upload call); `blockId` (string, required) — the block secret from the sender notification mail; `emailAddress` (string, required) — the SENDER’s e-mail address (validated against the transfer’s sender record, not the recipient address).

Request body

```
{
  "transferId": "263478632784632748237513",
  "blockId": "BLK-1234",
  "emailAddress": "kees@example.com"
}
```

Response

`ApiResult<Void>` — only `success` and `errorMessage` are populated. Possible error messages: `PORTAL_BLOCK_TRANSFER_NOT_FOUND` (generic catch-all for missing parameters, unknown transfer, wrong `blockId`, wrong sender e-mail or any unexpected exception — deliberately ambiguous so the server does not leak which field was wrong); `PORTAL_DOWNLOAD_TRANSFER_EXPIRED` (the transfer has already expired, status `DISABLED`); `PORTAL_BLOCK_TRANSFER_ALREADY_BLOCKED` (the transfer was blocked previously); `PORTAL_BLOCK_TRANSFER_ERROR` (the database update failed). The call is idempotent: a second call with the same parameters returns `PORTAL_BLOCK_TRANSFER_ALREADY_BLOCKED`, the underlying state is not changed and no extra notification mail is sent. Side-effect ordering: the DB update happens before the notification mail is sent, so if the mail server is unreachable the transfer is still blocked and only the mail failure is logged.

9. Device registration

Endpoints for registering an external device (a 3rd party system) with the FileCap server so it can call the API with its own per-device API key. The flow is: register → the server sends a verification code by mail → verify → caller receives a JWT / device token.

9.1 Register device

POST {{baseUrl}}/FileCap/api/user/devices/register

Content-Type: application/json

Initiate device registration. The server e-mails a verification code to emailAddress.

Request body

```
{
  "emailAddress": "kees@example.com",
  "apiKey": "HKADASD68768768ASDASDASDAD",
  "deviceName": "My integration server",
  "lang": "NL"
}
```

Response

ApiResponse<Void>.

Notes

- Rate-limited (RegisterDevice).

9.2 Verify device

POST {{baseUrl}}/FileCap/api/user/devices/verify

Content-Type: application/json

Exchange the e-mailed verification code for a JWT / device token.

Request body

```
{
  "email": "kees@example.com",
  "verificationCode": "123456"
}
```

Response

ApiResponse<String> — the issued token.

Notes

- Rate-limited (VerifyDevice).