
Digital Herbarium Documentation

Release

Dmitry E. Kislov

Aug 04, 2017

Оглавление

1	Digital Herbarium: Basic Usage	3
1.1	Features	3
1.2	Search menu	3
1.3	Filtering search results	6
2	Digital Herbarium's HTTP-API Description	11
2.1	Intro	11
2.2	Description of HTTP request parameters	11
2.3	Description of server response	13
2.4	Service usage limitations	17
2.5	Examples	17
3	Herbarium Record's citing	19
	Index	21

Sections “[Digital Herbarium: Basic Usage](#)” and “[Digital Herbarium’s HTTP-API Description](#)” are devoted peculiarities of user search activity using web-interface and programming languages respectively.



Printable version

DIGITAL HERBARIUM: BASIC USAGE

1.1 Features

Accessing to the Digital Herbarium's data is provided via either the [web-page](#) at the official website of the Botanical Garden Institute, or [HTTP API service](#) exploiting when performing search queries automatically (from [R](#), [Python](#) or other computational environment).

Only published herbarium records are shown when do search of any kind.

Main features of the search service:

- search in a given time interval either by date of collection or date of identification fields;
- accounting species synonyms when searching;
- search in a given rectangular region;
- search within additional species (only for multispecies herbarium records);
- search by record's codes (e.g. field number, inventory number etc.);
- search by the country of origin;
- search by taxonomic name, e.g. via family, genus or species epithet;

1.2 Search menu

General search possibilities is available via the search menu from the Herbarium catalog's webpage ([Fig. 1](#))

When search conditions are given simultaneously, the service trying to perform an AND-type query; it retrieves records satisfying all the search conditions. Therefore, only AND-type search queries are available (at least, currently). To perform OR-type queries it is recommended to look toward the [HTTP API service](#).

Values of Family, Genus and Country search fields are selected via drop-down menu that raised when typing.

Start date of collection and end date of collection are filled out by means pop-up calendar when the mouse is overed these fields.

If only start date of collection is given, the service retrieves records having greater dates in the corresponding field.

If only end date of collection is given, the service retrieves records having lesser dates in the corresponding field.

If start date of collection and end date of collection are given, the service retrieves records if its corresponding date interval intersect the given.

Find

Clear

Family:

Select Family

Genus:

Select Genus

Species epithet:

☐ Search within synonyms

☐ Search within additional species

Code:

Collector(s):

Text field

Identifier(s):

Text field

Country:

Select Country

Place of collection:

Place of collection

Collection start date:

Collection start date

Collection end date:

collection end date

Рис. 1.1: Fig. 1. Basic search menu

Regarding the following text fields — Species epithet, Code, Collectors, Identifiers, Place of collection the condition satisfaction assumes including the given value as a substring into the corresponding field (case insensitive comparison is performed).

If one performs search in the either Collectors or Identifiers fields and fills these fields with cyrillic letters, the service will automatically transliterate the given value into English (latin letters) and returns records satisfying both cyrillic and transliterated values. If you provide the value only in latin letters, no transliteration will be performed. Therefore, If you will try, for example, to find records including “bakalin” as a substring in field Collectors, the search engine will return the records which field Collectors (internally Collectedby field) includes the string “bakalin” (reverse transliteration (to cyrillic letters) in this case willn’t be performed); If you will try to search “бакалин” (cyrillic equivalent of ‘bakalin’) combined search results for both “bakalin” and “бакалин” queries will be returned.

Boolean fields Search within synonyms and Search within additional species indicate that, in the first case — the search engine will take into account known (to the system) table of species synonyms, and in the second — the search engine do searching with additional species if those are provided.

Warning: When do searching within species synonyms, the search engine uses the table of species synonyms that, in turn, is dynamically rebuilt each time records in the Table of known species are updated. The Table of known species can include errors, especially regarding species synonym relationships. This could lead to getting surprising search results. These type of drawbacks (caused by incorrectness of species synonym relationships) will be neglected in the future, as the Table of known species will become more errorless.

Note: Search within synonyms works in cases when exact names of the pair (genus, species epithet) are given. In other cases this search condition is ignored and the note is rised.

1.2.1 Search by Code field

Herbarium records stored in Digital Herbarium of the BGI use triple coding system. Each record is provided with 1) inventory number (optional), used in the Herbarium’s storage; 2) mandatory ID field (unique, digits only), assigned by the system automatically; 3) field number (code), assigned by the collector (it is optional and quite arbitrary);

Therefore, the table of search results includes the column Complex code accumulates codes of these three types.

Complex code has the following structure:

Note: Inventory number (if provided) or * symbol/ID code/Field code (if provided)

So, the Complex code values can look as follows:

- */27031/M.I.38 — denotes that the inventory number isn’t provided, ID = 27031, and field code is M.I.38;
- 42/27029 — denotes that the inventory number is 47, ID = 27029, field code isn’t provided;
- the following form of the code can take place as well: 132123/32032/F-3829-3k, where inventory number is 132123, ID is 32032 and field code is F-3829-3k (this is fake example, I didn’t find a real herbarium record with all three setted codes)

When do searching by Code one should provide either an inventory number, ID or field code. For example, if the search field's value is "231" the search engine will return records including "231" as a substring in either the inventory code, ID or field code.

1.3 Filtering search results

Standard filtering interface allows to restrict results of searching by herbarium's acronym, herbarium's subdivision or select desired number of items showed per page [Fig. 2](#).

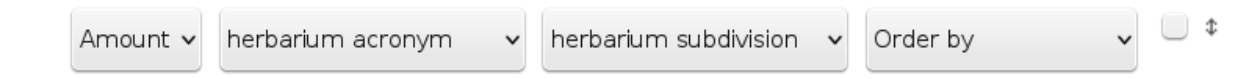


Рис. 1.2: Fig. 2. Search filtering menu

It has the following fields:

- Amount — the number of records showed per page;
- Herbarium acronym — filtering by Herbarium's acronym;
- Herbarium subdivision — filtering by Herbarium's subdivision;
- Order by — ordering rule (choose field you want to perform ordering the results);

Results of search request and filter applying is presented on the [Fig. 3](#).

In the tab Common Info showed a table with the records satisfying current search and filtering conditions (if no conditions were provided all published records are shown, with default its the number-per-page equal to 20)

The Details tab is activated when a specific Herbarium's record is clicked. It shows minified version of the Personal web-page of a record.


The Map tab is a copy of Common Info tab exclude records with no coordinates (records with coordinates are rendered on the Google map as clickable markers).

One can click Previous or Next to get another portion (switch page) of search results.

The Automatizatin tools tab include general information on query [automatization possibilities](#) provided by the web-application.

Working with the map, one can filter search results by user-defined rectangular area. To do that, just initialize a rectangular area by pressing [\[a\]](#), edit the rectangular region rised, and press [\[a\]](#) again to activate the search engine (See [Fig. 4](#), [Fig. 5](#)).

To clear any specific search condition click small-trash icon near the corresponding search field.

To clear all search conditions press the  button.

Search in polygonal regions doesn't allowed in the current version of the backend database, but such behaviour could be emulated programmatically with help of the [HTTP API Service](#).

Search results:

Page 1 of 606 Next

Total: 12116

Sheet's code (combined)	Species	Collection date	Collector(s)	Identifier(s)
*/27285	<i>Camptosorus sibiricus</i> Rupr.	2004-08-10	Kreshchenok I.A.	Kreshchenok I.A.
*/27284	<i>Camptosorus sibiricus</i> Rupr.	2004-08-15	Kreshchenok I.A.	Kreshchenok I.A.
*/27283	<i>Camptosorus sibiricus</i> Rupr.	2002-06-20	Vorob'eva A.N.	Vorob'eva A.N.
*/27282	<i>Camptosorus sibiricus</i> Rupr.	2015-06-24	Veklich T.N.	Veklich T.N.
*/27281	<i>Camptosorus sibiricus</i> Rupr.	2012-07-01	Veklich T.N., Darman G.F.	Veklich T.N., Darman G.F.
*/27280/16	<i>Asplenium tenuicaule</i> Hayata	2005-09-10	Koldaeva M.N.	Kreshchenok I.A.
*/27279	<i>Asplenium ruta-muraria</i> L.	2005-09-03	Kreshchenok I.A.	Kreshchenok I.A.
*/27278	<i>Asplenium ruta-muraria</i> L.	2007-07-31	Kreshchenok I.A., Rubtsova T.A., Fetisov D.M., Zaitseva N.V.	Kreshchenok I.A.
*/27277	<i>Aleuritopteris kuhnii</i> (Milde) Ching	2005-07-10	Kreshchenok I.A., Starchenko V.M., Darman G.F., Borisova I.G.	Kreshchenok I.A.
*/27276	<i>Aleuritopteris argentea</i> (S.G.Gmel.) Fee	2004-09-03	Kreshchenok I.A.	Kreshchenok I.A.

Рис. 1.3: Fig. 3. Search results tab

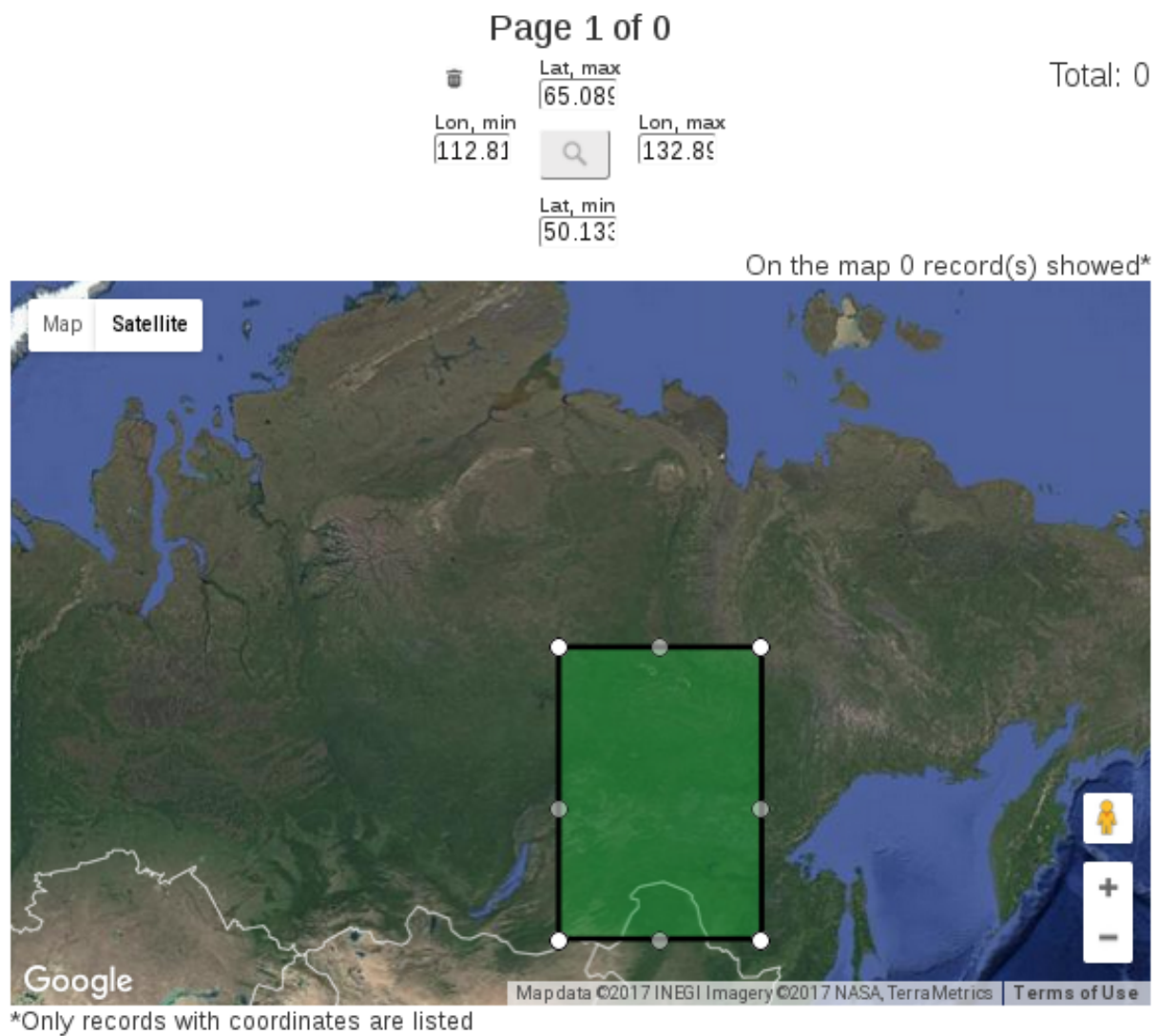


Рис. 1.4: Fig. 4. Initialize filtering region

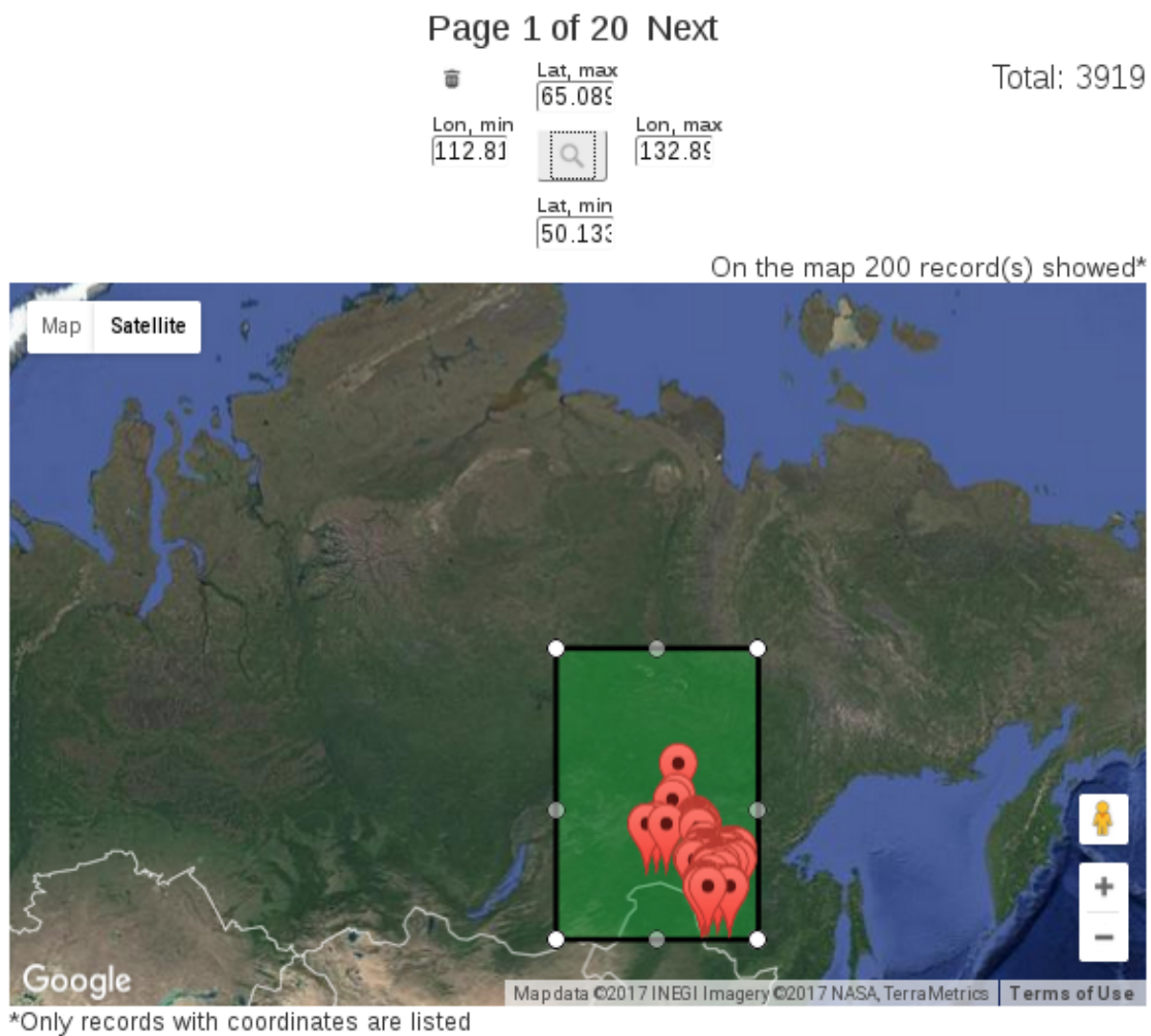


Рис. 1.5: Fig. 5. Getting results of geographical filtering/searching

DIGITAL HERBARIUM'S HTTP-API DESCRIPTION

- [Intro](#)
- [Description of HTTP request parameters](#)
- [Description of server response](#)
 - [Format of the data attribute](#)
 - * [History of species identifications and additional species](#)
- [Service usage limitations](#)
- [Examples](#)

2.1 Intro

This document describes HTTP-API (Application Programming Interface over HTTP protocol) used to get access Digital Herbarium Database of the BGI.

HTTP-API works in readonly mode. There is no way to make changes in the database using the API.

2.2 Description of HTTP request parameters

Only GET-requests are allowed when talking with the HTTP API service. To establish connection with the service, one can use either HTTP or HTTPS protocols.

Requests with multiple parameters, e.g. colstart=2016-01-01 and collectedby=bak, are treated as components of AND-type queries: in this example, all records collected after 2016-01-01 and including bak (case insensitive matching is performed) as a substring of Collectors field of Herbarium's record will be returned.

OR-type querying behaviour can be emulated by a series of consequent queries to the database and isn't natively implemented in the current version of the HTTP API.

List of allowed GET-parameters:

- family — family name (matching condition: case insensitive, the same family name as provided);
- genus — genus name (matching condition: case insensitive, the same genus name as provided), note: if the value contradicts with the family name provided in the same request, an error will be returned;

- `species_epithet` — species epithet (matching condition: case insensitive, a substring of the record corresponding field);
- `place` — place of collection (matching condition: case insensitive, a substring occurring in the one of listed fields: Place, Region, District, Note;;);
- `collectedby` — collectors (matching condition: case insensitive, a substring of the record corresponding field); if the field's value is given in cyrillic, search will be performed (additionally) on its transliterated copy;
- `identifiedby` — identifiers; (matching condition: case insensitive, a substring of the record corresponding field); if the field's value is given in cyrillic, search will be performed (additionally) on its transliterated copy;
- `country` — country's name (matching condition: case insensitive, a substring of the record corresponding field);
- `colstart` — date when herbarium object's collection was started (yyyy-mm-dd);
- `colend` — date when herbarium object's collection was finished (yyyy-mm-dd);
- `acronym` — name of the herbarium's acronym (matching condition: case insensitive, the same name as provided);
- `subdivision` — name of the herbarium's subdivision/branch (matching condition: case insensitive, the same name as provided);
- `latl` — latitude's lower bound, should be in (-90, 90);
- `latu` — latitude's upper bound, should be in (-90, 90);
- `lonu` — longitude's upper bound, should be in (-180, 180);
- `lonl` — longitude's lower bound, should be in (-180, 180);
- `synonyms` — boolean parameter, allowed values are false or true; absence of the parameter in GET-request is treated as its false value; true value (e.g. `synonyms=true`) tells the system to search records taking into account the table of species synonyms; Note: when performing search including known (known by the system) species synonyms one should provide both genus and `species_epithet` values, if only one of these is provided or both are leaved empty, a warning will be shown and the search condition will be ignored;
- `additional` — boolean field, allowed values are false or true; absence of the parameter in GET-request is treated as its false value; true value (e.g. `additional=true`) tells the system to search within additional species (if such is provided); some herbarium records could include more than one species (such records referred as multispecies records);
- `id` — record's ID (matching condition: the same value as provided); if this parameter is provided in GET-request, all the other search parameters are ignored and the only one record with the requested ID is returned (if it exists and is published);
- `fieldid` — field code/number; (matching condition: case insensitive, a substring of the record corresponding field);
- `itemcode` — storage number (matching condition: case insensitive, a substring of the record corresponding field);
- `authorship` — authorship of the main species (matching condition: case insensitive, a substring of the record corresponding field);

Note: The search engine performs only one-way transliteration of fields `collectedby` and `identifiedby` to English language. So, if you will try to search, e.g. `collectedby=боб` (that corresponds to bob in English), the

system will find records including (in the collectedby field) either боб or bob substrings. On the contrary, If you will try to send collectedby=bob search query, only records include bob will be found (regardless the text case).

Warning: Transliteration from cyrillic (Russian) to latin (English) is fully automatic and could be quite straightforward, e.g. Джон will be transliterated into something like Dzhon, instead of John, as it would expected.

2.3 Description of server response

The server response is a **JSON-formatted** text transferred via HTTP-protocol and having the following attributes:

- errors — array of errors (each error is a string) occurred during search request evaluation;
- warnings — array of warnings (each warning is a string) occurred during search request evaluation;
- data — array of structured data, i.e. result of the search query.

Note: Warnings are informative messages used to tell the user whats happened during the database intraction in an unexpected way: e.g. which search parameters contradict each other, which parameters were ignored, which parameters weren't recognized by the system etc.

2.3.1 Format of the data attribute

The data attribute is a JSON-formatted array. Each item of this array describes a herbarium record and has the following attributes:

- family — family name (latin uppercase letters);
- family_authorship — family authorship;
- genus — genus name;
- genus_authorship — genus authorship;
- species_epithet — species epithet;
- species_id — ID of species instance (unique integer value); don't mix with ID of the herbarium record. ID of the herbarium record is unique among all herbarium records, ID of the species instance is unique among all species instances;
- short_note — used in multispecies herbarium records; the field provides important information about the main species of the herbarium record (it could be empty);
- species_authorship — species authorship;
- species_status — current species status; the term species status is related to species instance not herbarium record; it describes a degree of acceptance the species by scientific community (current state); Possible values of species_status are 'Recently added' — the species was recently included to the database and wasn't checked by an expert, 'Approved' — the species was approved by an expert (a user having some prevelegies), 'Deleted' — the species name is probably obsolete and should be avoided, 'From plantlist' — the species was imported from the <http://theplantlist.org>;

- `species_fullname` — full species name, e.g. Genus + species epithet + species authorship;
- `significance` — measure of ambiguity regard the main species (possible values: “”, aff., cf.);
- `id` — integer identifier of a herbarium record, it is unique;
- `gpsbased` — boolean parameter, its true value means that a herbarium record position is obtained via the GNSS (GPS/GLONASS); true value — guaranties that coordinates were obtained via GNSS.
- `latitude` — latitude, degrees (WGS84);
- `longitude` — longitude, degrees (WGS84);
- `fieldid` — field number; an arbitrary string assigned by a collector;
- `itemcode` — inventory (storage) number, a string assigned by the herbarium’s curator; it is used to identify the place of the record in the herbarium storage;
- `acronym` — herbarium acronym (e.g. VBGI);
- `branch` — herbarium branch (e.g. “Herbarium of Fungi”, “Bryophyte Herbarium” etc.);
- `collectors` — collectors;
- `identifiers` — identifiers;
- `devstage` — development stage; available values: Development stage partly, Life form or empty string;
- `updated` — the date the record was saved/updated;
- `created` — the date the record was created;
- `identification_started` — the date a species identification was started;
- `identification_finished` — the date a species identification was finished;
- `country` — country;
- `country_id` — unique id of the country;
- `altitude` — altitude (sea level is treated as zero), this parameter is a string, therefore its form of altitude’s representation might be quite fuzzy: ‘100-300’, ‘100-300 m’, ‘100’, ‘100 m’ etc.
- `region` — region of collection;
- `district` — district of collection;
- `details` — environmental conditions of collection, additional info;
- `note` — everything that wasn’t yet included in the previous fields (this field could include information about the place of collection, details on environmental conditions etc.);
- `dethistory` — an array; history of species identifications for this herbarium record;
- `additional`s — some herbarium records could include more than one species, this array describes all of these;
- `images` — a list of images related to the herbarium record ([] — an empty list, means that no images attached to the herbarium record were found); the list is formatted as follows:
 - `http://...` — first field of image record; it is a path (link), where the image could be downloaded from;
 - image type — allowed values are either ‘p’ or ‘s’; ‘p’ = ‘place’ — the image is related to the place of collection (e.g. s. ‘s’ = ‘sheet’ — snapshot of the herbarium sheet;

- meta information – json-formatted string including auxiliary information about the image; e.g. snapshot authorship, snapshot date, etc. In case of snapshot authorship, sample meta-string would be “{‘photographer’: ‘Pavel Krestov’, ‘organization’: ‘Vladivostok Botanical Garden Institute’}” There is no restriction about names of meta-fields, such as ‘photographer’ or ‘organization’; meta-fields could be arbitrary, but ones having intuitive values are preferred.

List of images attached to the herbarium record (example):

```
[
('http://someresource.com/path/to/image1', 'image1 type', 'meta information1'),
('http://someresource.com/path/to/image2', 'image2 type', 'meta information2'),
...
]
```

Note: Attributes region, district, details, note, altitude could be filled in bilingual mode, that means it could include special symbol “|”. For instance, let’s consider region and its value “Russian Far East|Дальний Восток России”. The region string consist of two parts English and Russian separated by “|”. In current implementation the API service doesn’t care about what part of the string is really needed to the user and returns the entire string. Handling such cases, e.g. removing unnecessary substrings from left or right side of the “|” symbol, should be performed by the user.

Note: Unpublished records are excluded from search results.

Structure of dethistory and additional arrays are described below.

History of species identifications and additional species

History of species identifications

Each item of the array “History of species identifications” (dethistory) describes an attempt of identification/confirmation of the main species related to the herbarium record.

History of species identifications (dethistory) is an array having the following fields:

- valid_from — start date of species assignment validity;
- valid_to — start date of species assignment validity; empty field means that species assignment is actual since the valid_from date;
- family — family name;
- family_authorship — family authorship;
- genus — genus name;
- genus_authorship — genus authorship;
- species_epithet — species epithet;
- species_id — ID of species instance;
- species_authorship — species authorship;
- species_status — species instance status;
- species_fullname — full species name (Genus name + species epithet + species authorship);

Note: If herbarium record/sheet include more than one species, than “history of species identifications” is related to the main species of the record only.

Additional species

Items of the array “Additional species” (additional) describe all species attached to the current herbarium record/sheet and have the following fields (fields have almost the same meaning as for dethistory array):

- valid_from — beginning date of validity of identification;
- valid_to — ending date of validity of identification; empty field means that species assignment to the herbarium record is actual since valid_from date;
- family — family name;
- family_authorship — family authorship;
- genus — genus name;
- genus_authorship — genus authorship;
- species_epithet — species epithet;
- species_id — ID of species instance;
- species_authorship — species authorship;
- species_status — species instance status;
- species_fullname — full species name;
- significance — measure of ambiguity regard the current species (possible values: “”, aff., cf.);
- note — additional information about the current species;

Note: The note field could be filled out with bilingual mode support (e.g. using the “|” symbol); So, it behaves like described [early](#).

Example

Let us consider an example of additional array of the following form (not all fields are shown for short):

```
[
  { 'genus': 'Quercus', 'species_epithet': 'mongolica', ... , 'valid_from': '2015-05-05', 'valid_to': '2016-01-01' },
  { 'genus': 'Quercus', 'species_epithet': 'dentata', ... , 'valid_from': '2016-01-01', 'valid_to': '' },
  { 'genus': 'Betula', 'species_epithet': 'manshurica', ... , 'valid_from': '2015-05-05', 'valid_to': '' },
  { 'genus': 'Betula', 'species_epithet': 'daurica', ... , 'valid_from': '2015-05-05', 'valid_to': '' },
]
```

Interpretation:

So, if today is 2015, 1 Sept, than the array includes Quercus mongolica, Betula manshurica and Betula daurica, but Quercus dentata should be treated as out-of-date for this date.

If today is 2017, e.g. 1 Jan 2017, than out-of-date status should be assigned to Quercus mongolica, and, therefore, actual set of species includes Quercus dentata, Betula manshurica и Betula daurica.

2.4 Service usage limitations

Due to long evaluation time needed to handle each HTTP-request, there are some restrictions on creating such long running keep-alive HTTP-connections (when using the HTTP API Service).

The number of allowed simultaneous connections to the service is determined by `JSON_API_SIMULTANEOUS_CONN` value.

When the number of simultaneous connections is exceeded, the server doesn't evaluate search requests, but an error message is returned.

This behaviour isn't related to search-by-id queries. Search-by-id queries are evaluated quickly and have no special limitations.

Attempt to get data for unpublished record by its ID leads to an error message.

2.5 Examples

To get tested with the service, one can build a search request using web-browser (just follow the links below):

<http://botsad.ru/hitem/json/?genus=riccardia&collectedby=bakalin>

Follow through the link will lead to json-response that includes all known (and published) herbarium records with genus *Riccardia* and collected by bakalin.

Searching by ID (colstart will be ignored):

<http://botsad.ru/hitem/json?id=500&colstart=2016-01-01>

<http://botsad.ru/hitem/json?id=44>

<http://botsad.ru/hitem/json?id=5>

See also:

[Accessing Digital Herbarium using Python](#)

[Accessing Digital Herbarium using R](#)

HERBARIUM RECORD'S CITING

Not yet completed.

C

citing, 17

M

map, 6

R

rectangular area, 6

S

search by region, 6

search form, 3

search in a region, 6

search results filtering, 6