

FAQ

Last Modified on 12/23/2024 6:56 pm IST

In this article we cover common questions when integrating with the Kaltura Reach API.

Retrieving jobs

Do you send a notification when a new REACH job is available for us?

Kaltura does not send notifications to REACH vendors for new available jobs.

REACH vendors need to poll the Kaltura API and more specifically use the `getJobs` API call to verify if there are any new jobs available and to retrieve the details of these new jobs.

Do you have recommendations on what we should pass in for the `ClientTag` value if we do not use a client library?

If you are using direct API calls, you can replace the client library version by your vendor application version. In other words, the `<default clientTag>` can consist of the Vendor application Version Number. Example: `'app:18-11-11_vendorName_12345'`

Then each API request should include the `clientTag` parameter within the request body (the `clientTag` parameter that will include the vendor Application Version).

As a vendor should we use the list API endpoint or the `getJobs` API endpoint?

As a vendor you should be only using `getJobs`.

The list API endpoint is for REACH users only.

Note that when you use `getJobs` you should also add

`responseProfile[systemName]=reach_vendor` in the `getJobs` call to ensure you retrieve all the Reach profile information as well.

Can we filter jobs returned by the `getJobs` API endpoint?

The `getJobs` called was hardcoded to always returning only pending jobs (`status=1`).

Does the "getJobs" endpoint support any sort of partitioning or sharding?

Although we do not support partitioning or sharding, you can use the paging available in the `getJobs` query, add the `pager[objectType]=KalturaFilterPager` parameter. By default paging sends back 30 objects, and you can increase it to 500, ie. `pager[pageSize]=500`.

To go through the pages, you use the `pager[pageIndex]` parameter.

Can you list captionAssets?

It is not possible to list captionAssets, it is restricted for security reasons.

Are task IDs globally unique across Kaltura SaaS regions?

The task IDs are unique within a single SaaS region (ie. US or EU).

In other words task IDs may overlap between SaaS regions.


Reach Order management


How do we know if a Reach end-user cancelled a job?

Reach jobs can be in different status which is set by the Reach vendor, most importantly, when a Reach job is first created it is in PENDING status, next:

- for VOD jobs, those are moved to PROCESSING by the vendor, and then it is moved to READY by the vendor,
- for Live jobs, those are first moved to SCHEDULED by the vendor, then moved to PROCESSING, and finally they are moved to READY by the vendor

Reach end-users can only cancel jobs which are in PENDING status, they are not allowed to cancel jobs in PROCESSING status. Additionally, for Live jobs, end-users can cancel jobs as long as they respect the `minimalRefundTime` as set in the catalog item, this is also true for jobs in SCHEDULED state.

 Reach vendors do not necessarily need to implement a specific logic for VOD jobs but they do need to implement a check for Live jobs.

 When an end-user cancels a job and the Reach vendor calls the API:

- the job is no longer returned in the job list,
- the update task call is responded to with `ENTRY_VENDOR_TASK_NOT_FOUND` (see example in [Appendix](#)).

We recommend the following:

- supposing a task was in PENDING,
 - for VOD assets, the vendor must first move a task to PROCESSING before starting processing or returning deliverables

- for Live assets, the vendor must first move a task to SCHEDULED
- for live jobs which are already in SCHEDULED the vendor must run a check when the `minimalRefundTime` expires:
 - if the job is still present in the job list and/or if the task returns the task details then keep the job and task are considered valid
 - if the job is no longer present in the job list and/or if the task returns `ENTRY_VENDOR_TASK_NOT_FOUND` then the job is considered cancelled

By following the above recommendations, Reach vendors will be able to handle job and task cancellations.

Can we lock the experience down to specific orders? for example English language captions only?

That will be default behavior. When a Kaltura customer chooses a vendor, he will choose to have what we call catalog items: these catalog items will be enabled on one of his REACH profiles and the REACH profile will have a certain amount of credits associated to it.

The catalog items represent a specific service from the vendor, for example [English Human captioning 5 Business Days Turn Around Time] or [English to French translation 4 Days Turn Around Time].

When accessing the order screen, the users will only see the services enabled on the REACH profile based on what they have chosen to purchase. And they will be able to place orders as long as they have credits available.

Once vendor integration is completed we jointly define and then create multiple catalog items for the various services the vendor provides.

Do you support resubmissions?

Yes, this setting is not on the client REACH profile, instead it is enabled on the catalog item which represents the service you deliver as a vendor.

For example if you have a service for English machine captioning in 2 hours Turn around time, we can also allow or not allow resubmissions for this service.

We only allow resubmissions on Machine captions (not on human captions).

Should we stop delivering live captions after the Live session end time?

Customer pays by the original order, we do not have a way to calculate cost after the event is finished.

Hence, basically the Vendor should cut off the service when the original scheduled time slot of the live event ends.

You may choose to continue but the expected behavior is that captions will stop hence the recommendation to Reach end-users to reserve their Reach order with 15 minutes buffer before and after the scheduled time of the event.

Error handling

Do you propagate the `errDescription` to the user?

We currently do not provide the `errDescription` to the user but we are currently looking into exposing it.

What happens if processing takes more time than the expected Turn around time, or if my `accessKey` is expired?

Usually if processing takes more time than the expected Turn around time, the `accessKey` will expire. You will receive an `Expired access key` error message on the API calls which use the `accessKey`.

Here is an XML example of the error message:

```
<errDescription>Expired access key</errDescription>
<accessKey>djj8MTQ5MjMwMXwFAiOZCiI_ITZso6rF16cawjY6Zvy_Fw70g5zHFCd49v0Og8Z2QaChvHprVOhHR5g-4iHcgG49AvkrSC_r
```

In such cases you need to renew the access key using <https://developer.kaltura.com/api-docs/service/entryVendorTask/action/extendAccessKey>

What happens if I try to extend an `accessKey` on a non processing task?

If you try to extend an `accessKey` on a non processing task, you will receive the following error message:

```
Extending accessKey for non processing task is not allowed (CANNOT_EXTEND_ACCESS_KEY)
```

Here is JSON response:

```
{'code': 'CANNOT_EXTEND_ACCESS_KEY',
 'message': 'Extending accessKey for non processing task is not allowed',
 'objectType': 'KalturaAPIException',
 'args': []}
```

What should we do if the SRT used for translation has blocks with only 1 millisecond of duration?

For example if an SRT file has the following:

```
00:07:24,020 --> 00:07:24,021  
a terminar haciendo  
lo que estoy haciendo
```

In such case, you should error the job.

Turn Around Times (TATs)

How should we ensure that TATs are met?

In general we recommend that you have a contingency plan to make sure that jobs are delivered on time:

- for example you could have a logic which ensures that any job that reaches half the expected TAT is automatically raising an alert to ensure it will be manually examined and pushed to the finalization stage.
- and/or also trigger an alert for “stuck” jobs, when jobs are exceeding their expected TAT.

In all cases you should ensure you have a manual process to ensure the jobs are delivered within the expected TATs.

Reach jobs

Do you support speaker identification?

Yes, it can be enabled or disabled on a catalog item.

Do you support forcing the output format?

Yes, we can force the output format (SRT, DXFP or VTT) on the catalog item, or default to the customer’s settings.

Do you support dictionaries?

[For machine captioning](#)

Yes, for machine captioning. It is defined on a REACH profile.

REACH users can define one dictionary per language.

Each dictionary currently supports lists of words or phrases (with a limit of 8000 characters) that are relevant to the specific content.

These words are typically provided in a text-based format, where each word or phrase is separated by a line break.

For human captioning

The Kaltura REACH dictionary feature also allows users to provide additional guidance to human editors and reviewers who work on captioning tasks (Human Captions jobs). This feature is not used for machine-generated captions (ASR).

The "Instructions & Notes" text box displayed on the REACH ordering GUI is relevant only for Human Captions jobs, where human editors and reviewers are involved in captioning tasks.

It is meant to provide guidance and context to the human editors and reviewers, helping them produce accurate and context-aware captions.

The best practice is to use this feature for unique terminology, names, and context-specific information. This is similar to an ASR (Automated Speech Recognition) dictionary but is created using free text.

The purpose is to guide human caption editors on how to handle specific words, phrases, or terms that may not be easily understood without context.

What are the rules and constraints on dictionaries and notes?

We support 8000 characters in each REACH dictionary.

There is no limit of characters per word or words per phrase, the limit is the limit of characters per dictionary.

The dictionary values are in the "dictionary" key value pair of the job details.

Note that the dictionary values being set on a profile, these can be sent along with both machine and human jobs.

We support multiple dictionaries but only one dictionary per language.

For special instructions, the free form text blob is in the "notes" key value pair of the job details.

Special instructions are only sent with human jobs.

Should we process jobs with FlavorAsset.Language == "Undefined" ?

When media is uploaded to Kaltura, the system does not perform any linguistic analysis to determine the language of the content. Therefore, setting the language to "Undefined" by default is a practical choice. Without automated language detection capabilities, there's no efficient way to determine the language immediately upon upload.

Note that vendors are responsible for adding new flavor assets, particularly for tasks like dubbing, in such cases vendors would inherently know the language of the dubbed soundtrack. Therefore, vendors are expected to set the `flavorAsset.language` at this stage.

If most media is uploaded with "Undefined" as the default language, then not processing these would mean missing out on a significant portion of content. Especially in a production environment, it's crucial to ensure that every piece of content is processed as required, regardless of its default language setting.

The user is explicitly defining the `sourceLanguage` when submitting a REACH order, it becomes a clear indication of the language of the media. This explicit definition by the user is more reliable than the default setting of the media's language upon upload.

What should be the accuracy returned to Kaltura?

In Kaltura REACH, the `entryVendorTask[accuracy]` field represents the confidence or quality of the task performed by vendors. Vendors have to be cautious when defining the accuracy value, as it gives content owners an idea of how much they can trust the results. If it's a human job, the accuracy may be more subjective than in a machine job, but both have their challenges. Here's a suggested approach for both:

Human Jobs

1. **Tiered Accuracy Scale:** Divide human tasks into qualitative accuracy levels:

- 90-100%: Excellent - Almost no mistakes, highly polished.
- 80-89%: Good - A few minor mistakes, but generally accurate.
- 70-79%: Fair - Some mistakes but still usable.
- Below 70%: Poor - Numerous mistakes, needs review or rework.

2. **Peer Review:** If resources permit, have another individual review the work to assess accuracy. This second opinion can add a layer of quality control.

3. **Feedback Loop:** Incorporate feedback from the client. If a client constantly returns a task for revisions or identifies mistakes, it's an indication that the accuracy level might be set too high.
4. **Training and Calibration:** Regularly train and calibrate your human workers. As they get feedback and improve, they can better self-evaluate their accuracy.

Machine Jobs

1. **Confidence Score:** Many machine learning models will produce a confidence score. This score can directly be used or transformed to fit into the 0-100% scale of `entryVendorTask[accuracy]`.
2. **Validation Sets:** Use validation datasets to test the machine's output. This is especially important if the task has never been executed by the machine before. If the machine performs at 95% accuracy on the validation set, you can use that as a starting point for the accuracy value.
3. **Iterative Improvement:** As more data becomes available and the model is retrained, it might become more (or sometimes less) accurate. Regularly check the model's performance against validation sets or known ground truths.
4. **Versioning:** If multiple versions of the model exist (due to iterative improvements), keep track of which version produced which results. Some versions might be more accurate than others.

General Considerations

- **Transparency:** Always be transparent about how accuracy values are derived. If the client understands that a "90% accurate" machine transcription still means 1 in 10 words might be wrong, they can set their expectations accordingly.
- **Regular Updates:** Don't set the accuracy once and forget about it. Regularly review and, if necessary, adjust the accuracy values.
- **Qualitative vs. Quantitative:** Remember that accuracy for some tasks can be highly subjective. In such cases, it might be more beneficial to have a qualitative metric or feedback system in addition to the numerical accuracy score.

By systematically and transparently defining accuracy, vendors can instill more trust in their clients and ensure that the results they deliver align with expectations.

Live captioning and translation jobs

Should we verify the audio feed received for Live captions?

Yes, you should do Stereo vs. Mono Testing: In environments where audio channels might cause phase cancellation, it's crucial to verify both stereo and mono outputs to detect potential issues with audio streams.

You should ensure that any anomalies like missing captions or audio phase issues are logged clearly so that the partner is aware of potential causes and can take appropriate actions to rectify the situation.

Partners should have clear fallback instructions on how to handle scenarios like phase inversion, including tools or mechanisms to switch between stereo and mono or swap audio channels.

What is used by Kaltura to ensure live captions provided through websocket are in sync with the content?

Kaltura uses three attributes from the websocket:

- **Start pts** (presentation time stamp) - The first media chunk comes with a pts value (which may come from customer?) and it is added to captions that is sent through websocket, for the purpose of the synchronisation mechanism to be successful, then all successive media chunks come with a pts value as well.
- **Start epoch** - is when (epoch time) the media chunk is received by the Vendor. The vendor should maintain a mapping of `start` to `start_pts`. When pushing live captions to Web socket, the vendor should push the information from the mapping table.
- **Start** - starts from 0 and incrementally augments. 0 is beginning of audio stream that the vendor sees for the live event, it is triggered with the first media chunk received.