

Obtain Service Genre: *obtain*

Intellectual Property Rights: This document may be used under the *Creative Commons Attribution-ShareAlike 2.5 Australia License* [<http://creativecommons.org/licenses/by-sa/2.5/au/>].

The appropriate attribution for a derivative of this work is: “This document is derived from work created as part of the Federated Repositories for Education (FRED) Project within the Australian ADL Partnership Laboratory. Copyright © University of Southern Queensland and University of Memphis. 2008.”

Introduction

Obtain is the process by which an application fetches or retrieves one or more requested objects from a data source, in a specified form. An obtain service interface for the data source provides a mechanism for the external agent (requestor) to contact the data source to obtain the requested object(s). The requestor *MAY* specify how the requested object is to be presented in response to the request, i.e., the form of the returned object. This includes both the digital representation of the object content (e.g., file format), and the encoding and packaging preferred. The obtained data, returned as a whole object, is generally made available to an external user or application.

This is a general description of an obtain service genre, independent of application end point, resource, data object, or underlying communications protocols and service models. The service genre includes the notion that requested objects *MAY* have different representations, including file formats. The service genre includes a mechanism to authenticate clients. The service genre *MAY* be used in conjunction with authorization methods to control the return and filtering of results. Such authorization *MAY* be included as part of another service genre or as part of a service usage model that combines authorization with obtain.

The words *MUST*, *MUST NOT*, *REQUIRED*, *SHALL*, *SHALL NOT*, *SHOULD*, *SHOULD NOT*, *RECOMMENDED*, *MAY*, and *OPTIONAL* in this document are to be interpreted as described in [RFC 2119].

The service genre description that follows uses e-Framework service genre description elements as of 2006-12-09, updated to include draft e-Framework service classifications. Other terms, e.g., client, provider, data source, are used as defined in the e-Framework.

Items are tagged and identified using names assigned by the FRED project. Formal e-Framework names will be assigned by the e-Framework.

Service Genre Definition

Name

- FRED Service Genre Name: obtain
- FRED ID: hdl:FREDNA/6E2F735AEE7F481ABFED422699E4B9D9
- e-Framework Service Genre Registry Name: TBD

Rationale

The obtain service genre is authored to support content delivery applications relying on the two request parameters of “what do I want to get” and “how I do I want it to be presented to me”. The obtain service genre does not include additional functionality, such as multiple-resolution or content-dependent accessors. Other service genre used to delivery objects have different requests, and different ranges of legal return values. The range of contexts in which this particular service genre is not applicable is detailed under the Applicability element.

Classification

To be provided by the submitter:				
Domain(s)	<input type="checkbox"/> Learning & Teaching	<input type="checkbox"/> Research Libraries	<input type="checkbox"/> Administration IT Services	<input checked="" type="checkbox"/> Common
Maturity	<input checked="" type="checkbox"/> Immature	<input type="checkbox"/> Mature		
Deployment Scale	<input checked="" type="checkbox"/> Isolated	<input type="checkbox"/> Ubiquitous		
To be determined by the e-Framework:				
Status	<input type="checkbox"/> Approved	<input type="checkbox"/> Placeholder	<input type="checkbox"/> Superseded	
		<input type="checkbox"/> Unapproved	<input type="checkbox"/> Withdrawn	
Confidence Level	<input type="checkbox"/> High	<input type="checkbox"/> Medium	<input type="checkbox"/> Low	

Version

- FRED Version: 1.0.1 [hdl:FREDNA/6E2F735AEE7F481ABFED422699E4B9D9]
- e-Framework Service Genre Version: TBD

Version History

Version	Date	Author	Description	Organization/Project
0.1	2007-01-08	DR	Initial version hdl:FREDNA/6E2F735AEE7F481ABFED422699E4B9D9	FRED
0.2	2007-03-23	NN	Added business analysis, further clarification of terminology; included appropriate copy in genre	FRED
0.21	2007-03-27	DR	Editorial, review	FRED
0.3	2007-03-29	NN	Accepted refactoring service genre into Obtain (in which only labels are parameters) and Resolve (which maps generic parameters to labels)	FRED
0.31	2007-04-02	DR	Editorial, review. More uncoupling from appropriate copy.	FRED
0.32	2007-04-02	DR	Had to accept changes – Word lost the ability to format, ...	FRED
0.33	2007-04-04	NN	Editorial	FRED
0.34	2007-04-09	DR	Editorial	FRED
0.35	2007-05-21	NN	Finalise: neutralise FRBR language, present as “what to obtain, and how (what form) to obtain it in”	FRED
1.0.0	2007-05-21	DR	Submission to the e-Framework	FRED
1.0.1	2008-06-18	DR	Editorial to align with LSAL ADL Work	FRED

Description

The obtain service genre provides the mechanism to request objects from a data source. It is an example of a batch-oriented, request-response process. The data source is assumed to be a managed collection of objects, each of which is *retrievable* through a *label*. Each label identifies only one object in the collection. The prime use of the service genre is retrieving one or more of the objects in the collection by specifying their retrieval labels.

This service genre focuses on retrieving information from repositories and other similar data collections of content objects, making them available to external applications and end users. Typically, the data retrieved will be one or more of the objects in a repository. If there are multiple *representations* or *disseminations* of the object available from the repository, any of these MAY be requested. The service genre provides a direct content object accessor for an object (or object representation) in a data collection.

The service genre MAY be specialized in a service expressions to:

- obtain particular types of object representations and disseminations
- specify data packaging and exchange format for the objects returned
- specify communication protocols.

Data sources (collections, repositories) expose an “obtain” interface, defined in the service expressions that specialize this service genre. Clients MAY send requests to this interface to retrieve the objects stored or managed by the data source. The associated service end point will respond to the request with the requested set of objects out of those managed by the data source. The client MAY request only one or a collection of objects. The client MAY also specify one or more acceptable representations of the objects to be obtained. The service end point will determine what object representation (if any) to return in response to the obtain request. The details of the data model packaging and encoding used to return results sets to the client are defined in the service expressions that specialize this service genre.

The obtain service genre is access controlled. A client request will include necessary authorization and authentication credentials such that the obtain service end point MAY permit or deny access to the requested objects. The source of the authorization and authentication credentials is not defined by the service genre. The service end point for the data source is responsible for determining what results it will return, by applying filtering to responses according to authorization-based access control constraints.

Note, as with other service genres, there is no expectation that all service implementations will be interoperable. Different service expressions MAY take different approaches to defining interfaces and data models.

Functionality

The obtain service genre supports only a single function to retrieve objects from the data source.

A request SHALL specify:

- What objects to retrieve (label) [REQUIRED].
- Authentication credentials [REQUIRED].

A request MAY specify:

- What representation of the objects to present to the requestor (i.e., which representations or disseminations to return) [OPTIONAL].
 - Representations SHOULD refer to existing standards and SHOULD include version numbers or version information.
- How to package and encode results [OPTIONAL].
 - Packaging and encoding SHOULD refer to existing standards and SHOULD include version numbers or version information.

Mechanisms MAY exist to provide flow control so that large results sets are returned in chunks.

Requests for multiple objects are treated as a batch; all request and result control settings SHALL be applied uniformly to all objects in the batch.

No other functionality is defined. The functionality that is defined MAY be extended. Major new or additional functionality SHOULD NOT be included; extended capabilities SHOULD be included in other service genres.

Usage Scenarios

Obtain services are typically provided by an application that manages content objects, such as a repository. Obtain services are expected to be used in conjunction with a discovery service (such as browse or search). The objects to be retrieved by the obtain service must first be discovered; the user typically does not know beforehand which objects they wish to retrieve, and (more importantly) what labels those objects should be retrieved through.

Scenario A: Single object obtain by a label (either an extrinsic or an intrinsic attribute).

- End user searches for all objects in the data source written by Dante.
- Search returns metadata of matching objects:
 - a. {Author: Dante Aligheri; Title: Divine Comedy; Label: 823490}

- b. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: English; Label: 823494}
- c. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: Italian; Label: 823495}
- End user requests obtain of object with Label 823490 from the data source.
- Data source returns object to user in default representation (PDF) and default packaging/encoding (none).

Scenario B: Single object obtain by identifier, specifying representation.

- End user searches for all objects in the data source written by Dante.
- Search returns metadata of matching objects:
 - a. {Author: Dante Aligheri; Title: Divine Comedy; Label: 823490}
 - b. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: English; Label: 823494}
 - c. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: Italian; Label: 823495}
- End user discovers which representations are available.
- Allowed representations are PDF, DOC, HTML.
 - a. Alternative. 1: Each of the representations is a statically stored digital object in the data source. However, the representations are not presented to the user as independently accessible objects to retrieve: they are only presented as representations of object 823490.
 - b. Alternative. 2: Each of the representations is dynamically generated from an underlying representation of object 823490. The representations are again not presented to the user as available for retrieval independently of object 823490.
- End user requests obtain of object with Label 823490 and representation DOC from the data source.
- Data source returns object to user in specified representation (DOC) and default packaging/encoding (none).

Scenario C: Single object obtain by identifier, specifying packaging/encoding.

- End user searches for all objects in the data source written by Dante.
- Search returns metadata of matching objects:
 - a. {Author: Dante Aligheri; Title: Divine Comedy; Label: 823490}
 - b. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: English; Label: 823494}
 - c. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: Italian; Label: 823495}
- End user discovers which packaging options are available.
- Allowed packaging/encoding options are ZIP, TAR, MIME multipart/mixed.
- End user requests obtain of objects with Label 823490 from the data source, packaged as a ZIP file..
- Data source returns object to user in default representation (PDF) and specified packaging/encoding (ZIP).

Scenario D: Batch object obtain by identifier, specifying packaging.

- End user searches for all objects in the data source written by Dante.
- Search returns metadata of matching objects:
 - a. {Author: Dante Aligheri; Title: Divine Comedy; Label: 823490}
 - b. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: English; Label: 823494}
 - c. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: Italian; Label: 823495}
- End user discovers which packaging options are available.
- Allowed packaging/encoding options are ZIP, TAR, MIME multipart/mixed.
- End user requests obtain of objects with Label 823490, 823494 and 823495 from the data source, packaged as TAR.
- Data source returns objects to user in default representation (PDF) and specified packaging/encoding (TAR).

Scenario E: Single object obtain by identifier in federation.

The service genre provides a direct content object accessor for any content repository that is part of the federation. Combined with discovery from a federation registry, obtain provides the mechanism to get the discovered object.

- End user searches on the federation registry for all objects in the federation written by Dante.
- Search returns metadata of matching objects:
 - a. {Author: Dante Aligheri; Title: Divine Comedy; Label: 823490; Data source: α }

- b. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: English; Label: 823494; Data source: β }
- c. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; Language: Italian; Label: 823495; Data source: β }
- End user issues requests for obtain of object with Label 823490 to content repository α .
 - a. In this model, the specific content repository hosting the content is part of the metadata ingested by the central registry, and the end user must issue their retrieval request directly to that content repository.
- Content repository α returns object to user in default representation (PDF) and packaging (none).

The obtain service genre only deals with labels of objects in a single data source. A federation of data sources may present itself through an obtain interface as a single data source, but internally the request needs to redirect the access request to a specific participating data source. Internally it uses a Resolve To Label + Location service genre, which identifies the appropriate participating data source holding the object, and possibly the label for that object specific to the participating data source. (This maybe used in a retrieval service which knows about the requestor context; see discussion under Applicability element.) Identifying the location and local label is outside the scope of the obtain service genre.

Scenario F: Single object obtain by identifier in multi-location data source.

- End user searches on the federation registry for all objects in the federation written by Dante.
- Search returns metadata of matching objects:
 - a. {Author: Dante Aligheri; Title: Divine Comedy; ID: 823490; Location: α, β, γ }
 - b. {Author: Dante Aligheri; Title: De Vulgari Eloquentia; ID: 823494; Location: β, γ, δ }
- End user issues **request for obtain** of object with ID 823490 to the central retrieval service which knows about the requestor context.
 - a. In this model, the user does not issue the request to a particular location or repository, since they only ever see a single service genre interacting with the composite data source as a whole. The central instance of the service needs to decide which location the request should be routed to.
- The retrieval service instance determines that β is the best location for the data source to be retrieved from through a **Resolve To Label request**.
- The central retrieval service issues a request to obtain on behalf of the user to β , using the label returned by the Resolve To Label request.
- Content repository β returns object to the central obtain service in the default representation (PDF) and packaging (none).
- The central obtain service returns the object in the default representation (PDF) and packaging (none).

Applicability

The service genre is applicable for retrieving any defined data representation or dissemination for any specific object(s) managed by the data source. Objects have unique labels used for retrieval. Within the service genre, these labels are scoped only to the data source. For example, there is no requirement that the label be a unique identifier which is resolvable to the item within the data source.

A single representation and a single encoding and packaging is specified by the service genre as applying to all returned objects in the batch. The service genre is not applicable where distinct representations, encodings or packaging is required for returned objects in the same batch, as a result of a single service request.

The obtain service genre does not provide a mechanism for search. Retrieval is direct by label.

It is possible that an intrinsic attribute is used as a label (e.g., the author field has been identified in the service expression as a legitimate label for retrieval); but the coincidence of a direct mapping between retrieval labels and object attributes is incidental: only the uniqueness of the label is relevant to the service genre, not its meaning.

Use of an obtain service expression presupposes that the requestor and the service provider have a common understanding of the retrieval label. If the client has access to metadata attributes uniquely specifying the object but which are not defined by the service expression as retrieval labels, the service genre is not applicable (e.g., a user

knows an object's ISBN but items MAY only be retrieved by DOI). The client must convert the attributes to the appropriate retrieval key.

The service genre does not define the meaning of labels used to obtain objects. Service expressions that specialize this service genre MAY choose to align labels with meaningful attributes of objects, but such alignment is not a requirement.

Behaviour of the service genre is not defined when the obtain request has a single label mapping to multiple objects.

Behaviour of the service genre is defined when the data source requires authorization or access controls that are based on authorization credentials included in the service request.

Behaviour of the service genre is not defined when the data source requires its own authentication processes to permit retrieval.

Behaviour of the service genre is not defined if communications need to be secure.

The obtain service genre is distinct from other accessor service genre with different requests or return types:

- The obtain service genre is applicable when the user specifies representation and packaging as part of the service request. A different accessor service genre is applicable when the representation and packaging cannot be specified in the request, and when the simple data model of objects and object representations cannot be presumed. Such a service genre maps a label directly to a single digital object, with no choice of representation or encoding.
- The obtain service genre is applicable when the user specifies a single label for the object within a single data source. Other accessor service genres are applicable when the object may be stored in multiple locations or in different representations, and the business logic of the service genre is used to decide which object instance is returned.
- The obtain service genre is applicable when the user specifies the requested object alone. Other accessor service genres are applicable when the user also specifies other context entities through the request, such as information about the requestor or the service itself, with the expectation that this contextual information will determine which object instance is returned.
- The obtain service genre is applicable when each requested object maps to a unique object delivered in the response set. Other accessor service genres are applicable when the user expects to discover what all the available instances of the requested object are.

Requests & Behaviours

The format and definitions for requests and responses SHALL be defined by the service expressions that specialize the service genre. Requests and behaviours SHALL meet the following conditions:

- One *Obtain Request* SHALL be defined:
 - The request SHALL be capable of retrieving a single object given its label.
 - The request SHALL be capable of obtaining any labelled object from the collection managed by the data source.
 - The request MAY be capable of retrieving a set of objects given a set of labels. Each label identifies a single object to be retrieved.
 - The request MAY permit the client to specify one or more acceptable representations for the requested objects.
 - The same representation SHALL be returned for all objects.
 - The request MAY permit the client to specify the data encoding or packaging for the requested objects.
 - The same packaging or encoding SHALL be returned for all objects.
 - The request MAY specify contextual information, to allow the service genre to determine which object instance and/or representation should be retrieved.
 - The request SHALL include authentication credentials used to control access to the service end.
- Responses SHALL include error indicators or other needed control information:
 - Error indicators SHALL be available for the request as a whole (e.g., the requested representation is unknown).

- Separate error indicators SHALL be used to describe the availability of the results.
- If the request is for multiple objects, error responses about the availability of the results SHALL be at the level of individual objects (e.g., for any requested object, the object is not available or the label is invalid).

Use & Interactions

The model for a client to interact with a service implementation end point SHALL be defined by the service expression that specialize the service genre.

Structure

The service genre assumes the following logical data model for accessed and retrieved objects:

- Objects are within collections in a managed *data source*. The physical manifestation of storage of the objects is hidden behind the service interface.
- Each object has a unique label.
- *Labels* or *access keys* are attributes of an object (possibly a composite set of attributes that MAY be treated as a single surrogate), that discriminate any object from all the other distinct objects held in the data source:
 - A label uniquely identifies an object in the context of a data source.
- Each object MAY have one or more digital *representations* or *disseminations*:
 - Different representations MAY be stored as static objects or they MAY be generated from the same underlying object on the fly in response to a request. Different representations of an object MAY be considered as distinct objects by other systems (e.g., they MAY their own labels). No further assumptions on relations between objects or representations of objects are made.
- Representations of the content of the object at different levels of specificity are different objects (e.g., full text, abstract, metadata record). The service genre does not assume a fully elaborated data model such as FRBR.
- Each object MAY have one or more *encodings*:
 - An object MAY need to be encoded or packaged according to some protocol or standard. This can include ZIP, TAR, IMS CP for packaging, and BINHEX, URL-Encode, UTF-8 for encoding.
- An obtain request for an object is a request for an object or objects to be transmitted to the requestor from the data source. The requested object representation and encoding is considered an object to be returned to the client.
 - The objects SHALL be packaged and encoded according to the request, which MAY be specified by the requestor or a default packaging and encoding MAY be returned if not specified.
 - The representation or dissemination returned MAY be specified by the requestor or a default representation MAY be returned if not specified.
- Additional encodings MAY be required to support transport-level message exchange.

The structure of the service genre SHALL be defined by the service expressions that specialize the service genre.

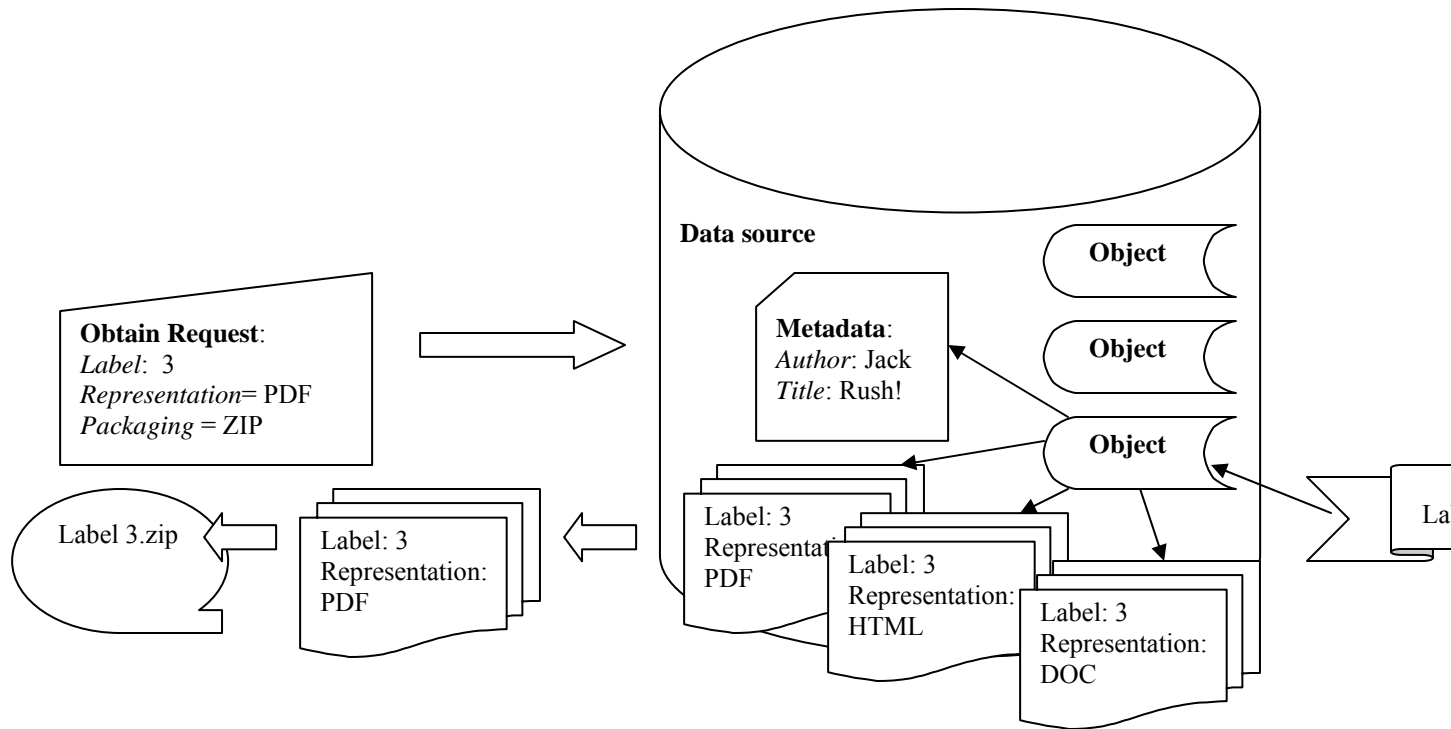


Illustration: Obtain request, specifying representation and packaging.

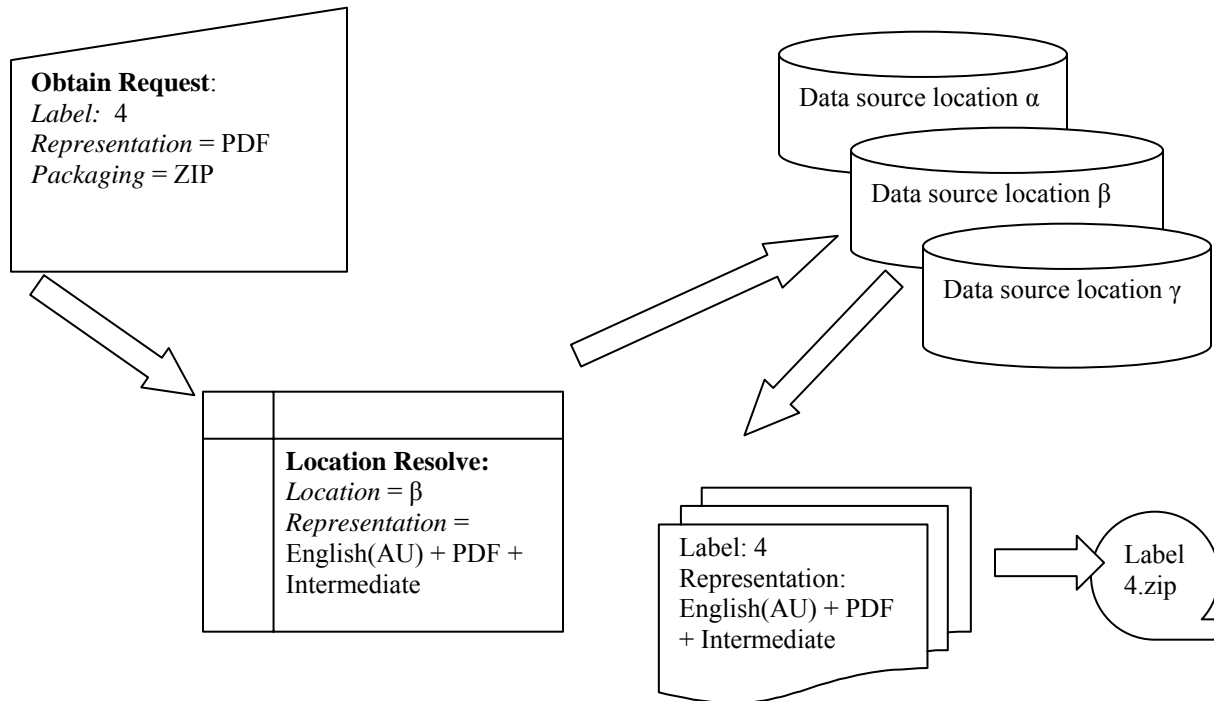


Illustration: Obtain request, orchestrated with service selecting data source (cf. Scenarios E, F)

Applicable Standards

None. No standards are directly applicable to the service genre as a whole.

The service expressions that specialize the service genre SHALL be defined in terms of standards:

- Service expressions SHALL specify applicable digital representation standards for the returned objects.
- Service expressions SHALL specify applicable data encoding and packaging formats for the returned objects.
- Service expressions SHALL specify applicable communications, encoding and transport protocols.
- Service expressions SHALL specify applicable standards-based authentication models.

Design Decisions & Tradeoffs

The process of associating a label with an object, or the meaning of labels, is not defined; the service genre only defines interfaces and observable behaviours, not internal models and structure.

The service genre supports batch processing, with all elements of the batch treated as a set. Since it uses request-response messaging, requests for single objects may be made. If necessary a simplified separate service genre that returns only a single object at a time may be defined. Alternatively, as defined, a service expression may specialize the service genre to return only a single object.

Authentication is an essential part of this service genre.

The service genre could include an *Inspection* request. The *Inspection* request returns to clients the range of permissible values for service request parameters (representations and encodings in particular). This service genre does not include such a request.

Implementation Guide & Dependencies

The following design decisions apply to the service expressions that specialize the service genre.

Structure and Design:

- The service expression MAY include flow control for managing results. The service expression SHALL define whether flow control is supported and limits on request size and results sets.
- Any label that uniquely identifies an object within a data source MAY be used as a retrieval label.
- A label may be an *intrinsic* or an *extrinsic* attribute of an object. An intrinsic attribute reflects meaning about the object; an extrinsic attribute has no semantic attachment to the object, and can be arbitrarily changed. A label being intrinsic or extrinsic, however, has no impact on use to access an object.
- The choice of retrieval label is made while developing the service expression. Extrinsic attributes of the object (e.g., semantically opaque identifiers) are a better choice as labels than intrinsic attributes (e.g., timestamp) as intrinsic attributes are more persistent.
- Certain classes of extrinsic labels, such as file locations, are fragile, being dependent on object state, and may not have long-term persistence. Such labels SHOULD NOT be used as retrieval labels in contexts requiring long-term persistent of retrieval keys.
- The label only needs be unique within the scope of the data source. It need not be globally unique.
- The label need not be externally actionable: it may be defined as a retrieval key through an ad hoc, one-off arrangement between requestor and provider, with no expectation that the label will be stored and managed externally; be actionable by any other external service; or survive past the current obtain transaction.
- All design decisions related to the meaning of retrieval labels and their association with objects are deferred to the design of the service expression.
- While the service implementation accesses a specific data source, the data source itself MAY be a separately managed data source. A service expression MAY specify that the data source is a single physical storage system or it MAY specify that the data source is a distributed collection. The data source MAY be managed by a separate collection of storage manager services. All design decisions related to realisation of the data source are deferred to the design of the service expression.
- Transport-level encodings (e.g., UTF8 over HTTP) of results SHALL be defined by the service expression that specialise the service genre.

Consistency:

- The service implementation SHALL ensure that all labelled (identified) objects and all representations managed by the data source are obtainable.

Performance:

- A service implementation SHALL be capable of handling simultaneous requests from different clients.
- A service implementation SHOULD implement an indexing scheme or equivalent method to permit efficient access to the requested object.
- Load balancing SHOULD be implemented for large resources or those that are accessed frequently (continuously).
- Flow control SHALL be implemented in contexts where returned representations are sizeable, or when large sets of objects are retrieved. Large results sets MAY be broken up over multiple request-response pairs.

Security and Privacy Considerations:

- Service implementations MAY be subject to denial-of-service attacks.
- Care should be taken to maintain privacy of any personal data or other records that MAY disclose usage patterns.
- The client should not be able to discern existence of access-controlled objects by examining error codes.

Know Uses

Actual: The service genre is specialized in the D³UI development-to-deposit prototype using *Reload* to retrieve content from a repository.

Actual: The service genre is specialized in the D³UI discovery-to-delivery prototype using the *ADL SRTE* to retrieve content from a repository.

Potential: The service genre could be used in any content management or content creation process to retrieve content from a managed resource (resource collection).

Potential: The service genre could be used in a service usage model for a repository federation. The service genre supports retrieval of content into any of the resource components of the repository federation (repositories, registries).

Service Genre Dependencies

authenticate: Vx.xx.

[\[link to service genre\]](#)

The service genre requires authentication for use. The authenticate service genre is part of the identity service usage model [\[link to service usage model\]](#). The identity service usage model is an integrated set of service genres that provides the operational user identity infrastructure that includes the creation of user rights and roles, user authentication and user authorization.

authorize: Vx.xx.

[\[link to service genre\]](#)

The service genre uses authorization to filter results. The authorize service genre is part of the identity service usage model [\[link to service usage model\]](#). The identity service usage model is an integrated set of service genres that provides the operational user identity infrastructure that includes the creation of user rights and roles, user authentication and user authorization.

Related Service Usage Models

(FRED) repository federation: V1.0.0.

[\[link to service usage model\]](#)

Obtain (genre) is a part of the repository federation service usage model (genre based) and is used to retrieve content from the repositories and collections that participate in the federation. The repository federation service usage model provides an integrated set of service genres used to populate and use the metadata registry that supports a repository

federation. Functionality includes content management (creation and management of metadata objects within a repository federation), content discovery (discovery of content objects from a repository federation) and content delivery (retrieval of and access to content objects discovered through a repository federation)

(ADL-R) repository federation: V1.0.0.

[[ADL-repository-federation-sum-v100](#) / hdl:1870/E2FE4AD428A1468FA284E270245F72D7]

The (ADL-R) repository federation service usage model is a derivative subset of the (FRED) repository federation service usage model. For the purposes of this service genre they are interchangeable.

(D³UI) content authoring: V1.0.0.

[D³UI content-authoring-sum-v100 / hdl:1870/3FD212C661F1464AA79DBB22D3DE00FF]

Obtain (genre) is part of the (D³UI) content authoring service usage model (genre based) and is used within an authoring and content deposit workflow to retrieve content from a repository.

(D³UI) delivery: V1.0.0.

[D³UI delivery-sum-v100 / hdl:1870/01D0643B992B489ABEB4D89806918450]

Obtain (genre) is part of the (D³UI) delivery service usage model (genre based) and is used within a discovery-to-delivery workflow to retrieve content from a repository.

(FRED) roap obtain: Vx.xx.

[[link to service usage model](#)]

The roap obtain service usage model is a combination of four services used to access an object given an object identifier. A request, in the form of an identifier, is resolved [R] (including multiple resolution based on a FRBR model to obtain the appropriate work, expression, manifestation or copy) yielding the source data source that manages the object. An obtain accessor [O] is used to get the object, subject to access control filtering [A]. The resulting object representation is encoded and packaged in the requested format [P].

repository storage manager: Vx.xx.

[[link to service usage model](#)]

The repository storage manager service usage model is an integrated set of service genres that provides access operators (CRUD, versioning, audit, ACID transactions) for a stored data collection, e.g., the metadata registry, collection registry, repository registry, archival repository, escrow repository.

Related CORE SUMs

None.

References

None.