PASTA+ Web Services (a Primer)

by Mark Servilla
Environmental Data Initiative (EDI)
Table of Contents

- Client/server relationship (aka request/response cycle)
- HTTP
- Web services
- PASTA+ architecture
- PASTA+ web services (common methods)
- Discussion/Q&A
Client/Server relationship

Web-browser

request

Web-server
Client/Server relationship

Web-browser

response

Web-server
Client/Server relationship

Web-browser

https://portal.edirepository.org

Web-server
Client/Server relationship

Web-browser

Web-server

(html)
Client/Server relationship

Web-browser

request

Web-server

response
HyperText Transfer Protocol (HTTP)

Web-browser → request ← Web-server
**HyperText Transfer Protocol (HTTP)**

request = Action (verb) + URL + optional content

GET
HEAD
POST
PUT
DELETE
CONNECT
OPTIONS
TRACE
PATCH
HyperText Transfer Protocol (HTTP)

request = Action (verb) + URL + optional content

GET
HEAD
POST
PUT
DELETE
CONNECT
OPTIONS
TRACE
PATCH
HyperText Transfer Protocol (HTTP)

request = Action (verb) + URL + optional content

GET*
HEAD
POST*
PUT
DELETE
**HyperText Transfer Protocol (HTTP)**

request = Action (verb) + URL + optional content

- GET*
- HEAD
- POST*
- PUT
- DELETE

- Headers
- Message Body
HyperText Transfer Protocol (HTTP)

request = Action (verb) + URL + optional content

GET*, HEAD, POST*, PUT, DELETE

Headers
Message
Body

"key-value pairs"
**HyperText Transfer Protocol (HTTP) - request**

**Action (Verb):**
GET

**URL:**
https://pasta.lternet.edu/package/eml/knb-lter-gce/177/22

**Headers:**
Connection: keep-alive
User-Agent: Apache-HttpClient/4.3.3 (java 1.5)
Host: pasta.lternet.edu:443
Accept-Encoding: gzip, deflate
HyperText Transfer Protocol (HTTP)

Web-browser

response

Web-server
HyperText Transfer Protocol (HTTP)

response = Status + optional content

1xx - Information
2xx - Success
3xx - Redirect
4xx - Client Errors
5xx - Server Errors

Headers
Message
Body
HyperText Transfer Protocol (HTTP)

response = Status + optional content

1xx - Information
2xx - Success
3xx - Redirect
4xx - Client Errors
5xx - Server Errors

Headers
Message Body

“key-value pairs”
**HyperText Transfer Protocol (HTTP)**

`response = Status + optional content`

- **1xx - Information**
- **2xx - Success**
- **3xx - Redirect**
- **4xx - Client Errors**
- **5xx - Server Errors**

**Headers**

- 200: OK
- 307: Temporary redirect
- 401: Not authorized

**Message Body**

- 404: Not found
- 500: Internal server error
- 502: Bad gateway
**HyperText Transfer Protocol (HTTP) - Response**

**Status:**
HTTP/1.1 200 OK

**Headers:**
- Server: Apache-Coyote/1.1
- Web-Service: DataPackageManager-1.0
- Content-Type: text/plain
- Date: Fri, 07 Jul 2017 02:48:36 GMT
- Content-Length: 382

**Message Body:**
- https://pasta.lternet.edu/package/data/eml/knb-lter-gce/177/22/dd05325f7f991f7760d6a2ab4c7c6a5a
- https://pasta.lternet.edu/package/data/eml/knb-lter-gce/177/22/e4d5ade74037037b3ad439de27291e8b
- https://pasta.lternet.edu/package/metadata/eml/knb-lter-gce/177/22
- https://pasta.lternet.edu/package/eml/knb-lter-gce/177/22
Web-accessible objects on servers are “resources”; thus, the name Uniform Resource Locators (URL).

Resources can be represented in different ways: XML, JSON, plain text, binary streams, ...

Actions between clients and servers are “stateless”.

“Stateful” action may be simulated with sessions.
Web Services

A web service is a function, program, or application on the Internet that performs some task for another function, program, or application that is also on the Internet.
Web Services

A web service is a function, program, or application on the Internet that performs some task for another function, program, or application that is also on the Internet.

-- Mark Servilla, 2017
Web Services

Application → request → Application
Application ≤ response ≤ Application
Web Services

A web service is a function, program, or application on the Internet that performs some task for another function, program, or application that is also on the Internet.

-- Mark Servilla, 2017

Web services allow platform-independent computing between servers that are connected to the Internet (WWW).
Web Services

Linux, Windows, OS X...

request

response

Linux, Windows, OS X...
Web Services

Java, Python, PHP, C++, Cobol...

Java, Python, PHP, C++, Cobol...

request

response
Web Services

- Remote Procedure Call (RPC) - mid 1990's, used in distributed computing environments and very platform specific; follow-on by XML-RPC
Web Services

- Remote Procedure Call (RPC) - mid 1990’s, used in distributed computing environments and very platform specific; follow-on by XML-RPC
- Simple Object Access Protocol (SOAP) - late 1990’s, developed originally by Microsoft; platform independent, but fairly complex setup
Web Services

- Remote Procedure Call (RPC) - mid 1990’s, used in distributed computing environments and very platform specific; follow-on by XML-RPC
- Simple Object Access Protocol (SOAP) - late 1990’s, developed originally by Microsoft; platform independent, but fairly complex setup
- Representational State Transfer (REST) - early 2000’s, not a protocol, but more of design pattern/style; utilizes existing HTTP protocol
REST

REST = Representational State Transfer
REST

REST = Representational State Transfer

Roy Fielding

REST

REST = Representational State Transfer

“REST is a design pattern for application-to-application processing of web resources based on the HTTP protocol.”
REST

REST = **Representational State Transfer**

“REST is a design pattern for application-to-application processing of web resources based on the HTTP protocol.”

-- Mark Servilla, 2017
REST = Representational State Transfer

“REST is a design pattern for application-to-application processing of web resources based on the HTTP protocol.”

-- Mark Servilla, 2017
REST

REST = Representational State Transfer

“REST is a design pattern for application-to-application processing of web resources based on the HTTP protocol.”

-- Mark Servilla, 2017

“...architectural style...”

-- Roy Fielding, 2000
REST - APPLICATION-TO-APPLICATION PROCESSING

Processing of web resources:
- Create
- Read
- Update
- Delete
REST - APPLICATION-TO-APPLICATION PROCESSING

Processing of web resources:
- Create
- Read
- Update
- Delete

HTTP request actions (verbs):
- POST
- GET
- PUT
- DELETE
REST - APPLICATION-TO-APPLICATION PROCESSING

Processing of web resources:
- Create
- Read
- Update
- Delete

HTTP request actions (verbs):
- POST
- GET
- PUT
- DELETE
REST - APPLICATION-TO-APPLICATION PROCESSING

Processing of web resources:
- Create
- Read
- Update
- Delete

HTTP request actions (verbs):
- POST
- GET
- PUT
- DELETE
REST - APPLICATION-TO-APPLICATION PROCESSING

**Processing of web resources:**
- Create
- Read
- Update
- Delete

**HTTP request actions (verbs):**
- POST
- GET
- PUT
- DELETE
REST - APPLICATION-TO-APPLICATION PROCESSING

Processing of web resources:
- Create
- Read
- Update
- Delete

HTTP request actions (verbs):
- POST
- GET
- PUT
- DELETE
REST - APPLICATION-TO-APPLICATION PROCESSING

Processing of web resources:
- Create
- Read
- Update
- Delete

HTTP request actions (verbs):
- POST
- GET
- PUT
- DELETE
REST - Application-to-Application Processing

Processing of web resources:
- Create
- Read
- Update
- Delete

HTTP request actions (verbs):
- POST
- GET
- PUT
- DELETE

"CRUD" API
REST - HOW TO INTERACT WITH A REST API?

- Web browser (for GET)
- Web browser plugins (for complete enchilada; GET, POST, PUT...)
- Command line applications like CURL
- Programming languages that support Internet connections
PASTA+

client

request

REST API

response

PASTA+
SOA = Service Oriented Architecture
PASTA+ REST WEB SERVICES

PASTA data package unique identifiers:

edi.10.2
PASTA+ REST WEB SERVICES

PASTA data package unique identifiers:

edi.10.2

scope:identifier:revision
PASTA+ REST WEB SERVICES

PASTA data package unique identifiers:

edi.10.2

scope:identifier:revision

String value that identifies the organization, project, or theme of the data package
PASTA+ REST WEB SERVICES

PASTA data package unique identifiers:

edi.10.2

scope:identifier:revision

Integer value that uniquely identifies the data package in the namespace of the scope
PASTA+ REST WEB SERVICES

PASTA data package unique identifiers:

edi.10.2

scope:identifier:revision

Integer value in increasing order that identifies the version of the data package
PASTA+ REST WEB SERVICES

- List operations - Read operations about the structure of the repository, uses HTTP GET
PASTA+ REST web services

- List operations - Read operations about the structure of the repository; uses HTTP GET

- Read operations - Read operations about a specific data package or other resource (audit records or event subscription); uses HTTP GET or HEAD
PASTA+ REST WEB SERVICES

- List operations - Read operations about the structure of the repository; uses HTTP GET

- Read operations - Read operations about a specific data package or other resource (audit records or event subscription); uses HTTP GET or HEAD

- Create/Update/Delete operations - Modifying operations about a specific data package; uses HTTP POST, PUT, or DELETE
PASTA+ REST WEB SERVICES

- List operations - Read operations about the structure of the repository; uses HTTP GET

- Read operations - Read operations about a specific data package or other resource (audit records or event subscription); uses HTTP GET or HEAD

- Create/Update/Delete operations - Modifying operations about a specific data package; uses HTTP POST, PUT, or DELETE
PASTA+ REST web services - List

Data package scopes

https://pasta.lternet.edu/package/eml

Data package identifiers for a given scope

https://pasta.lternet.edu/package/eml/<scope>

Data package revisions for a given scope and identifier

https://pasta.lternet.edu/package/eml/<scope>/<identifier>
curl -i -X GET https://pasta.lternet.edu/package/eml/edi

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Web-Service: DataPackageManager-1.0
Content-Type: text/plain
Date: Sun, 09 Jul 2017 18:35:16 GMT
Content-Length: 39
PASTA+ REST web services - Read

Data package resource map

https://pasta.lternet.edu/package/eml/<scope>/<identifier>/<revision>

Data package EML metadata

https://pasta.lternet.edu/package/metadata/eml/<scope>/<identifier>/<revision>

Data package data file

https://pasta.lternet.edu/package/data/eml/<scope>/<identifier>/<revision>/<entityId>
curl -i -X GET https://pasta.lternet.edu/package/metadata/eml/edi/1/1

HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Web-Service: DataPackageManager-1.0
Content-Type: application/xml
Date: Sun, 09 Jul 2017 18:37:56 GMT
Content-Length: 27795

<?xml version="1.0" encoding="UTF-8"?>
<eml:eml packageId="edi.1.1" system="https://pasta.lternet.edu"
  xmlns:eml="eml://ecoinformatics.org/eml-2.1.1"
  xmlns:stmml="http://www.xml-cml.org/schema/stmml_1.1"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="eml://ecoinformatics.org/eml-2.1.1
  http://nis.lternet.edu/schemas/EML/eml-2.1.1/eml.xsd">
  <access authSystem="https://pasta.lternet.edu/authentication"...
PASTA+ REST web services - Data Package life-Cycle

Step 1: User uploads only data package EML* to PASTA

Step 2: PASTA parses EML and downloads data files from source locations as declared in metadata

Step 3: PASTA performs quality checking against metadata, data, and metadata/data congruency; quality report is generated

Step 4: PASTA registers metadata, data, and report in local registry; a “create” event is recorded in PASTA’s audit log; event notifications are sent for all subscriptions

Step 5: PASTA assigns Digital Object Identifier to data package; DOI registered at DataCite

Step 6: PASTA synchronizes data package with DataONE

*EML = Ecological Metadata Language
PASTA+ REST web services - Create, Update, & Delete

Create data package; EML in request message body

(POST) https://pasta.lternet.edu/package/eml

Update data package; EML in request message body

(PUT) https://pasta.lternet.edu/package/eml/<scope>/<identifier>

Delete data package; all revisions are archived

(DELETE) https://pasta.lternet.edu/package/eml/<scope>/<identifier>
curl -i --user "uid=EDI,o= LTER,dc=ecoinformatics,dc=org:<PASSWORD>" -X POST https://pasta.lternet.edu/package/eml --header "Content-type: application/xml" --data-binary @edi.1.1.xml

HTTP/1.1 100 Continue

HTTP/1.1 202 Accepted
Set-Cookie: auth-token=dWlkPUVESSxvPUxURVIsZGM9ZWNvaW5mb3JtYXRpb24gMTQ5OTY1NDcwNjcwNQphdXRoZW50aWNhdGVk-KDfCxdYfgoXCiH11JYjhpbi+IPJdjoOs1EpWpkXiTzFAgHrbDo6HorbZuLaTTXC1dy1j1j1NO6Q1x2M5XMGxuE4UorPhisSpMU+no52HODYnJjUIe19TtB8vXZjW3pdk8N+REV2cILINeDjxB3SusAeYeQIxe3Ijx/1RZWkWL3Btx4DYknFtWE2+0xYrb8WqcAOXuaQB7Ht2ujRORvTYTjHIiLC+y6B+ABQQswLwvAGfycJtQgZSmry+2UsKdTbNZ2LEAiFKgsnFPAi27MS/szY9pZR3x5TT26C18K1QYQXqHlaPu0yRH5zfPdmr2AcVV+vJExN1RnijNUeU3loqvTDDvNoNL/MWo7vRu4r3RWLtA3UAx6b3HjA8b7oFmykxBP/s60JpxoPTFpjmPaaKmUU4HLYvgfUfgmyBFZ1QcJ+wsez4aAKm?cMGwGQ85bM2O2KFvxx82j4y19z0R/miPUjkgIyyey4xf415k3caxOBVZ0s8BLNryGwVx59kr27kr74rAnyogeL8cKmkHuKj8JjgbARwFqhfzvFXTTPMdIRPegF7KowplM9/D8xmL99u4FDQ/o/nHrxNqr8lnzlwALRcZ83vby/a7sFKLdg/n7JjWxgjXGrIMPMJz4rr7LzQvGQG1iQ3if5r9Yjix608DhzJVv4Y3/qjYkgGY=;Expires=Thu, 15-Jan-2065 21:30:12 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Server: Apache-Coyote/1.1
Web-Service: DataPackageManager-1.0
Content-Type: text/plain
Date: Sun, 09 Jul 2017 18:45:07 GMT
Content-Length: 25

create_149962590731841208
PASTA+ REST web services - User Authentication

Two forms of request authentication:

1. Login credentials using LDAP DN and password bind
   
   `uid=EDI,o=LTERR,dc=ecoinformatics,dc=org`

2. PASTA created authentication token

Otherwise, all requests are accepted as anonymously "public"
PASTA+ REST web services - User Authentication

curl -i --user "uid=EDI,o=LTR,dc=ecoinformatics,dc=org:<PASSWORD>" -X POST https://pasta.lternet.edu/package/eml --header "Content-type: application/xml" --data-binary @edi.1.1.xml

HTTP/1.1 100 Continue

HTTP/1.1 202 Accepted
Set-Cookie: auth-token=dWlkPUVESSxvFUXvURVIgZMG9ZWNvaW5mb3JtYXRpY3MsZGM9b3JnKmh0dHBzOi8vcGFzdGEuzWRpcmVwb3NpdG9yeS5vcmcvYXV0aGVudGljYXRpb24gMTQ5OTY1NdcwNjcwNChdGVk-KDfcXyYgoXCiH11Yjhpb-iPJdjoOs1EpWpkXiTzFAgHrbDo6HorbZuLaTTXc1dy1jj1NO6Q1x2M5XMGxAE4WUorPhSpMU+no52HODYNnjUIE19TtbB8vXZjW3pk8N+REV2cdLINleDjxDBSusAeYeQIxe3Ijx/1RZWkWL3BtXw4DYknFTfWE2+QxYpb8WqcAOXuaQB7Ht2ujROvTYjtiHLC+y6B+ABQQswLvAGfycJtQgZSmry+2UsKdTBnPZ2LEAiFkg5mNfPAi27MS/yz9pZ3x5TT26CI8KlQYQxqHaPu0yRH5zfPdmr2AcV++VjEAxn1RnijNUEU3loqvTDdVNoNL/MWo5vuRi5r3RLtA3UA6x3hJAbB7oFmyxXPi/b60JpoxFTPfpjmPaaKmUU4HIYVgfUGfmyBFpZ1QcJ+wzez4aAKm7cMGwGQ85bM2O2KFVx82j4y1z0R/miuPjkgIyey4Xf45k3caxOY022b8BLNMrYWvVx59kr27kT4rAnyogezL8cMkmHUkI8jJgAwFrcFhbzvFXTTPMdIRREgF7Kowp1M9/D8xmL99u4FQD/o/nHrxNqr8lnzlwALRcZ33vby/amsFKld/gN7JwXgjXGrIMPMJz4rr7LzQvGQG1Q3if5rJ9YjiX608DhzJVvY3/qjYkgGy=;Expires=Thu, 15-Jan-2065 21:30:12 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Web-Service: DataPackageManager-1.0
Content-Type: text/plain
Date: Sun, 09 Jul 2017 18:45:07 GMT
Content-Length: 25

create_149962590731841208
PASTA+ REST web services - User Authentication

curl -i --user "uid=EDI,o=LTER,dc=ecoinformatics,dc=org:<PASSWORD>" -X POST
https://pasta.lternet.edu/package/eml --header "Content-type: application/xml" --data-binary @edi.1.1.xml

HTTP/1.1 100 Continue

HTTP/1.1 202 Accepted
Set-Cookie:
auth-token=dWlkPUVESSxvPUxURVIsZGM9ZWNvaW5mb3JtYXRpY3MsZGM9b3JnKmh0dBzOi8vcGFzdGEuZWRpcmVwb3NpdG9yeS5vcmcv
XV0aGVudGljYXRpb24qMTQ5OTY1NDcwNjcwNphdGVk-KDfCxdYfgoXC1hIlJYjhpbi+iPJdjoOs1EwpkxiTzFAgHrbD0
6HorbZuLaTTXCldy1j1j1NO6Qlx2M5XMGx0E4U0orPhispu9 Username=EDI"+
Expires: Thu, 15-Jan-2065 21:30:12 GMT
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Server: Apache-Coyote/1.1
Web-Service: DataPackageManager-1.0
Content-Type: text/plain
Date: Sun, 09 Jul 2017 18:45:07 GMT
Content-Length: 25

create_149962590731841208
PASTA+ REST web services - Event Notification
PASTA+ REST web services - Event Notification
PASTA+ REST web services - Event Notification

Diagram:
- **EDI 1.1** flows into **PASTA+**
- **PASTA+** sends **EDI 1.1** to **Workflow #1** and **Workflow #2**
PASTA+ REST web services - Event Notification

Diagram showing a server labeled 'PASTA+' connected to workflow #1, workflow #2, and workflow #N through HTTP POST requests.
PASTA+ REST web services - Audit Reporting

curl -i --user "uid=EDI,o=LTER,dc=ecoinformatics,dc=org:<PASSWORD>" -X GET 
"https://pasta.lternet.edu/audit/report?limit=10&resourceId=edi/1/1"

HTTP/1.1 200 OK
Set-Cookie: auth-token=dWlkPUVESSxvPUxURVI...
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Server: Apache-Coyote/1.1
Cache-Control: no-cache
Web-Service: AuditManager-0.1
Content-Type: application/xml
Date: Sun, 09 Jul 2017 20:17:07 GMT
Content-Length: 5222

<auditReport>
<auditRecord>
<oid>25302843</oid>
<entryTime>2016-12-01T13:05:53</entryTime>
<category>info</category>
<service>DataPackageManager-1.0</service>
<serviceMethod>createDataPackage</serviceMethod>
<responseStatus>200</responseStatus>
<resourceId>https://pasta.lternet.edu/package/eml/edi/1/1</resourceId>
<user>uid=NTL,o=LTER,dc=ecoinformatics,dc=org</user>
<groups>authenticated</groups>
<authSystem>https://pasta.lternet.edu/authentication</authSystem>
<entryText></entryText>
</auditRecord>
...

Recap

Web services are application-to-application processing of resources that take place over the Internet.
Recap

REST uses the HTTP protocol to effectively provide web services
RECAP

PASTA+ provides data repository services using REST web services and a SOA architecture.
RECAP

The EDI Portal is a reference implementation of PASTA+ web services

https://portal.edirepository.org
Resources

Read-the-docs - http://pastaplus-core.readthedocs.io

Package API - https://pasta.lternet.edu/package/docs/api

Audit API - https://pasta.lternet.edu/audit/docs/api

PASTA+ on GitHub - https://github.com/PASTAplus/PASTA

Examples - https://github.com/EDIorg/tutorials

Tech Questions - support@environmentaldatainitiative.org
Thank you

Powered By
PASTA