

# USING THE PASTA+ SEARCH API



Duane Costa  
Environmental Data Initiative



# PASTA+ SEARCH API: MOTIVATION FOR USING

- Build a custom search interface
  - Let users query PASTA in a way that differs from the search interface in the centralized data portal. For example, restrict search results to a particular geographic area.
- Local data catalog
  - Automatically synchronized with contents of the central data catalog, avoiding inconsistencies
  - Lower maintenance. Let PASTA automatically retrieve the list of data packages it holds for your local site instead of manually maintaining this list on your own.
- Data Mining (where EML metadata is the data to be mined). A few examples:
  - What is the average number of keywords used in EML metadata?
  - How many LTER sites are using LTER “Core Areas” keywords, and which keywords?
  - How many keywords found in PASTA data packages are found in the LTER Controlled Vocabulary?
- Efficiency
  - A single web service call can return lots of information, and it's *fast*, even for very complex queries. Can often be more efficient than using the search interface provided by the centralized data portal.

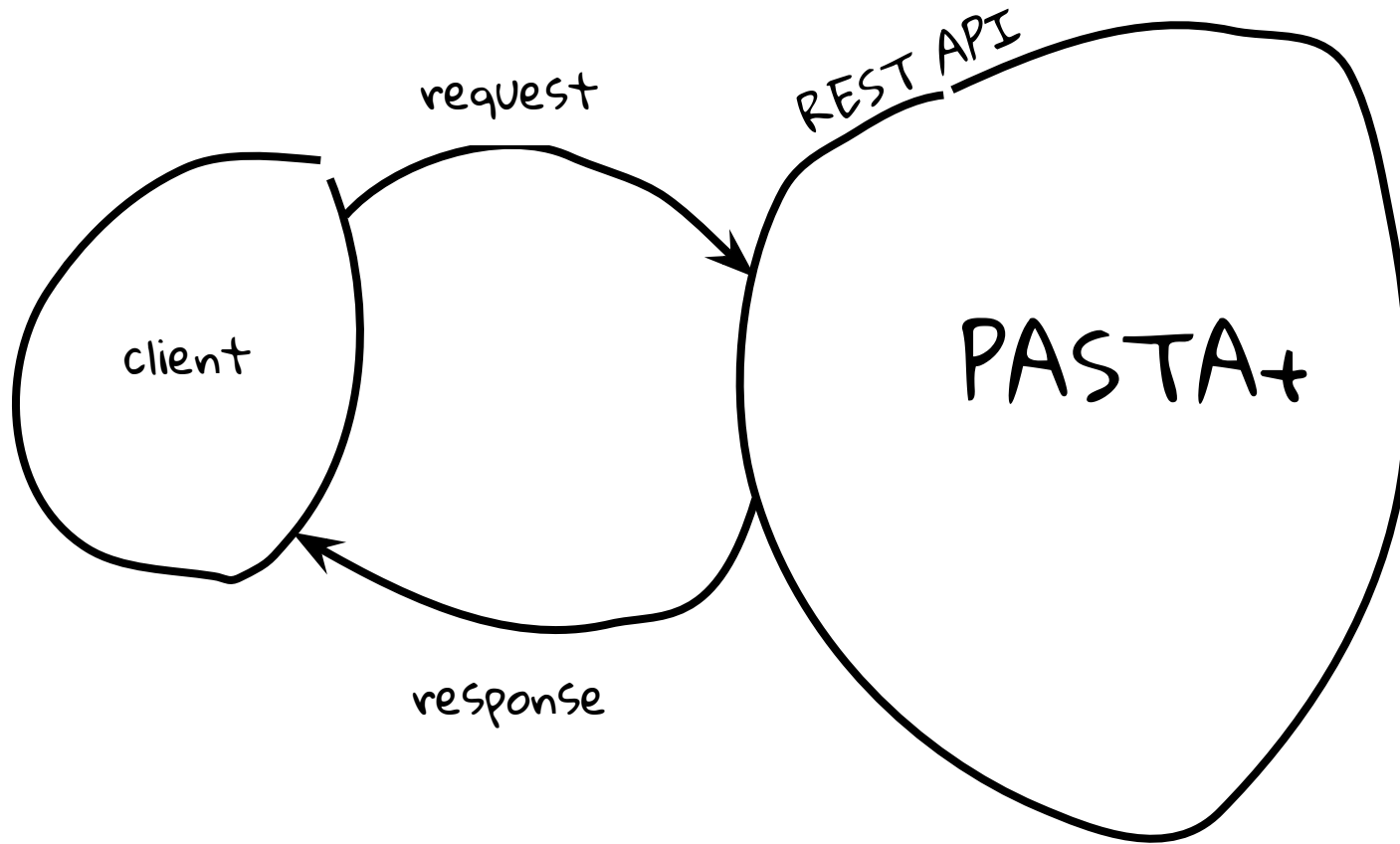
# PASTA+ SEARCH API IS BUILT ON SOLR

Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene<sup>TM</sup>.

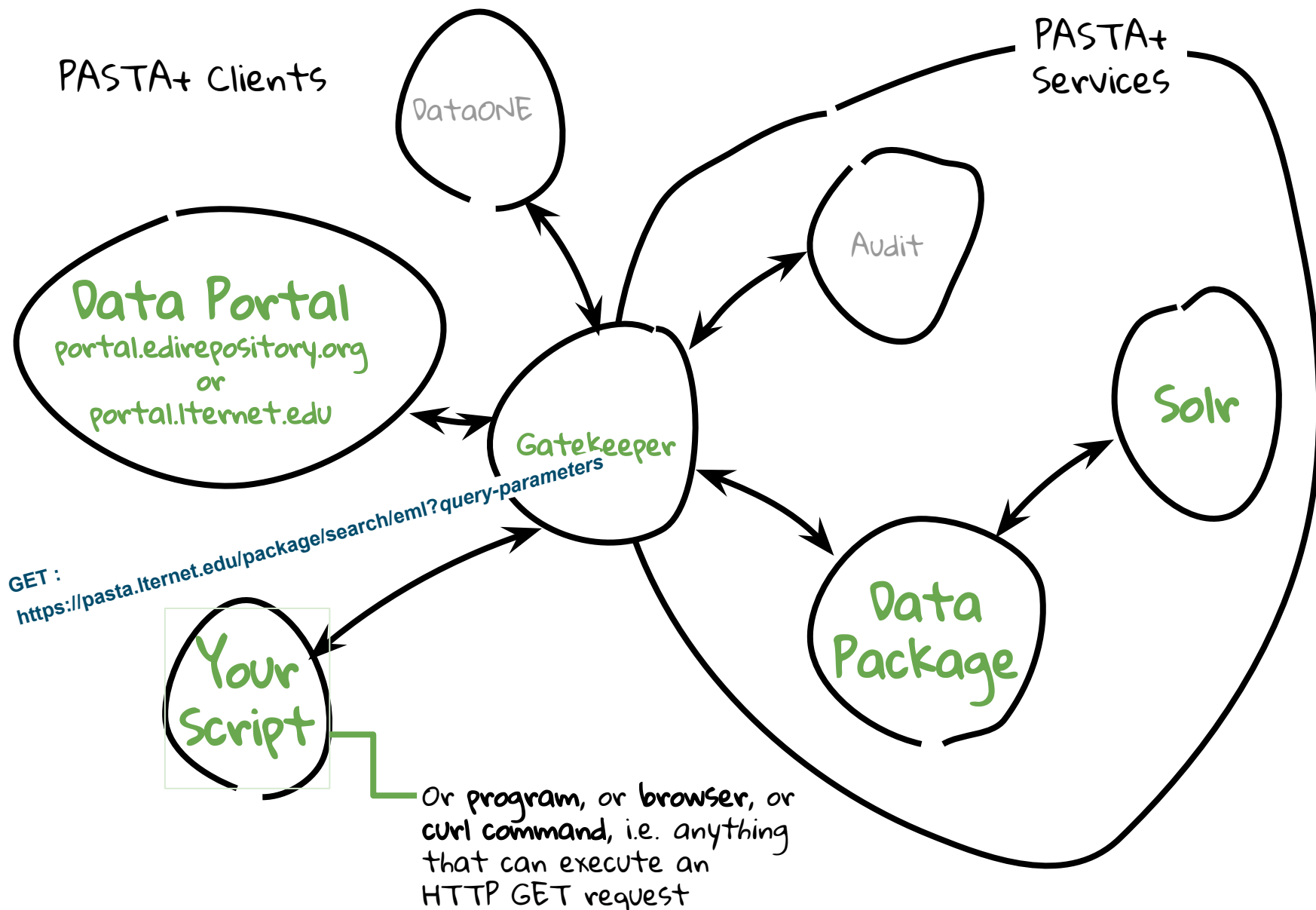
<http://lucene.apache.org/solr/>

# PASTA+ SEARCH API IS JUST ANOTHER WEB SERVICE

(THOUGH I MIGHT ARGUE, KIND OF SPECIAL)



A highly technical diagram



# PASTA+ SEARCH API: THREE SIMPLE STEPS TO UNDERSTANDING

1. Indexing. What metadata fields can I query? Note: not all EML metadata values are indexed. For example:

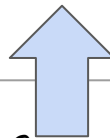
```
<formatString>YYYY-MM-DD</formatString>
```

We have to ask ourselves, would anyone really want to search on "YYYY-MM-DD"? If the answer is "no", then why bother indexing it?

2. Querying. How do I compose a query and send it to PASTA+?
3. Search Results. What does PASTA+ send back to me?

# INDEXING EML METADATA IN SOLR

| Identifiers   | Strings & Text  | Temporal                                      | Spatial     |
|---|---|---|-------------|
| scope (e.g. edi)<br>id (e.g. edi.1)<br>packageid (e.g.edi.1.1)<br>doi | title<br>keyword<br>author<br>organization<br>responsibleParties<br>abstract<br>methods<br>funding<br>taxonomic<br>timescale<br>geographicdescription<br>projectTitle<br>relatedProjectTitle<br>subject | pubdate<br>singledate<br>begindate<br>enddate | coordinates |



These are all the metadata fields we currently treat as searchable content. But, of course, we could add more, if needed.

# QUERYING METADATA: API BASICS

## *Search Data Packages*

GET : <https://pasta.lternet.edu/package/search/eml?query-parameters>



# QUERYING METADATA: FIRST EXAMPLE

<https://pasta.lternet.edu/package/search/eml?q=keyword:disturbance&fl=id>

- `q=keyword:disturbance`
  - Match only data packages that have the “disturbance” keyword
    - Case insensitive
- `fl=id`
  - Return only the id field for each matched document
    - e.g. “edi.1”

Only two query parameters. Easy, right?

# QUERYING METADATA: EXAMPLE 2

**curl -X GET**

```
"https://pasta.lternet.edu/package/search/eml?q=%22air+temperature%22+AND+title:arctic&fq=-scope:ecotrends&fl=packageid,title,score&sort=score,desc&sort=packageid,asc&defType=edismax&start=0&rows=10"
```

- The entire URL is quoted on the command line because “&” has special meaning to the shell.
- Match “air temperature” (as a single search phrase) anywhere in the document AND arctic in the title. Note the “%22” used for double-quotes and “+” used for space. Sometimes we need URL-encoding of certain characters depending on how we’re using the URL.
- Filter query, exclude all documents with a scope value of “ecotrends”
  - Note the negative sign in “-scope:ecotrends”
  - fq is very useful when filtering on spatial or temporal criteria (examples in demo)
- Return the packageid and title fields for each matched document
  - Use fl=\* to return all fields
- Sort on the relevance score, descending order
- Secondary sort on the package identifier, ascending order
- Use the “edismax” query parser (next slide will explain why we would use this)
- Return documents beginning at row 0
- Return only 10 rows of matching documents (note: same as the default rows value)

You can try this example in your browser, too! Copy or type everything between the pair of quotes.

# SOLR QUERY PARSER TYPES: LUCENE VS DISMAX VS EDISMAX

- defType stands for default type, which means it specifies the default query parser type for all queries
- Lucene
  - The standard query parser you get when you omit the “defType” parameter
  - Expressive syntax
  - Fussy about user input, not as user-friendly, can generate errors
- DisMax
  - The Disjunction Maximum parser; defType=dismax
  - Deprecated in favor of the newer eDisMax
- eDisMax
  - Extended Disjunction Maximum parser; defType=edismax
  - Essentially combines the best of Lucene and DisMax
  - Best choice for applications that accept search terms directly from the user

# SEARCH RESULTS XML (1 OF 2)

[https://pasta.lternet.edu/package/search/eml?q=packageid:knb-lter-nin.1.1&fl=\\*](https://pasta.lternet.edu/package/search/eml?q=packageid:knb-lter-nin.1.1&fl=*)

"fl=\*" means "Give me all the fields you've got stored for the matching document(s)."

```
<resultset numFound="1" start="0" rows="10">
  <document>
    <abstract>
      This data package consists of Daily Water Sample Nutrient Data for North Inlet Estuary, South Carolina, from
      1978 to 1992,
      (Truncated to save space)
    </abstract>
    <begindate>1978-09-01</begindate>
    <doi>doi:10.6073/pasta/0675d3602ff57f24838ca8d14d7f3961</doi>
    <enddate>1992-04-21</enddate>
    <funding/>
    <geographicdescription>
      North Inlet encompasses about 2,630 hectares of tidal marshes and wetlands near Georgetown, South Carolina,
      USA. (Truncated to save space)
    </geographicdescription>
    <id>knb-lter-nin.1</id>
    <methods/>
    <packageid>knb-lter-nin.1.1</packageid>
    <pubdate>2013</pubdate>
    <responsibleParties>
      North Inlet LTER NIN-LTER Vernberg, F. John Blood, Elizabeth
    </responsibleParties>
    <scope>knb-lter-nin</scope>
```

# SEARCH RESULTS XML (2 OF 2)


```
<site>nin</site>
<taxonomic/>
<title>
  Daily Water Sample Nutrient Data for North Inlet Estuary, South Carolina, from 1978 to 1992, North Inlet LTER
</title>
<authors>
  <author>Vernberg, F. John</author>
  <author>Blood, Elizabeth</author>
</authors>
<spatialCoverage>
  <coordinates>-79.2936 33.1925 -79.1002 33.357</coordinates>
</spatialCoverage>
<sources></sources>
<keywords>
  <keyword>nutrient dynamics</keyword>
  <keyword>North Inlet Estuary</keyword>
  <keyword>Baruch Institute</keyword>
  <keyword>Georgetown, South Carolina</keyword>
</keywords>
<organizations>
  <organization>North Inlet LTER</organization>
  <organization>NIN-LTER</organization>
</organizations>
<singledates></singledates>
<timescales></timescales>
</document>
</resultset>
```

# ALTERNATIVE METHOD FOR SEARCHING PASTA METADATA: THE "LIST AND READ" APIs

The "list & read" approach. Write a program or script that:

1. Uses the **PASTA list services** to find all the data packages that match the scope.identifier.revision pattern for package identifiers you are interested in searching;
2. For each data package identifier:
  - a. Call the PASTA web service to **read metadata**;
  - b. **Parse the metadata XML**, examining/processing the specific EML elements you are interested in.

# SEARCH API VS "LIST & READ" APIs APPROACH

| Search API Approach   | Alternative Approach ("List & Read")  |
|---|---|
| One & done! One call to the search API returns everything you need  | Could require hundreds or thousands of calls to PASTA web services to list package identifiers and then read individual metadata documents  |
| Fast! Even complex queries execute in a flash   | Could take several minutes to execute   |
| Can query on spatial and temporal criteria  | No built-in capability to query on spatial or temporal criteria   |
| Takes advantage of sophisticated Solr and Lucene query logic  | If you need query criteria, you provide the logic by programming it yourself  |
| Limited to newest revision  | Can access all revisions, not just newest   |
| Limited to queryable subset of EML content (title, keywords, abstract, methods, spatial coordinates, dates & times, DOI, many others) | Can access all EML content, e.g.:<br><b>&lt;formatString&gt;YYYY-MM-DD&lt;/formatString&gt;</b><br><br>We don't index this in Solr |

# PASTA SEARCH API: POTENTIAL FUTURE ENHANCEMENTS

- Index additional EML metadata fields?
  - Perhaps there are some fields we've overlooked that should be indexed. Not too difficult: requires minor code changes, testing, and then an overnight re-indexing of PASTA in Solr.
- Search results returned in Solr native XML format instead of PASTA+ custom XML format
- Other?
  - Please let us know how we can improve the PASTA+ Search API (or any of the other PASTA+ APIs)



# ESIP 2017 SUMMER MEETING

Tuesday, July 25 • 2:00pm - 3:30pm

## Use Application Programming Interfaces of Data Repositories to Create Local Data Catalogs

The Environmental Data Initiative (EDI) data repository is a platform that allows ecological researchers to archive data. However, while the repository provides search, download, and other data cataloging functions that facilitate data discoverability and access, research groups are often required to maintain a local catalog featuring those same data but on a project-specific website. Meeting this need is traditionally addressed by running two parallel systems: (1) the data submitted to the EDI repository, and (2) maintaining a local copy of the data catalog. This approach is inefficient and invites inconsistencies between systems. Although most repositories and DataONE provide APIs to access data in this breakout session, we will discuss and demonstrate how data within the EDI repository may be accessed using the PASTA+ API. The API may be used to harvest data associated with a particular research group, project, or station, which can then be branded and styled for display on a project website. Using this approach, a research group can generate a local catalog of project data by capitalizing on EDI data repository functionality, and avoid the overhead of maintaining two separate data catalogs.

Speakers | Moderators



Duane Costa



Stevan Earl



Gastil Gastil-Buhl



John Porter

# RESOURCES

PASTA Documentation - <http://pastaplus-core.readthedocs.io>

Browse & Discovery -

[http://pastaplus-core.readthedocs.io/en/latest/doc\\_tree/pasta\\_api/data\\_package\\_manager\\_api.html#browse-and-discovery](http://pastaplus-core.readthedocs.io/en/latest/doc_tree/pasta_api/data_package_manager_api.html#browse-and-discovery)

PASTA Data Package API - <https://pasta.lternet.edu/package/docs/api>

Search API - <https://pasta.lternet.edu/package/docs/api#GET:/search/eml>

Jupyter Notebook Examples - [https://github.com/EDIdorg/tutorials/pastaplus\\_webservices/PASTA\\_Plus\\_Search\\_API.ipynb](https://github.com/EDIdorg/tutorials/pastaplus_webservices/PASTA_Plus_Search_API.ipynb)

EDI General Questions - [info@environmentaldatainitiative.org](mailto:info@environmentaldatainitiative.org)

EDI Technical Issues - [support@environmentaldatainitiative.org](mailto:support@environmentaldatainitiative.org)

Books - [Solr in Action](#) by Trey Grainger and Timothy Potter, Manning Publications. 2014.



Powered By  
**PASTA**

THANK YOU, AND HAPPY SEARCHING!

# QUERYING EML METADATA IN SOLR: DEMO

- Browser demo
- Curl demo
- Python demo (Jupyter notebook)