* [Docs](https://docs.opendatakit.org/) »

* Briefcase Aggregate API
* [Edit on GitHub](https://github.com/opendatakit/docs/blob/master/odk1-src/briefcase-api.rst)

**Briefcase Aggregate API**

**Introduction**

This describes the interfaces ODK Briefcase uses when interacting with ODK Aggregate v1.0.

In general, the server should adhere to the OpenRosa API described [here](https://bitbucket.org/javarosa/javarosa/wiki/OpenRosaAPI).

**Pull**

Interfaces used during Pull actions.

The APIs used:

1. OpenRosa Form Discovery API
2. Download forms and media files using URLs form discovery.
3. view/submissionList to obtain a chunk of submission keys.
4. view/downloadSubmission to download an individual submission.

**Push**

Interfaces used during Push actions.

1. *formUpload* to upload a form and its media files to ODK Aggregate.
2. view/submissionList to obtain a chunk of submission keys.
3. OpenRosa Form Submission/Overwrite API (with extensions)

**Interface Details**

**OpenRosa Form Discovery API**

Vanilla conformance.

**OpenRosa Form Submission/Overwrite API**

**Server Processing**

The server does special processing to look for an instanceID or a submissionDate attribute on the top-level element of the submission.

If the OpenRosa Metadata block is not present, it uses the instanceID attribute value as the instance ID for this submission.

If the submissionDate is present, it uses that value as the submission date for this submission (rather than the current datetime). SubmissionDate should be in ISO8601 format.

**Response**

The response from the server looks like:

**<OpenRosaResponse** xmlns="http://openrosa.org/http/response"**>**

**<message>**...**</message>**

**<submissionMetadata** xmlns="http://www.opendatakit.org/xforms"

id="formid" version="..." encrypted="yes" instanceID="..."

submissionDate="..."

isComplete="true" markedAsCompleteDate="..."**/>**

The <submissionMetadata/> tag contains all the metadata for this submission that the server has assigned it.

The form id and version are given (version is omitted if null).

If the form is an encrypted form, the encrypted attribute is present, otherwise it is omitted. If present, it has a value of *yes* (vs. *true*).

The instanceID is either the server-assigned instance id or the instanceID from the OpenRosa Metadata block.

The submissionDate is the date of first submission of this record on the server.

The isComplete flag is *true* if the server has all attachments associated with this filled-in form. Otherwise, it will be *false*. If it is *true*, the markedAsCompleteDate will be the date when the submission became complete.

The date fields are formatted using:

**import** **org.javarosa.core.model.utils.DateUtils**;

DateUtils.formatDateTime(d, DateUtils.FORMAT\_ISO8601)

The only 2 fields ODK Briefcase currently uses are the instanceID and submissionDate fields. It may eventually use the isComplete flag.

**Briefcase Treatment**

After a submission has been pushed to the server, it updates the top-level element, inserting the instanceID and submissionDate fields if they are not already present.

If they are present, it compares the fields and warns if there is any discrepancy between them.

**Download forms and media files**

It is same as the API ODK Collect uses.

ODK Briefcase downloads the forms so that it has the form model available for CSV exports and to ensure that a push uses matching form definitions.

**formUpload**

This is FormUploadServlet in ODK Aggregate.

It is a multipart POST. The POST always contains the form definition file, and has as many media files as would fit in under 10MB. This follows the behavior of the OpenRosa Form Submission/Overwrite API.

Because of the limitations of HTML file input tags, all media files must be in a single folder (no sub-directories). The server assumes this is the case.

Here is a sample web page fragment to post to this servlet:

**<form** id="ie\_backward\_compatible\_form" action="/formUpload"

enctype="multipart/form-data" encoding="multipart/form-data"

method="POST" accept-charset="UTF-8"**>**

**<table** id="uploadTable"**>**

**<tbody>**

**<tr>**

**<td>**

**<label** for="form\_def\_file"**>**Form definition (xml file):**</label>**

**</td>**

**<td>**

**<input** id="form\_def\_file" class="gwt-Button" type="file"

name="form\_def\_file" size="80"**>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<label** for="mediaFiles"**>**Optional Media file(s):**</label>**

**</td>**

**<td>**

**<input** id="mediaFiles" class="gwt-Button" type="file"

multiple="" name="datafile" size="80,20"**>**

**<input** id="clear\_media\_files" class="gwt-Button"

type="button"

onclick="clearMediaInputField('mediaFiles')"

value="Clear"**>**

**</td>**

**</tr>**

**<tr>**

**<td>**

**<input** class="gwt-Button" type="submit"

value="Upload Form" name="button"**>**

**</td>**

**</tr>**

**</tbody>**

**</table>**

**</form>**

**GET view/submissionList**

Download the list of completed submissions for a given form. Incomplete or partial submissions **must not** be returned. You may wish to report only approved submissions if you have a QA review step.

There are 3 query arguments:

1. formId – identifies the form. Just the id from the form definition (e.g., geo\_tagger\_v2).
2. cursor – an opaque string used by the server to as a query resume point. Omit on the first call.
3. numEntries – the number of entries to return.

**Returned Document**

The returned XML document is of the form:

**<idChunk** xmlns="http://opendatakit.org/submissions"**>**

**<idList>**

**<id>**uuid:e5aa4247-cfb8-4cc5-87cd-52fbfe491b13**</id>**

**<id>**uuid:8ca4433a-fed2-4d1c-b333-c0378203f68f**</id>**

**</idList>**

**<resumptionCursor>**opaquedata**</resumptionCursor>**

**</idChunk>**

Where the idList contains a series of id elements containing strings that can be used to construct the needed string to pass to the view/downloadSubmission API.

The resumptionCursor holds opaque data that is used by the server to track the location at which to resume the list of ids.

**ODK Briefcase Treatment**

ODK Briefcase repeatedly calls this API, passing in the previous response's resumptionCursor value until the returned resumptionCursor value matches that given in the request. Once it no longer changes, ODK Briefcase assumes that all id data has been downloaded from the server.

**GET view/downloadSubmission**

Download an individual submission.

One Query argument: formId – a multipart XPath-like string query that identifies the individual submission.

The formId is a string of the form:

formid[@version=null and @uiVersion=null]/topElement[@key=idvalue]

* formid is the form id, as would be provided to the submissionList API.
* version is either null if not used or the version number of the form (ODK Aggregate only supports integer version strings at this time).
* topElement is the name of the top-level element in the submission. This is the element nested just within the <instance> element of the <model> (it also has the id attribute, which should be equal to the formid).
* idvalue is the value of the individual submission id returned by the view/submissionList API, usually a UUID.

An example, where no version was attached to the form definition uploaded to Aggregate:

GET /view/downloadSubmission?formId=my\_odk\_form[@version=null and @uiVersion=null]/data[@key=uuid:38fd4ef4-28b9-441e-a818-dd8cbe514b2c]

While it is desirable for the idvalue to be the instanceID of the submission, it is not required. For ODK Aggregate v1.x, it will be the instance ID; for ODK Aggregate 0.9.x, it is not the instance ID.

**Note**

formid may be a URL (since it might be an xmlns), so when parsing this query parameter, it is safest to find the last instance of @version and split the string at that location. In any case, it is your string to parse and interpret for your server configuration.

**Response Document**

The response is of the form:

**<submission** xmlns="http://opendatakit.org/submissions"

xmlns:orx="http://openrosa.org/xforms" **>**

**<data>**

...reconstructed submission XML...

**</data>**

**<mediaFile>**

**<fileName>**...**</fileName>**

**<hash>**md5:...**</hash>**

**<downloadUrl>**...**</downloadUrl>**

**</mediaFile>**

...repeat as needed...

**</submission>**

The <mediaFile> tag has the same interpretation as it does in the OpenRosa Form Listing API's manifest XML.

The reconstructed submission XML generally does not respect the namespaces of the original form definition. As a special case, if it finds a form group that could be interpreted as the OpenRosa Metadata block, it does use the orx namespace for that.

Like the <submissionMetadata> tag on the Form Submission/Overwrite API response, the top-level element in the submission XML contains all the metadata fields supplied by the server.

[Launching ODK Collect from External Apps](https://docs.opendatakit.org/launch-collect-from-app/)[Encrypted Forms](https://docs.opendatakit.org/encrypted-forms/)

[Download this documentation as a PDF.](https://docs.opendatakit.org/_downloads/ODK-Documentation.pdf)

If you still need help, you can ask support questions in the [ODK Forum](https://forum.opendatakit.org/)

If you find a problem with this documentation, please [file an issue](https://github.com/opendatakit/docs/issues)

You are also encouraged to [fork our Github repo](https://github.com/opendatakit/docs/)and [become a contributor](https://docs.opendatakit.org/contributing/)

© Copyright 2017, Open Data Kit. This document is licensed under a Creative Commons Attribution 4.0 International License.

Built with [Sphinx](http://sphinx-doc.org/) using a [theme](https://github.com/snide/sphinx_rtd_theme) provided by [Read the Docs](https://readthedocs.org/).