

[Collapse All](#) | [Expand All](#)

[Home](#)

› [Archived](#)

› [Acknowledgements](#)

[Blog and News](#)

▼ [Documentation v2-0-alpha](#)

[Installation - Linux Source 2-0-Alpha-from-git](#)

[Development upgrade notes to 2-0](#)

[Admin - Configuration directives v2-0](#)

[Admin - Reference guide v2-0](#)

**RESTful API v2-0**

[Language handling and localisation reference v2-0](#)

[Detailed feature list v2-0](#)

› [Documentation v1-6](#)

› [Documentation v1-5](#)

› [Documentation v1-1](#)

› [Download](#)

› [Developer](#)

[Existing installations](#)

[Feature requests](#)

[Online Demo](#)

› [Project organisation](#)

[Roadmap](#)

› [Release Schedule](#)

[Release status and life cycle](#)

› [Reviews and prototypes](#)

[Support and Mailinglists](#)

[Twitter](#)

[FileSender's workflows](#)

[Installation - Linux Source 2-0-Alpha-from-svn](#)

## RESTful API v2-0

Version 7, last updated by [meijer](#) at over 1 year

### Endpoints

All data sent to the REST server (except if mentionned) must be valid JSON with a `Content-Type: application/json`.

All data returned by the REST server (except if mentionned) will be JSON with a `Content-Type: application/json`, except for `JSONP GET` requests, in which case a `Content-Type: text/javascript` is returned.

### Transfer

#### GET /transfer

List of user available transfers (same as `GET /transfer/@me`).

Accessible to : [users](#).

Returns an `array of Transfer`, see [Transfer](#).

#### GET /transfer/@all

List of all available transfers (admin only).

Accessible to : [admin](#).

Returns an `array of Transfer`, see [Transfer](#).

#### GET /transfer/{id}

Get info about a specific [Transfer](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) transfer unique identifier

Returns an `array of Transfer`, see [Transfer](#).

Request example :

```
GET /transfer/42
```

Response example :

```
{
  "id": 42,
  "user_id": "145879@idp.tld",
  "user_email": "foo@bar.tld",
  "subject": "Here is the promised files",
  "message": null,
  "created": {
    "raw": 1421328866,
    "formatted": "15/01/2015"
  },
  "expires": {
    "raw": 1422365666,
    "formatted": "27/01/2015"
  },
  "options": ["email_download_complete", "email_report_on_closing"],
  "files": [
    {
      "id": 25,
      "transfer_id": 42,
      "uid": "7ecab9c0-9abf-eee9-9ab4-00000cc39816",
      "name": "report.doc",
      "size": "102598",
      "sha1": null
    },
    {
      "id": 31,
      "transfer_id": 42,
      "uid": "179558c7-c5be-5428-0b2e-00007ce9c792",
      "name": "funny.ppt",
      "size": "23589654",
      "sha1": null
    }
  ],
  "recipients": [
    {
      "id": 59,
      "transfer_id": 42,
      "token": "0c27af59-72d1-0349-aa59-00000a8076d9",
      "email": "globi@bar.tld",
      "created": {
        "raw": 1421328866,
        "formatted": "15/01/2015"
      },
      "last_activity": null,
      "options": null,
      "download_url": "https://filesender.org/?s=download&token=0c27af59-72d1-0349-aa59-00000a8076d9",
      "errors": []
    }
  ]
}
```

**GET /transfer/{id}/options**

Get the options of a specific [Transfer](#).

Accessible to : [owner](#).

Parameters :

- `{id}` (`integer`) transfer unique identifier

Returns an `array of string`, see [Transfer](#).

**GET /transfer/{id}/auditlog**

Get the [Audit logs](#) of a specific [Transfer](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) transfer unique identifier

Returns an `array of Audit log`, see [Audit log](#).

Returned array will be empty if the audit logging is turned off.

#### GET /transfer/{id}/auditlog/mail

Sends the [Audit logs](#) of a specific [Transfer](#) to the current user by email.

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) transfer unique identifier

Returns `true`.

#### POST /transfer

Create a new [Transfer](#) from request (including files and recipients).

Accessible to : [users](#).

Request body fields :

- `files` : (`array of object`) files to be uploaded
  - `name` : (`string`) the file name
  - `size` : (`integer`) the file size in bytes
  - `mime_type` : (`string`) the file MIME type (optionnal), will be used to set the `Content-Type` header for downloaders so that their browsers handle the file better
  - `cid` : (`string`) an optionnal client side identifier which will be returned in the response as a property of the corresponding created [File](#), it may help the client to match back without having to use name and size
- `recipients` : (`array of string`) recipients email addresses
- `options` : (`array of string`) options (values defined in `classes/constants/TransferOptions.class.php`), under [Guest](#) authentication it will be ignored and the `Guest transfer_options` will be used
- `expires` : (`integer`) wanted expiry date as a UNIX timestamp, MUST respect `max_transfer_days_valid` configuration parameter
- `from` : (`string`) choosen sender email address, under [Service Provider](#) authentication it MUST be one of the authenticated user email addresses, under [Guest](#) authentication it will be ignored and the `Guest email` will be used and under [Remote Application](#) or [Remote User](#) it will be taken as is with just a format check
- `subject` : (`string`) subject sent to recipients (may be empty)
- `message` : (`string`) message sent to recipients (may be empty)

Returns a [Transfer](#) object.

Returns a `Location` HTTP header giving the path to the new [Transfer](#).

Request example :

```
POST /transfer

{
  "files": [
    {
      "name": "report.doc",
      "size": "102598",
      "mime_type": "application/ms.word",
      "cid": "rnd_341546546545646"
    },
    {
      "name": "funny.ppt",
      "size": "23589654",
      "mime_type": "application/ms.powerpoint",
      "cid": "rnd_67765146157651"
    }
  ],
  "recipients": ["globi@bar.tld"],
  "options": ["email_download_complete", "email_report_on_closing"],
  "expires": 1422365666,
  "from": "foo@bar.tld",
  "subject": "Here is the promised files"
}
```

Response example :

```
{
  "id": 42,
  "user_id": "145879@idp.tld",
  "user_email": "foo@bar.tld",
  "subject": "Here is the promised files",
  "message": null,
  "created": {
    "raw": 1421328866,
    "formatted": "15/01/2015"
  },
  "expires": {
    "raw": 1422365666,
    "formatted": "27/01/2015"
  },
  "options": ["email_download_complete", "email_report_on_closing"],
  "files": [
    {
      "id": 25,
      "transfer_id": 42,
      "uid": "7ecab9c0-9abf-eee9-9ab4-00000cc39816",
      "name": "report.doc",
      "size": 102598,
      "sha1": null
    },
    {
      "id": 31,
      "transfer_id": 42,
      "uid": "179558c7-c5be-5428-0b2e-00007ce9c792",
      "name": "funny.ppt",
      "size": 23589654,
      "sha1": null
    }
  ],
  "recipients": [
    {
      "id": 59,
      "transfer_id": 42,
      "token": "0c27af59-72d1-0349-aa59-00000a8076d9",
      "email": "globi@bar.tld",
      "created": {
        "raw": 1421328866,
        "formatted": "15/01/2015"
      },
      "last_activity": null,
      "options": null,
      "download_url": "https://filesender.org/?s=download&token=0c27af59-72d1-0349-aa59-00000a8076d9",
      "errors": []
    }
  ]
}
```

**POST /transfer/{id}/recipient**

Add a recipient to a specific [Transfer](#).

Accessible to : [owner](#).

Parameters :

- `{id}` (`integer`) transfer unique identifier

Request body fields :

- `recipient` : (`string`) recipient's email address

Returns a [Recipient](#) object.

Returns a `Location` HTTP header giving the path to the new [Recipient](#).

**PUT /transfer/{id}**

Update a specific [Transfer](#).

Accessible to : [owner](#); [chunk upload security](#) fallback accepted for `complete` property setting only.

Parameters :

- `id` (`integer`) transfer unique identifier

Request body fields :

- `complete` : (`boolean`) if set to `true` this signals the server that every file has been uploaded and that the [Transfer](#) should be made available and sent to [Recipients](#) (optionnal)
- `closed` : (`boolean`) if set to `true` this signals the server that the [Transfer](#) should be closed (optionnal)
- `remind` : (`boolean`) if set to `true` this asks the server to send a reminder of the [Transfer](#) availability to its [Recipients](#) (optionnal)

Returns `true`.

Request example :

```
PUT /transfer/42

{
  "complete": true
}
```

Response example :

```
true
```

#### **DELETE /transfer/{id}**

Close a specific [Transfer](#).

Accessible to : [owner](#); [chunk upload security](#) fallback accepted for early [Transfer](#) statuses (`created`, `started` and `uploading`, but not `available`).

Parameters :

- `id` (`integer`) transfer unique identifier

Returns `true`.

### **File**

#### **GET /file/{id}**

Get info about a specific [File](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) file unique identifier

Returns a [File](#).

#### **POST /file/{id}whole**

Upload file body as a whole (legacy mode).

Accessible to : [owner](#); [chunk upload security](#) fallback accepted.

Parameters :

- `id` (`integer`) file unique identifier

The request content type MUST be `multipart/form-data`, at least a `file` part is expected.

The request body can also contain a field whose name is the same as the server's `session.upload_progress.name` PHP configuration parameter and whose value is randomly choosen to be able to get upload progress informations (see `GET /legacyuploadprogress/{key}`).

Returns the [File](#).

Returns a `Location` HTTP header giving the path to the [File](#).

#### **PUT /file/{id}/chunk{offset}**

Add a chunk to a [File](#)'s body at the given `offset`.

Accessible to : [owner](#), [chunk upload security](#) fallback accepted.

Parameters :

- `id` (`integer`) file unique identifier
- `offset` (`integer`) body offset in bytes

The request content type MUST be `application/octet-stream`.

Several request headers SHOULD be provided so that the server can check validity of sent data :

- `X-Filesender-File-Size` (`integer`) should be equal to the file size
- `X-Filesender-Chunk-Offset` (`integer`) should be equal to `offset`
- `X-Filesender-Chunk-Size` (`integer`) the sent chunk size

The sent chunk size MUST not be over `upload_chunk_size`.

Returns the [File](#).

Request example :

```
PUT /file/27/chunk/25000000
X-Filesender-File-Size: 5987465211
X-Filesender-Chunk-Offset: 25000000
X-Filesender-Chunk-Size: 5000000

<5000000 bytes of data from the file>
```

Response example :

```
{
  "id": 27,
  "transfer_id": 42,
  "uid": "7ecab9c0-9abf-eee9-9ab4-00000ccc39816",
  "name": "big_file.tgz",
  "size": 5987465211,
  "sha1": null
}
```

#### **PUT /file/{id}**

Update a [File](#).

Accessible to : [owner](#), [chunk upload security](#) fallback accepted.

Parameters :

- `id` (`integer`) file unique identifier

Request body fields :

- `complete` : (`boolean`) if set to `true` this signals the server that all chunks for the file have been uploaded



Returns `true`.

#### **DELETE /file/{id}**

Delete a [File](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) file unique identifier

If the related [Transfer](#) still has [Files](#) the recipients will be notified of the [File](#) removal.

If the [Transfer](#) only had the current [File](#) it will be closed.

Returns `true`.

### **Recipient**

#### **GET /recipient/{id}**

Get info about a specific [Recipient](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) recipient unique identifier

Returns a [Recipient](#).

#### **DELETE /recipient/{id}**

Delete a specific [Recipient](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) recipient unique identifier

If the related [Transfer](#) still has [Recipients](#) the recipient will be notified of its removal.

If the [Transfer](#) only had the current [Recipient](#) it will be closed.

Returns `true`.

### **Guest**

#### **GET /guest**

List of user's [Guests](#) (same as `GET /guest/@me`).

Accessible to : [users](#).

Returns an `array of Guest`, see [Guest](#).

#### **GET /guest/@all**

List of all available [Guests](#).

Accessible to : [admin](#).

Returns an `array of Guest`, see [Guest](#).

#### **GET /guest/{id}**

Get info about a specific [Guest](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) guest unique identifier

Returns a [Guest](#).

#### POST/guest

Create a new [Guest](#) from request.

Accessible to : [users](#).

Request body fields :

- `recipient` : (`string`) recipient email addresse
- `from` : (`string`) choosen sender email address, under [Service Provider](#) authentication it MUST be one of the authenticated user email addresses and under [Remote Application](#) or [Remote User](#) it will be taken as is with just a format check
- `subject` : (`string`) subject sent to recipient (may be empty)
- `message` : (`string`) message sent to recipient (may be empty)
- `options` : (`array of string`) options (values defined in `classes/constants/GuestOptions.class.php`)
- `transfer_options` : (`array of string`) created [Transfer](#) options (values defined in `classes/constants/TransferOptions.class.php`)
- `expires` : (`integer`) wanted expiry date as a UNIX timestamp, MUST respect `max_guest_days_valid` configuration parameter

This will send an email to the created [Guest](#).

Returns a [Guest](#) object.

Returns a `Location` HTTP header giving the path to the new [Guest](#).

Request example :

```
POST /guest

{
  "recipient": "globi@bar.tld",
  "options": ["valid_only_one_time"],
  "transfer_options": ["email_download_complete", "email_report_on_closing"]
,
  "expires": 1422365666,
  "from": "foo@bar.tld",
  "subject": "Send me a file :)"
}
```

Response example :

```
{
  "id": 31,
  "user_id": "145879@idp.tld",
  "user_email": "foo@bar.tld",
  "email": "globi@bar.tld",
  "token": "7a48ff78-442f-9449-c292-00002399a22f",
  "transfer_count": 0,
  "subject": "Send me a file :)",
  "message": null,
  "options": ["valid_only_one_time"],
  "transfer_options": ["email_download_complete", "email_report_on_closing"]
,
  "created": {
    "raw": 1421328866,
    "formatted": "15/01/2015"
  },
  "expires": {
    "raw": 1422365666,
    "formatted": "27/01/2015"
  },
  "errors": []
}
```

**PUT /guest/{id}**

Update a specific [Guest](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) transfer unique identifier

Request body fields :

- `remind` (`boolean`) If set to `true` this asks the server to send a reminder to the [Guest](#)

Returns `true`.

Request example :

```
PUT /transfer/42

{
  "remind": true
}
```

Response example :

```
true
```

**DELETE /guest/{id}**

Delete a [Guest](#).

Accessible to : [owner](#).

Parameters :

- `id` (`integer`) recipient unique identifier

The [Guest](#) will be notified.

Returns `true`.

**User****GET /user/@me/frequent\_recipients**

Get a list of frequent recipients of the current user.

Accessible to : [users](#).

This endpoint can filter its response with the `filterOp[contains]` filter, example : `GET /user/@me/frequent_recipients?filterOp[contains]=foo`.

Returns an `array of string`.

#### **GET /user/{id}/frequent\_recipients**

Get a list of frequent recipients of a specific user.

Parameters :

- `id` (`string`) user unique identifier

Accessible to : user itself, [admin](#).

This endpoint can filter its response with the `filterOp[contains]` filter, example : `GET /user/user_uid/frequent_recipients?filterOp[contains]=foo`.

Returns an `array of string`.

#### **PUT /user**

Set current user preferences.

Accessible to : [users](#).

Request body fields :

- `lang` : (`string`) lang code of user's preferred language

Returns `true`.

#### **PUT /user/{id}**

Set user preferences

Accessible to : user itself, [admin](#).

Parameters :

- `id` (`string`) user unique identifier

Request body fields :

- `lang` : (`string`) lang code of user's preferred language

Returns `true`.

### **Misc endpoints**

#### **GET /info**

Get informations about the instance.

Accessible to : public.

Returns an `object` whose properties are informations about the FileSender instance.

The set of returned informations depends on the `disclose` configuration parameter.

At the very least the `url` property is set and contains the URL of the FileSender instance.

This endpoint may be queried in order to know useful configuration parameters for uploading like `upload_chunk_size`, `default_transfer_days_valid` or more when they are disclosed.

#### **PUT /config**

#### **GET /legacyuploadprogress/{key}**

Get information about a legacy (whole file) upload progress.

Parameters :

- `key` ( `string` ) the file's chosen upload tracking random (see `POST /file/{id}/whole` )

Returns an `object` :

- `error` ( `integer` ) not 0 if something bad happened
- `done` ( `boolean` ) is the upload complete
- `start_time` ( `integer` ) UNIX timestamp of when the server started receiving data
- `bytes_processed` ( `integer` ) number of bytes already received

#### GET /lang

Get translations (merged between default language, configured language and maybe user preference or browser language).

Returns an `object` whose keys are string identifiers and values are the translation related to them.

#### GET /echo

Echoes back info about your request, useful for testing.

Parameters : any

Returns `object` :

- `args` : ( `array of string` ) the decoded URL tokens
- `request` : ( `mixed` ) the decoded request body
- `user` : ( `string` ) authenticated user unique identifier
- `auth` : ( `object` )
  - `remote` : ( `boolean` ) is the detected authentication a [Remote Application](#) or a [Remote User](#)
  - `attr` : ( `object` ) authentication attributes

## Errors

Errors (or exceptions) are returned in a structured format with the following fields : \* `message` : ( `string` ) translatable id telling what happened \* `uid` : ( `string` ) unique error identifier, is included in the logs \* `details` : ( `array of string` ) details about the error (may be empty, null or even not set)

All possible errors messages are not listed here but can be found in the files under `classes/exceptions/`.

Common errors to all endpoints are : \* `rest_authentication_required` : need an authenticated user \* `rest_ownership_required` : need ownership of target resource (admin owns all) \* `rest_admin_required` : need an authenticated admin \* `rest_missing_parameter` : an URL token is missing \* `rest_bad_parameter` : an URL token does not have the expected type

Example :

```
{
  "message": "rest_authentication_required",
  "uid": "57gf523gsdfg3"
}
```

## Objects fields definition

### Transfer

- `id` : ( `integer` ) transfer unique identifier
- `user_id` : ( `string` ) uploader / guest creator unique identifier
- `user_email` : ( `string` ) uploader (user or guest) email

- `subject` : ( `string` ) subject sent to recipients
- `message` : ( `string` ) message sent to recipients
- `created` : ( `Date` ) creation [Date](#)
- `expires` : ( `Date` ) expiry [Date](#)
- `options` : ( `array of string` ) options (values defined in `classes/constants/TransferOptions.class.php`)
- `files` : ( `array of File` ) see [File](#)
- `recipients` : ( `array of Recipient` ) see [Recipient](#)

#### File

- `id` : ( `integer` ) file unique identifier
- `transfer_id` : ( `integer` ) related [Transfer](#) unique identifier
- `uid` : ( `string` ) random unique identifier
- `name` : ( `string` ) file original name
- `size` : ( `integer` ) file size in bytes
- `sha1` : ( `string` ) file hash, unused, unset @TODO should we remove it until we have a viable technical solution ?

#### Recipient

- `id` : ( `integer` ) recipient unique identifier
- `transfer_id` : ( `integer` ) related [Transfer](#) unique identifier
- `token` : ( `string` ) download unique random token, used for authentication
- `email` : ( `string` ) recipient email address
- `created` : ( `Date` ) creation [Date](#)
- `last_activity` : ( `Date` ) last activity [Date](#)
- `options` : ( `null` ) unused, unset @TODO what was it for ?
- `download_url` : ( `string` ) Download page URL
- `errors` : ( `array of Tracking error` ) see [Tracking error](#)

#### Guest

- `id` : ( `integer` ) guest unique identifier
- `user_id` : ( `string` ) guest creator unique identifier
- `user_email` : ( `string` ) guest creator email
- `email` : ( `string` ) guest email address
- `token` : ( `string` ) guest unique random token, used for authentication
- `transfer_count` : ( `integer` ) number of transfers uploaded
- `subject` : ( `string` ) subject sent to guest
- `message` : ( `string` ) message sent to guest
- `options` : ( `array of string` ) options (values defined in `classes/constants/GuestOptions.class.php`)
- `transfer_options` : ( `array of string` ) created transfers options (values defined in `classes/constants/TransferOptions.class.php`)
- `created` : ( `Date` ) creation [Date](#)
- `expires` : ( `Date` ) expiry [Date](#)
- `errors` : ( `array of Tracking error` ) see [Tracking error](#)

#### Tracking error

- `type` : ( `string` ) error type (values defined in `classes/constants/TrackingEventTypes.class.php`)
- `date` : ( `Date` ) error [Date](#)
- `details` : ( `string` ) technical details about what happened

#### Audit log

- `date` : ( `Date` ) event [Date](#)

- `event` : ( `string` ) event type (values defined in `classes/constants/LogEventTypes.class.php`)
- `author` : ( `object` ) information about who did the action
  - `type` : ( `string` ) author type, can be `User`, `Recipient` or `Guest`
  - `id` : ( `string` ) author's unique identifier
  - `ip` : ( `string` ) author's IP address
  - `email` : ( `string` ) author's email address (may not be provided)
- `target` : ( `object` ) information about the target of the event
  - `type` : ( `string` ) target type, can be `Transfer`, `File` or `Recipient`
  - `id` : ( `string` ) target's unique identifier
  - `name` : ( `string` ) original name if `type` is `File` (not provided otherwise)
  - `size` : ( `integer` ) size in bytes if `type` is `File` (not provided otherwise)
  - `email` : ( `string` ) email address if `type` is `Recipient` (not provided otherwise)

## Date

- `raw` : ( `integer` ) unix timestamp
- `formatted` : ( `string` ) human readable version according to lang configuration

## Authentication

### Levels of authentication

#### User

Anybody authenticated through a Service provider, a [Remote User](#) or a [Remote Application](#) providing the `remote_user` argument.

Authentication through using [Recipient](#) token is not considered as being a user.

#### Owner

An authenticated user who is owner of the current resource ([Admin](#) is considered as owner of everything).

#### Admin

An authenticated user whose `id` is listed in the `admin` configuration parameter, or a [Remote Application](#) whose configuration include the `isAdmin` entry set to `true`.

### Authentication evaluation workflow

The servers look up for authentication in this order and stops on the first detected one :

1. [Guest token](#)
2. [Service provider](#)
3. [Remote application](#) (if enabled)
4. [Remote user](#) (if enabled)

#### Guest token

This mode requires a `vid` URL argument corresponding to an existing [Guest](#).

This mode has restricted rights in most of the application.

#### Service Provider

This mode requires the configured service provider to return a valid authentication.

#### Remote application

This mode is intended towards giving access to applications FileSender can fully trust, like a big file generating script of yours that need to send its file to recipients.

This mode expects a [signed request](#) and a `remote_application` URL argument.

Additionally if a `remote_user` argument is provided all operations will be conducted with this identity.

### Remote user

This mode is intended to give users the possibility to create a trust between another of their application and their "account" on FileSender, like a mail client being able to send big attachments through FileSender.

This mode expects a [signed request](#) and a `remote_user` URL argument.

### Signed request

A signed request is formed by adding several informations to a request to be made :

- a `timestamp` URL argument containing a UNIX timestamp of the current date and time, this ensures the request will not be replayed later as FileSender will reject any signed requests older than 15 seconds. In case of uploads with a slow connexion you may have to put this in the future.
- a `remote_application` URL argument in the case of a [Remote Application](#) authentication, this contains the name of the application, shared between FileSender and the application sending the request
- a `remote_user` URL argument in the case of a [Remote User](#) authentication, this contains the unique user identifier, shared between FileSender and the application sending the request
- a `signature` URL argument, this is a SHA1 HMAC signature of the prepared request (see below) with :
  - the shared application secret in case of a [Remote Application](#) authentication
  - the shared user secret in case of a [Remote User](#) authentication

### Preparing a request for signing

To prepare a request for signing one must concatenate following parts, in order, using `&` separator :

- lowercased HTTP method : `get`, `post`, `put` or `delete`
- URL with alphabetically ordered arguments, without scheme or signature argument, example `filesender.org/rest.php/transfer/@me?remote_application=foo&remote_user=user_id&timestamp=1234567890`
- request body as a string (if there is one)

### Examples

Remote application request to get a user's transfers list : `get&filesender.org/rest.php/transfer/@me?remote_application=foo&remote_user=user_id&timestamp=1234567890`

Remote user request to upload the chunk at offset 2597874 in file with id 42 :  
`put&filesender.org/rest.php/file/42/chunk/2597874?remote_user=user_id&timestamp=1234567890&<chunk_data>`

Sample PHP code :



```

$base = 'https://filesender.org/rest.php';

$method = 'POST';

$resource = '/guest';

$data = array(
    'recipient' => 'foo@bar.tld',
    'from' => 'user_email@domain.tld',
    'subject' => 'For the project report',
    'message' => 'Come ! Send me the file !',
    'options' => array('can_only_send_to_me'),
    'transfer_options' => array(),
    'expires' => time() + 7 * 24 * 3600
);

$user = 'user_id';
$secret = 'user_secret';

$args = 'remote_user='.$user.'&timestamp='.time();
$body = json_encode($data);

$to_sign = strtolower($method).'&'.preg_replace('`https?://`', '', $base).$resource;
$to_sign .= '?'.$args;
$to_sign .= '&'.$body;

$url = $base.$path.'?'.$args.'&signature='.hash_hmac('sha1', $to_sign, $secret);

// Send request with curl with $method to $url and $body content

```

#### More information

You will find an sample client under `scripts/client/FilesenderRestClient.class.php`.

#### Chunk upload security

As when authenticated with a [Service Provider](#) the session can be broken during upload depending on the [Service Provider](#) configuration (absolute session lifetime ...) it is possible to enable the key based chunk upload security.

This mode is enabled by setting the `chunk_upload_security` configuration parameter to `key` `@TODO{ need default value }`.

In this mode you may send the chunk's related [File's](#) `uid` property along with the request as the `key` URL argument.

In a [Transfer](#) context any `uid` property from any [File](#) related to the [Transfer](#) will be accepted.

Note that this mechanism is a fallback in the case your client lost all other kind of authentication, this is not necessary under [Remote Application](#) or [Remote User](#) as they are always valid as long as the [signature](#) is right.

#### The upload process

To upload files one must follow a set of operations in order.

1. Create the transfer using `POST /transfer`, this gives back data about the transfer, especially file identifiers to be used later
2. Send file data using `PUT /file/{file_id}/chunk/{chunk_offset}`, chunks don't need to be in order, it is possible to upload several chunks at the same time by making requests in parallel
3. Once a file data has been sent signal file completion using `PUT /file/{file_id}` with payload `{"complete":true}`
4. Once all file completions signals have been sent signal transfer completion using `PUT /transfer/{transfer_id}` with payload `{"complete":true}`

Viewing version 7 created  
over 1 year by meijer

Rollback to this version

Recent Versions

Choose two versions to  
compare, or click the link to  
view it.

- ☒ 8. 10 months ago by  
jornane
- ☒ 7. over 1 year ago by  
meijer
- ☐ 6. over 2 years ago by  
etiennemeleard
- ☐ 5. over 2 years ago by  
etiennemeleard
- ☐ 4. over 2 years ago by  
etiennemeleard
- ☐ 3. over 2 years ago by  
etiennemeleard
- ☐ 2. over 2 years ago by  
etiennemeleard
- ☐ 1. over 2 years ago by  
etiennemeleard

Show Diff

Clear All

History Key

- New Content
- ~~Removed content~~