

Uploading Files to ScienceBase

Uploading Files

Although uploading files to ScienceBase RESTfully technically falls into the POST/PUTs category, it is slightly different and thus deserves its own instructions.

First things first, Headers. As with all REST calls for JSON on ScienceBase we need the correct Accept header of application/json:

Headers

```
[name: Accept, value: application/json]
```

You must also have your JOSSO_SESSIONID from the JOSSO_SESSIONID cookie. To find out more about this see [Authenticating](#).

uploadAndCreateItem or uploadAndUpdateItem

There are two ways to upload files, you can upload files and create a new item with them or you can upload files to an existing item. In both cases you need a ScienceBase Id (sb:id). When you use the uploadAndCreateItem service this sb:id will be the id of the parent item that you want to create this new item under. When you use the uploadAndUpdateItem service the sb:id will be the id of the item you wish to upload files to. Both of these services are located at <https://www.sciencebase.gov/catalog/file>

The request is made is a multipart/form-data request (you can add this header value with [name: Content-Type, value: multipart/form-data] just to be safe). The parts consist of files, each of which is named "file" and a sbJSON part which is named item. With these parts you can upload as many files as you wish as well as creating (or updating) with sbJSON.



You may also have to manually set your boundary for your files. See example request at the bottom of this page to see all of the headers for the entire request as well as each part.

files

At least one file is required (otherwise you should be using the faster item services). Each file's name is "file", its Content-Disposition is "form-data", its filename should be its filename, its Content-Type should be "application/octet-stream", and its Content-Transfer-Encoding should be "binary". You may not have to manually set all of these, but you should be aware of them. As of 8/30/12 you can send as many files as you want, each with the name "file" and they will be iterated through and added.

item sbJSON

You can also send in sbJSON with your item, this way you do not have to make separate calls creating/updating the item's metadata and uploading files. Its Content-Disposition is "form-data", its name is "item", its Content-Type is "text/plain", its charset is "US-ASCII", and its Content-Transfer-Encoding is "8bit". The content of this should be a string representing the sbJSON you are inputting.



If you are updating a record that already contains files the item's file list will be updated with the new file. If you wish to delete the old files set "files: []" in the sbJSON you submit.

Example Upload Paths

uploadAndCreateItem service

[https://www.sciencebase.gov/catalog/file/uploadAndCreateItem/\[sb:parentId\]?josso=\[JOSSO_SESSIONID\]](https://www.sciencebase.gov/catalog/file/uploadAndCreateItem/[sb:parentId]?josso=[JOSSO_SESSIONID])

uploadAndUpdateItem service

[https://www.sciencebase.gov/catalog/file/uploadAndUpdateItem/\[sb:id\]?josso=\[JOSSO_SESSIONID\]](https://www.sciencebase.gov/catalog/file/uploadAndUpdateItem/[sb:id]?josso=[JOSSO_SESSIONID])

Example Request

This request was caught using reflect.py, which reflects local requests to the command line so you can evaluate them. It is available for download here: <http://gist.github.com/raw/814831/fb48dd96467a5e52edeff2010d53f30278926391/reflect.py>

```
----- Request Start ----->

/file/uploadAndCreateItem/5033ff1ae4b068b9cdc54853?josso=robb
Accept: application/json
Content-Length: 12747
Content-Type: multipart/form-data; boundary=_txlIVyvBlA6lJIq9-laQOG94lj_hyCS6rKfp4
Host: localhost:8080
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.2.1 (java 1.5)

--_txlIVyvBlA6lJIq9-laQOG94lj_hyCS6rKfp4
Content-Disposition: form-data; name="file"; filename="file1.txt"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

[... file 1 contents ...]

--_txlIVyvBlA6lJIq9-laQOG94lj_hyCS6rKfp4
Content-Disposition: form-data; name="file"; filename="file2.txt"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

[... file 2 contents ...]

--_txlIVyvBlA6lJIq9-laQOG94lj_hyCS6rKfp4
Content-Disposition: form-data; name="item"
Content-Type: text/plain; charset=US-ASCII
Content-Transfer-Encoding: 8bit

{title: 'File Upload Test', provenance:{html: 'File upload from example in the ScienceBase Documentation'}}
--_txlIVyvBlA6lJIq9-laQOG94lj_hyCS6rKfp4--

<----- Request End -----
```