Development & Integration Guide

https://sags.digital

About this guide

This guide will help you with integrating sags.digital into your software systems. It is our top priority to make the integration of the system as easy and comfortable as possible for you. If you have any questions, please feel free to contact our support at any time.

If you are missing an integration function or have an idea how we can make your life easier, please do not hesitate to contact us.

Inhalt

Basic terminology

It is important to have a clear understanding of the following terms, used within this guide:

- **Single Message**
  One of two options for creating contents in sags.digital is to create a single video message, wherein other people's comments appear as "on the timeline of the video". We call this type of content "Single Message".

- **Thread**
  One of two options for creating contents in sags.digital is to create a whole discussion, with people communicating one after another (chronologically). We call this type of content "Thread".

- **Sags.digital content**
  This is a thread or a single message.

- **Folder**
  Folders can be used to organize sags.digital contents, just like a folder in your computer's filesystem. Folders might also contain subfolders.

- **Reply Message**
  When a user adds a comment to a single message, we call this comment a "reply message". Reply messages are always related to a certain point in time inside the main video (single message).

- **Thread Message**
  When a user adds a new message inside a thread, we call these messages "thread messages".

- **Internal Share**
  When a user shares something with another registered user (inside his customer account), then this is called "internal share".

- **External Share**
  When a user shares something with an external person (not registered) via a secret share link, then this is called "external share". An external share is always bound to a concrete name – so it cannot / should not be used to share something to a group of people.

- **Public share link**
  All shareable contents in sags.digital have two "public share links" (one for readonly access, one with rights for commenting). Those links are like the secret links used for "external shares", but they are not bound to a concrete name. Instead, users opening the link have to first enter their name to get access to the contents. This makes it easy to share content with a group of people, but you have to keep in mind that entered usernames might be maliciously wrong.

- **Guest video creation**
  A feature of sags.digital which allows unknown external users to create videos inside your account. For example, you might enable your website visitors to create sales requests by recording a sags.digital video.

- **Join page**
  When using guest video creation, the user might have to enter information (e.g. name or email) prior to recording the video. This information is to be entered on a "join page".
  Also, if a user opens an external share link (e.g. to view a video that has been shared with him), the system might show a "join page" and ask the user for its email address. This way, the user can get informed about updates to the video that occur after he has initially watched it.

Integration of sags.digital contents into other websites (Embedding)

To embed sags.digital contents into arbitrary websites, follow these steps:

1. Get the public share link (read-only) for your content.
2. Add a "/embed" to that URL. We call this the "embed URL".

   Example:
   https://app.sags.digital/public-share/thread/62ddee18-7d66-4ff9-a378-b923368e999e
   will become
   https://app.sags.digital/public-share/thread/62ddee18-7d66-4ff9-a378-b923368e999e/embed

3. Use the following code to place an iframe onto your website:
   (Please replace "YOUR_URL" with the embed url from the previous step.)

```
<iframe width="560" height="315" frameborder="0" allow="accelerometer;
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture;
camera; microphone; fullscreen" allowfullscreen src="YOUR_URL"></iframe>
```

4. The website should now show the video player.

Feel free to customize the iframe attributes to your needs and use css to style the iframe. The iframe may also be adjusted to the screen's width/height and all its contents should behave in a responsive manner.

Optionally, once the iframe is embedded, you are also able to get to know what happens inside the iframe programmatically. See chapter "Receive events as parent page" for more information.
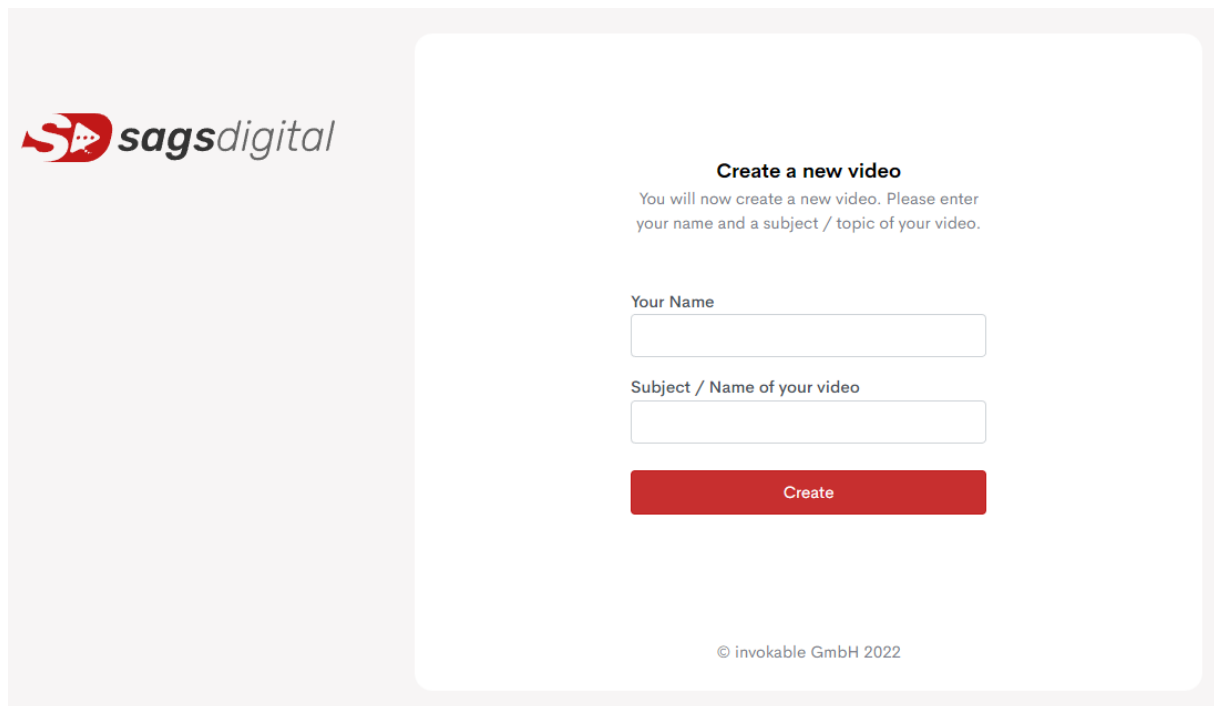
Let external users create video contents (Guest video creation)

You might not only want to show contents to users of your system, but also enable them to create new contents. Then you should follow these steps:

1. Open your sags.digital account and create a new folder. Inside that folder, contents of external users will be stored. (Hint: If you expect a big number of incoming videos, it might be best to create a separate user inside your customer account only for storing those videos.) In this example, we will call the folder "Incoming sales requests", as we want to collect new sales requests from our website. Do not go into that folder, but wait for the folder to appear in the folder explorer on the left.
2. In the folder explorer on the left side, open the options menu of the folder and choose "Guest video creation".
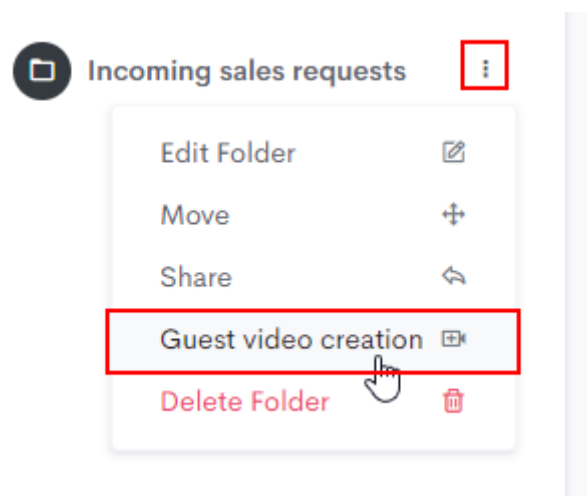
3. If guest video creation is not enabled, enable it by clicking on "Activate".
4. You will be provided with a link. When users open that link, they will be able to enter their name and a subject for the new video. Afterwards the recorder will open and a new video is created inside your folder. Of course, you will receive a notification about that new video inside your account.

5. In the guest video creation window, you also have the option to specify advanced settings:
    a. You can specify, whether a single video or a whole discussion should be created.
    b. You can specify, whether an additional external share link should be sent to an arbitrary email address (e.g. someone in your team, or even to a ticket system). If you enter an email address, you will also have to enter a name for that external share link. E.g. if you enter "peter@yourcompany.com" and "Peter", then Peter will always be invited to each new discussion. After entering an email address you can also specify whether the receiver of the mail is a ticket system. Then, we will send a very condensed text email to that mail address only.
    c. You can specify a webhook URL, which is invoked after a new video has been created. (see chapter "Getting informed about new external videos").
    d. You can require that the user has to enter an email address (which makes sense if you do not know the user creating the video)
    e. You can customize the appearance of the "join page" (e.g. change the title / description text or the labels of some fields).
    f. You can enter a "success message". This is a message that will be shown to the user after he recorded his video. After recording the user will be shown his newly created video and the user might ask himself "now, what?". So it is a good idea to then show a message telling him, that the video has been successfully recorded and that someone will follow up on his request.

Getting informed about new external videos

Per default, if a new video has been created via guest video creation, the following happens:

- The owner of the folder the video has been created in will receive an email. This mail informs about the new video.

- The owner of the folder the video has been created in will be presented with "unread messages" when opening his sags.digital account.
- The creator of the video (external user) might also receive an email with a secret link (external share) to his video. This way, the user can follow the discussion of the video. The email is only sent if the user provides an email address on the join page.

If you specified that an external share link should be send to an email address:

- An external share (with the specified name) will be created.
- An email is sent to the specified email address, which contains the secret link of the external share. This mail is either sent with a good-looking html mail template, or with a condensed text template (e.g. optimal for processing in ticket systems).

In you specified a webhook URL:

- The webhook URL is invoked after the video has been created. A "POST" request will be issued, which contains the following form-data parameters:
  - **name**
    Name of the video that has been created (subject). As entered on the join page.
  - **id**
    ID of the video that has been created. Might be used to further process the video via the API of sags.digital.
  - **username**
    The name of the user that created the video. As entered on the join page.
  - **email**
    The email address of the user that created the video. As entered on the join page.
  - **url**
    The public share link of the created video (with rights to view & comment).
  - **url_read_only**
    The public share link of the created video (with rights to view only)
  - **url_embed**
    The embed url (read only) which can be used inside iframes
  - **url_deletion**
    When this URL gets opened, the video will be deleted. No additional authorization is required, as the link already includes a unique, random secret.
  - **gvcExtRef**
    External reference as provided as a parameter for the join page (see "")

Guest video creation: Join page parameters

When redirecting a user to a guest video creation link, a join page will appear and the user has to provide information like his name, mail or the subject of the video.

In some cases, the values of the join page are known to the system redirecting the user. The redirecting system might then pre-fill the values of the fields via these query-parameters (appended to the secret guest video creation url):

- **prefillName**
  The value of the "username" field will be prefilled with the value of this query parameter.

- **prefillSubject**
  The value of the "subject" field will be prefilled with the value of this query parameter. Please ensure unique subjects for each video. Otherwise it might be hard for you to find the right video inside the UI of sags.digital.
- **prefillEmail**
  The value of the "email" field will be prefilled with the value of this query parameter. This parameter might even be added if the user is not required to enter his mail.
- **recorderOnly**
  If set to 1, the application will present an optimized layout for popups or iframes that are only used to record a video. After the recording ends, there is a centered big status that shows the upload progress. After uploading completes, this status will show "Please wait" forever, until the popup/iframe is closed/removed by the parent application.
- **recorderOnlyShowCloseOnEnd**
  If recorderOnly and recorderOnlyShowCloseOnEnd are both set to 1, the application will show a "Close this window. Recording has been saved successfully" message instead of "Please wait" after the recording has been submitted.
- **jsonResponse**
  If set to 1, the application will respond with a json string which contains all relevant information for recording a video and using it later. This requires that all mandatory fields are prefilled and only works for a guest video creation type of "Create single video" (not for discussions).
  The json provides the following information:
  - recordingUrl
    This URL will show the recorder to record the message. You can redirect a user to this URL.
  - statusPollUrl
    This URL will answer with a json string and tells, whether the message is waiting for a recording (status=new) or whether it has been recorded and is available for viewing (status=available).
  - messageInfo
    Object that contains the exact same information as provided by the webhook callback (see "Getting informed about new external videos")

Providing prefill values via query parameters is completely optional. However, if you provide enough information (that is, all fields are pre-filled), the sags.digital system will just accept your values and skip the join page completely. The user is then directly redirected to the recorder.

You will also have the need to identify the user or the context the user is coming from: Imagine the user is logged in to a helpdesk system and now wants to create a new comment inside an existing ticket. The new comment should be a sags.digital video. Then the helpdesk system will redirect the user to a guest video creation link, After the user recorded the video, it has to be appended to the ticket. The admins have to options here: either an email is sent to the helpdesk system or the helpdesk system will receive the webhook. However, the helpdesk system now needs to map the received email or the received webhook to the ticket and the user that created it.

To accomplish this, when redirecting to the guest video creation link, an additional parameter may be passed to sags.digital:

- **gvcExtRef**
  The "guest video creation external reference" is an arbitrary string (max. 255 chars), which is used to identify the user / context of the user. It will not be processed by the sags.digital system in any way, but it will be passed back via the external share email and/or via the webhook.
  Please be aware, that query parameters may be maliciously manipulated on the client side. Please use unguessable ids to identify your users or use symmetric encryption to encrypt important data before passing it as a gvcExtRef parameter.

In our example, the helpdesk system might create a unique secret id, which represents the logged in user and the ticket. When the helpdesk system receives the mail or the webhook, it is able to e.g. create a new ticket comment inside the correct ticket on behalf of the logged-in user containing the iframe embedded video, or just the public share link to the video.

Receive events as parent page

Once you have successfully embedded sags.digital as an iframe, or you opened a guest video creation page in a popup / new tab and you have a reference to that window, you can listen to what is happening on the sags.digital page programmatically. We send events via "postMessage" to window.opener and window.parent, if applicable.

```
window.addEventListener("message", (event) => {
  if (event.origin !== "https://app.sags.digital")
    return;

  console.log(event);
}, false);
```

This code will console.log all events that are received. There are a lot of events sent to the parent frame (e.g. recording started/paused/resumed/ended, uploading finished, player playing, player timechanges, …).

For a lot of integrations, the most important event might be "**recorder-uploading-finished**" which indicates, that a recording has been recorded, stopped and successfully uploaded to the server. Typically this is the event that is used by third party applications to close the recorder window. It also carries all the data, that you would also be able to receive via the callbackUrl-webhook on guest video creation pages (see chapter "Getting informed about new external videos" for a description of these parameters).

Usage of the API

We are currently working on an external API for our customers. You can expect it to be available shortly, as our whole frontend UI is using the API already. We will just have to make it publically available (e.g. offering login via API and produce a good developer documentation). Then you will be able to use each feature of sags.digital via our API.