Punya Prasad Sapkota

From: John Milner <john.milner@kobotoolbox.org>
Sent: Saturday, May 19, 2018 7:36 PM
To: Punya Prasad Sapkota
Cc: Tino Kreutzer; Kashif Siddiqi
Subject: Re: KoBo API

Hi Punya,

Thank you for your patience! In response to you request, we actually have modified the API to make XLSForm imports easier. We deployed the change yesterday.

I've written some documentation and examples for you at https://hackmd.io/0vmZCV_SRV-Upzk5S8w1Ag#. I'll paste the content below as well, but if the formatting does not come through, please consult the URL.

Best regards,

John

API Examples

I've used `curl` to demonstrate these API calls. You'll notice that all my `curl` commands begin with `curl --silent --user jnm_api:test-for-punya --header 'Accept: application/json'` and end by being piped to `python -m json.tool`. The `python` call simply formats the JSON response to make it easier to read. The `--silent` flag to `curl` is not required, but it makes the output cleaner for the purposes of this demonstration. Basic authentication with `--user username:password` as I've done is fine, but you may also authenticate using a token retrieved at https://kobo.humanitarianresponse.info/token/ by sending the header `Authorization: Token [your-token]`. Including the header `Accept: application/json` for all requests is **highly recommended** to avoid having the API return "browseable" HTML that is not easily parsed.

Importing a form

Uploading the XLSForm

`POST` a new `import` as a multipart form. The necessary parameters are `library=false` and `file`, which cotains the XLSForm:

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' -X POST https://kobo.humanitarianresponse.info/imports/ --form library=false --
form file=@kobo_1701_NW.xlsx | python -m json.tool
{
    "status": "processing",
    "uid": "iGYukk6NVEA64zwMsPtgRD",
    "url": "https://kobo.humanitarianresponse.info/imports/iGYukk6NVEA64zwMsPtgRD/"
}
```

Getting the resulting `asset` UID

The asset UID is given by `messages.created[0].uid` in the JSON response when `GET`ting an `import`. Multiple `GET`s may be required until the server indicates `"status": "complete"` in the response:

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' https://kobo.humanitarianresponse.info/imports/iGYukk6NVEA64zwMsPtgRD/ | python -m
json.tool
{
    "date_created": "2018-05-19T15:21:49.709161Z",
    "messages": {
        "created": [
            {
                "kind": "asset",
                "owner__username": "jnm_api",
                "summary": {
                    "columns": [
                        "type",
                        "name",
                        "label",
                        "required",
                        "select_from_list_name",
                        "constraint_message",
                        "constraint",
                        "default",
                        "appearance",
                        "choice_filter",
                        "hint",
                        "relevant"
                    ],
                    "default_translation": null,
                    "geo": false,
                    "labels": [
                        "\u062c \u0628\u064a\u0627\u0646\u0627\u062a \u062c\u0627\u0645\u0639
\u0627\u0644\u0645\u0639\u0644\u0648\u0645\u0627\u062a",
                        "**\u062c1 \u0631\u0642\u0645
\u0627\u0644\u0627\u0633\u062a\u0628\u064a\u0627\u0646**",
                        "**\u062c2 \u0627\u0644\u062a\u0627\u0631\u064a\u062e
(\u064a\u0648\u0645/\u0634\u0647\u0631/\u0633\u0646\u0629) **",
                        "**\u062c3 \u0631\u0645\u0632 \u062c\u0627\u0645\u0639
\u0627\u0644\u0645\u0639\u0644\u0648\u0645\u0627\u062a **",
                        "**\u062c4 \u062c\u0646\u0633 \u062c\u0627\u0645\u0639
\u0627\u0644\u0645\u0639\u0644\u0648\u0645\u0627\u062a **"
                    ],
                    "languages": [
                        null
                    ],
                    "row_count": 1269
                },
                "uid": "apLBsTJ4JAReiAWQQQBKNZ"
            }
        ]
    },
    "status": "complete",
    "uid": "iGYukk6NVEA64zwMsPtgRD"
}
```

Working with existing forms

Deploying an `asset`

Construct a URL using the `asset` UID: https://kobo.humanitarianresponse.info/assets/[your-asset-UID]/deployment/; then, `POST` with the form parameter `active=true`.

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' -X POST
https://kobo.humanitarianresponse.info/assets/apLBsTJ4JAReiAWQQQBKNZ/deployment/ --form active=true
| python -m json.tool
{
    "active": true,
    "backend": "kobocat",
    "identifier": "https://kc.humanitarianresponse.info/jnm_api/forms/apLBsTJ4JAReiAWQQQBKNZ",
    "version_id": "vh9JnfrRNwVFjJZnSwRYqd"
}
```

Rename a form

Send a `PATCH` to the `asest` URL, i.e. [https://kobo.humanitarianresponse.info/assets/[your-asset-UID/](https://kobo.humanitarianresponse.info/assets/[your-asset-UID/), with the form parameter `name=[Your New Name]`. If your form is large, **the response will be large** as well! I have omitted the output below for that reason.

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' -X PATCH https://kobo.humanitarianresponse.info/assets/apLBsTJ4JAReiAWQQQBKNZ/ --
form name='My New Name' | python -m json.tool
```

⚠ If the form was already deployed, you must **redeploy** the form for the name change to appear in data collection clients.

Redeploy a form

First, retrieve the UID of the latest version of the `asset` by making a `GET` request to the `asset` URL. Again, if your form is large, the response will be large as well. Here, I use a simple `grep` to restrict the output to the property I want: `version_id`.

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' https://kobo.humanitarianresponse.info/assets/apLBsTJ4JAReiAWQQQBKNZ/ | python -m
json.tool | grep version_id
    "deployed_version_id": "vh9JnfrRNwVFjJZnSwRYqd",
    "version_id": "ve338oJQooAbkyar7qjukR",
```

Note that `deployed_version_id` is not equal to `version_id`, indicating that the form has been changed since the last deployment.

Now, send a `PATCH` to [https://kobo.humanitarianresponse.info/assets/[your-asset-UID]/deployment/](https://kobo.humanitarianresponse.info/assets/[your-asset-UID]/deployment/) with the form parameter `version_id` set to the UID of the latest version of the form obtained above.

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' -X PATCH
https://kobo.humanitarianresponse.info/assets/apLBsTJ4JAReiAWQQQBKNZ/deployment/ --form
version_id=ve338oJQooAbkyar7qjukR | python -m json.tool
{
    "active": true,
    "backend": "kobocat",
    "identifier": "https://kc.humanitarianresponse.info/jnm_api/forms/apLBsTJ4JAReiAWQQQBKNZ",
    "version_id": "ve338oJQooAbkyar7qjukR"
}
```

Clone a form

`POST` to https://kobo.humanitarianresponse.info/assets/ with the form parameter `clone_from` set to the UID of the `asset` that should be cloned. You may also include a `name` parameter that sets the name of the new `asset`. Again, if you form is large, the response will be large. Just to make this demonstration easier to read, I pipe the output to `tail`:

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' -X POST https://kobo.humanitarianresponse.info/assets/ --form
clone_from=apLBsTJ4JAReiAWQQQBKNZ --form name='My Cloned Form' | python -m json.tool | tail
            "row_count": 1269
        },
        "tag_string": "",
        "uid": "a9Cufbcm2GsqD5oh7u5wXd",
        "url": "https://kobo.humanitarianresponse.info/assets/a9Cufbcm2GsqD5oh7u5wXd/",
        "version_count": 1,
        "version_id": "vq7Xi8wmQYuB4BT3uBdxE3",
        "xform_link": "https://kobo.humanitarianresponse.info/assets/a9Cufbcm2GsqD5oh7u5wXd/xform/",
        "xls_link": "https://kobo.humanitarianresponse.info/assets/a9Cufbcm2GsqD5oh7u5wXd/xls/"
}
```

The **new** `asset`‘s UID is given by the `uid` property in the response.
⚠ The new `asset` will **not** be deployed automatically, even if the source `asset` was deployed.

Share a form

Sharing requires a `POST` to https://kobo.humanitarianresponse.info/permissions/ with three form parameters:

- `content_object`: the URL, relative or absolute, of the `asset` to be shared, e.g. `` `/assets/aSAKqcFRv9nYWqiKGvZC7w/ ``;
- `permission`: a string indicating the permission to grant, e.g. `change_asset`. This can be any assignable permission for an `asset`;
- `user`: the URL, relative or absolute, identifying the user who will receive the new permission, e.g. `/users/jnm/`. The last component of the URL is the username.

```
john@scrappy:/tmp/api_demo$ curl --silent --user jnm_api:test-for-punya --header 'Accept:
application/json' -X POST https://kobo.humanitarianresponse.info/permissions/ --form
content_object=/assets/apLBsTJ4JAReiAWQQQBKNZ/ --form permission=change_asset --form
user=/users/jnm/ | python -m json.tool
{
    "content_object": "https://kobo.humanitarianresponse.info/assets/apLBsTJ4JAReiAWQQQBKNZ/",
    "deny": false,
    "inherited": false,
    "kind": "objectpermission",
    "permission": "change_asset",
    "uid": "pnjnsQXCYEYVSTSXvKAYum",
    "url": "https://kobo.humanitarianresponse.info/permissions/pnjnsQXCYEYVSTSXvKAYum/",
    "user": "https://kobo.humanitarianresponse.info/users/jnm/"
}
```

On Thu, May 10, 2018 at 12:28 PM, Punya Prasad Sapkota <sapkota1@un.org> wrote:

> Dear John,
>
> I am writing to check about the update on API documentation. I appreciate if you could let me know the progress.
>
> Best regards,
>
> Punya
>
> ---
>
> **From:** Punya Prasad Sapkota
> **Sent:** Friday, April 27, 2018 9:32 AM
> **To:** John Milner <john.milner@kobotoolbox.org>
> **Cc:** Tino Kreutzer <tino.kreutzer@gmail.com>; Kashif Siddiqi <siddiqik@un.org>
> **Subject:** RE: KoBo API
>
> Dear John,
>
> Thank you for the message and your willingness to develop API guidance.
>
> Please let me know when it's done and available.
>
> Best regards,

> Punya
>
> **Punya Prasad Sapkota** | Information Management Officer | Information Management & Analysis Unit (IMAU)
>
> United Nations Office for the Coordination of Humanitarian Affairs (OCHA), Turkey
>
> Tel: +90 342 211 8623 | Mob: +90 530 842 5684 | E-mail: sapkota1@un.org
>
> Coordination Saves Lives
>
> Join OCHA Turkey mailing list | Syria cross-border response from Turkey | unocha.org | reliefweb.int | HDX
>
> **From:** John Milner [mailto:john.milner@kobotoolbox.org]
> **Sent:** Tuesday, April 24, 2018 8:51 PM
> **To:** Punya Prasad Sapkota <sapkota1@un.org>
> **Cc:** Tino Kreutzer <tino.kreutzer@gmail.com>
> **Subject:** Re: KoBo API
>
> Hi Punya,
>
> I'm working on your case today and will get back to you with API guidance ASAP. One thing that I wanted to bring up quickly: the Postman screenshot on the Google group includes the authorization header, which contains the base64-encoded username and password. This is trivial to decode, e.g. on a Linux machine "echo cHVsssTA= | base64 -d" yields "pu:..." (redacted last part of password). Please change your password for this account unless it's something you're comfortable sharing with the public.
>
> Thank you,
>
> John Milner
>
> Lead Developer, KoBoToolbox
>
> On Thu, Apr 19, 2018 at 5:02 AM, Punya Prasad Sapkota <sapkota1@un.org> wrote:
>
> > Dear Tino,
> >
> > Thank you for your explanation and guidance.
> >
> > 1.**Importing xlsform**
> >
> > - I have posted information in the google group
> > - Would it be possible to check from you side whether import method is actually working or not?

2) Deployment of an Asset

- I uploaded a xlsform and got the id
- When I deployed it with API, now it goes to archive.

Return of the Posting of deployment

```
[1] "{\"active\":false,\"identifier\":\"https://kc.humanitarianresponse.info/punya/forms
/a37dnoBZNdXbSUezkpk77v\",\"version_id\":\"v7WE9kPHqsg6LiDCa8v5om\",\"backend\":\"koboca
t\"}"
```

Thank you for your suggestion on splitting scripts to multiple files. Updated files are now in the https://github.com/ppsapkota/kobohr_apitoolbox

Best regards,

Punya

Punya Prasad Sapkota | Information Management Officer | Information Management & Analysis Unit (IMAU)

United Nations Office for the Coordination of Humanitarian Affairs (OCHA), Turkey

Tel: +90 342 211 8623 | Mob: +90 530 842 5684 | E-mail: sapkota1@un.org

Coordination Saves Lives

Join OCHA Turkey mailing list | Syria cross-border response from Turkey | unocha.org | reliefweb.int | HDX

From: Tino Kreutzer [mailto:tino.kreutzer@gmail.com]
Sent: Thursday, April 19, 2018 4:56 AM

To: Punya Prasad Sapkota <sapkota1@un.org>
Cc: Anwar Mahfoudh <mahfoudh@un.org>; Kashif Siddiqi <siddiqik@un.org>; Emily Ferguson <emily.ferguson@kobotoolbox.org>; John Milner <john.milner@kobotoolbox.org>
Subject: Re: KoBo API

Hi Punya,

I'm not sure why the POST isn't working with either your script or with Postman, but I'm glad the other parts of the script seem to work. I looked at your R script and I would suggest splitting it up into different smaller

> > files that can be called together as needed. This way you can separate what works from what doesn't and keep tab on the readme.md file. That will make it easier for someone (including our programmers) to help out.
> >
> > I suggest posting a message with your issue to our https://groups.google.com/forum/#!forum/kobo-users group. People there are quite technical and there is a chance someone may have done this before. Being able to say that you can use the KPI API for other calls but not for POSTing an XLSForm will help (because your problem is not, say, authentication). I suggest keeping the message about just that: "Unable to POST XLS form file to new KoBo API". Please include the exact steps you took in configuring Postman (and/or pointing to the specific script that someone else can run easily, i.e. just for posting the XLS form) so that others can try to repeat those steps and see why it's not working. I'll do what I can to get the right people to weigh in on and share their insight with you.
> >
> > Best,
> >
> > Tino
> >
> > On Wed, Apr 18, 2018 at 5:20 PM Punya Prasad Sapkota <sapkota1@un.org> wrote:
> >
> > > Dear Tino,
> > >
> > > In addition to my previous email, I tried to test it with Postman. It seems that there is an error in the POSTing process to upload Xlsform file.
> > >
> > > Best regards,
> > >
> > > Punya
> > >
> > > **Punya Prasad Sapkota** | Information Management Officer | Information Management & Analysis Unit (IMAU)
> > >
> > > United Nations Office for the Coordination of Humanitarian Affairs (OCHA), Turkey
> > >
> > > Tel: +90 342 211 8623 | Mob: +90 530 842 5684 | E-mail: sapkota1@un.org
> > >
> > > Coordination Saves Lives
> > >
> > > Join OCHA Turkey mailing list | Syria cross-border response from Turkey | unocha.org | reliefweb.int | HDX

From: Punya Prasad Sapkota
Sent: Wednesday, April 18, 2018 7:35 PM
To: Tino Kreutzer <tino.kreutzer@gmail.com>

Cc: Anwar Mahfoudh <mahfoudh@un.org>; Kashif Siddiqi <siddiqik@un.org>; Emily Ferguson <emily.ferguson@kobotoolbox.org>; John Milner <john.milner@kobotoolbox.org>

Subject: RE: KoBo API

Dear Tino,

Thank you for the response. I had tried import earlier but I was not successful. I will check the import part tomorrow and provide you update.

The piece of code I am using in R which may give you some hints

```
###################---KPI--------##########################################

####------------------ASSETS IMPORT------------

kobo_form_xlsx<-"./xlsform/kobo_1701_NW.xlsx"

kb_url <- "https://kobo.humanitarianresponse.info/imports/"

result<-httr::POST (kb_url,

                  body=list(

                    xls_file=upload_file(path=kobo_form_xlsx)),

                  authenticate(kobo_user,Kobo_pw)

)
```

I get server error - `[1] "<h1>Server Error (500)</h1>"`

When I change url to https://kc.humanitarianresponse.info/api/v1/forms - above code is working and created form in KC interface. Would it be possible to check the parameter names that I need to provide when uploading files. The error may be linked to xls_file para.

To test the approach, I uploaded a xlsform manually, got the uuid and tried to deploy using the API. It is working.

```
###-----------ASSETS DEPLOYMENT - WORKING-----------

url <-paste0("https://kobo.humanitarianresponse.info/assets/aVJ3qxffPCPttQ79stbdNL/","deployment/")

d <- list(owner='https://kobo.humanitarianresponse.info/users/punya/')

result<-httr::POST (url,

                    body=d,

                    authenticate(kobo_user,Kobo_pw))
```

If import method does not work, I am thinking about alternatives:

1. I can upload a form manually, get the uuid and deploy it. – can I bulk import XLSForms? – I can deploy them using API.
2. Copy/clone the asset and redeploy it – need POSTing link for this one
3. Change parameter (form name etc) – need POSTing link guidance
4. Share with the user – need POSTing link guidance

Looking forward to your support. I have already spent good amount of time on and don’t want to give up yet. If I am successful on it, it will really help in saving time in future projects.

Best regards,

Punya

Punya Prasad Sapkota | Information Management Officer | Information Management & Analysis Unit (IMAU)

United Nations Office for the Coordination of Humanitarian Affairs (OCHA), Turkey

Tel: +90 342 211 8623 | Mob: +90 530 842 5684 | E-mail: sapkota1@un.org

Coordination Saves Lives

Join OCHA Turkey mailing list | Syria cross-border response from Turkey | unocha.org | reliefweb.int | HDX

From: Tino Kreutzer [mailto:tino.kreutzer@gmail.com]
Sent: Wednesday, April 18, 2018 6:49 PM
To: Punya Prasad Sapkota <sapkota1@un.org>
Cc: Anwar Mahfoudh <mahfoudh@un.org>; Kashif Siddiqi <siddiqik@un.org>; Emily Ferguson <emily.ferguson@kobotoolbox.org>; John Milner <john.milner@kobotoolbox.org>
Subject: Re: KoBo API

Hi Punya,

Apologies for the slow response. I know you have spoken to Emily in the meantime, but I'll try to address your issue directly now (Forms uploaded to KC.hr.info are not reflected in the KPI project list):

1. Direct uploads to KC are still possible but no longer supported. We strongly encourage that you only deploy forms to KPI directly, ideally in the UI
2. If you can't then these projects will not be available in KPI. We had previously synched the two tools from KC to KPI but had to disable it as it slowed down the server for everyone.
3. If you want to use the KPI API, you need to import your form first (POST to https://kobo.humanitarianresponse.info/imports/), which will create a new asset in KPI which you can then deploy (by POSTing to https://kobo.humanitarianresponse.info/assets/[asset ID number]/deployment/)

For point (1) you can also deploy forms via the KPI library instead of uploading an XLSForm file each time. You can also use the API to make changes to that form after cloning it (e.g. to reflect the name of the NGO in the title or delete questions). All of this is possible via the KPI API, not the old KC API. It's much more powerful. We're still lacking documentation for it, sadly. BUT you can see how it works and build your script accordingly by keeping the network tab open in your browser as you upload/modify/deploy projects, which will show you exactly how all of these step work. The UI is using the KPI API for everything so you can mirror everything it does in your own script.

We definitely want to help you make this coordinated assessment a success and I'll work with you directly to get us there.

We don't have example python scripts outside of what is done in KPI itself, but there may be others who have done this. I suggest posting a question on the IM Skype group about this.

> > > Best,
> > >
> > > Tino
> > >
> > > On Fri, Apr 6, 2018 at 3:50 PM, Punya Prasad Sapkota <sapkota1@un.org> wrote:
> > >
> > > > Dear Tino,
> > > >
> > > > I hope that you are well.
> > > >
> > > > I am writing back again seeking your support for the KoBo API problem. So far, I am able to crate project, upload XLSform, assign uploaded form to a project and share the form with another account. However, the project appears in Legacy UI only. I cannot find them in the KPI project list and also cannot find a way to publish in the new KPI form list. I am using https://kc.humanitarianresponse.info/api/v1/ and related links using R.
> > > >
> > > > I tried to use https://kobo.humanitarianresponse.info/imports/ and assets url but could not succeed. Do you have any working examples that upload xlsform in kpi project list and also share with another account? Example in Python is also fine.
> > > >
> > > > The solution I am trying is to achieve the following:
> > > >
> > > > - We have over 150 Syrian NGOS who submits data for their 3Ws. So far we are using a single KoBo form for all to submit information. We update this information quarterly and every time organisations have to enter all information.
> > > > - Majority of the partners are familiar with KoBo and use it for assessments. The solution we are planning is to create multiple copies of the same xlsform (one for each partner) and share with their kobo account. This will help them to open the existing record and edit information to maintain 3W information up-to-date for themselves and also accessible to us at the same time. Since situation in the field is changing rapidly, the provision of allowing individual organisations maintain their operational presence information and share with us at the same time greatly helps us.
> > > > - With the use of API programming (R or Python), we see a potential to automate the whole process (form creation, sharing, updating, downloading data). The only problem I am facing now is to show the deployed forms in KPI project list. Your support would help us to achieve this. The same solution could also be used for the Multi-sector needs assessment that is planned for the HNO during July/August. For this type of assessments, we have been deploying over 50 copies of the same form in previous years. We may be exploiting the KoBo system by using it the way it wasn't planned for.
> > > >
> > > > I am using 'punya' kobo account for this purpose. I appreciate for your support.

Best regards,

Punya

Punya Prasad Sapkota | Information Management Officer | Information Management & Analysis Unit (IMAU)

United Nations Office for the Coordination of Humanitarian Affairs (OCHA), Turkey

Tel: +90 342 211 8623 | Mob: +90 530 842 5684 | E-mail: sapkota1@un.org

Coordination Saves Lives

Join OCHA Turkey mailing list | Syria cross-border response from Turkey | unocha.org | reliefweb.int | HDX