Morphological Analysis Service Contract Description - v1.1.1

Brief description

Version

This page describes the Morphological Analysis Service API, v. 1.1.1

To discover the version and other metadata about deployed service code that fulfills this API, please utilize the *Service Catalog Service*.

Overview and Definitions

The Morphological Analysis Service responds to requests for morphological analysis of texts, submits them to the appropriate morphology engine for processing and returns the results in XML adhering to a standard morphology schema. The Service supports retrieval of texts for analysis from remote repositories as well as user-supplied chunks of text. URL based and CTS repositories are supported. Where retrieval from a CTS enabled repository is requested CTS URNs are supported as document identifiers. Where retrieval from a URL based repository is requested, URIs are supported as document identifiers.

For a full description, visit the Morphological Analysis Service Description and Assumptions - v1.1.1 page of this wiki (a child to this page).

Assumptions

None.

References

${**REPOSITORY_ROOT**} is http://svn.code.sf.net/p/projectbamboo/code/

- Entity Diagrams
- Codebase: ${REPOSITORY_ROOT}/platform-services/morphology-service/trunk/morphology-service/

Known Issues

- BSPSVC-102 Does not yet support the ignore accents feature of the Morpheus engine
- BSPSVC-87 Cannot configure which TEI tags to ignore when tokenizing text for analysis

ROA Layer API

RESTful service methods performing Create and List functions over Analyses, Morphology Engines and Text Repositories.

Base URLs

It is assumed in this documentation that no centrally-run instances of the Bamboo Services Platform will be running after the project ends on 31 March 2013. Therefore, base URLs are assumed to be on a developer's machine, localhost. The base URL with a port number assumes that the BSP is running unsecured; the URL without a port number assumes that security is enforced and Apache Web Server is intercepting and redirecting service calls. Please see the page *Identity and Access Management - Authentication and Authorization* for context, as well as links to installation and configuration instructions for secured instances of BSP.

Note that *ONLY services at v0.9 or greater* will run properly in a secured instance of the BSP.

Currently available base URLs:

- http://localhost:8181
- http://localhost/services

Client Responsibilities in Requests to Secured BSP Web Services

As of March 2013, policy for the Morphological Analysis Service permits anonymous access.

This section of the Service API documentation describes a client application's responsibilities when making requests to secured Web Services hosted on the Bamboo Services Platform (including this service).

A secured instance of the Bamboo Services Platform (BSP) implies a significant set of installation and configuration tasks for which the operator of the BSP is responsible. These are described in overview on the wiki page *Identity and Access Management - Authentication and Authorization* , and in detail on pages linked from that one.

(1) A client must be configured as a Trusted Application if requests are to be treated other than as *Anonymous*

A client application – whether a web app or a simple testing client such as Firefox Poster or curl – may make requests anonymously or as a "Trusted Application." Only a Trusted Application may assert the identity of a user on behalf of whom the request is made, and scoped roles to be assigned to that user; Bamboo Services trust such clients to assert the identity and assigned-roles *only* of users who have authenticated in the current session of application activity. (A special-case type of client application, termed *Innovation Licensed* applications, are trusted to assert the identity of and roles assigned to a fixed set of special-case users without those users having to authenticate in the current session.)

Configuration of client applications are described in detail in this wiki page: *Configure Apache Web Server for Client Auth*. It is assumed in #2, below, that this configuration has been performed.

(2) A Trusted client is expected to pass HTTP Request Headers to identify itself and an authenticated user

A client application that is *Trusted* in the security context of the Bamboo Trust Federation (cf. *Identity and Access Management - Authentication and Authorization*) must augment each request to a service hosted by a secured instance of the Bamboo Services Platform (BSP) with a set of HTTP headers, as follows:

- **X-Bamboo-AppID**: A UUID that identifies the client research environment, application, tool, or service; this UUID is issued as part of the process of *registering* a trusted client in the Bamboo Trust Federation as described in overview on the page *Identity and Access Management - Authentication and Authorization*; and in detail with respect to physical establishment of trust on the page *Configure Apache Web Server for Client Auth.* The value of this header is linked to the X.509 certificate by which the application establishes an SSL connection to the BSP host in the registration process, and a match between this *Application ID* and the linked X.509 certificate is checked by the BSP on receipt of *every* request.

- **X-Bamboo-BPID**: A UUID that identifies the logged-in user on whose behalf the request is being sent; this value, a Bamboo Person Identifier, or BPId, is obtained via a call to the *Person Service* that occurs in time between user login and any other service request. See the *Read a BambooPersonId* method of the *Person Service API* for details. [†]
- **X-Bamboo-Roles**: A pipe-delimited (|) set of scoped roles asserted by the trusted client to belong to the logged-in user, of the form role@domain, which are to be evaluated as factors in the determination of whether the request satisfies policies (access restrictions) that apply to the requested resource. If a user is *authenticated*, the client is expected to include the role *undefined@domain* where *domain* identifies the organization that authenticated the user (example: *undefined@google.com* is a client app's assertion that the user authenticated to Google). This header is otherwise optional (depending on policies governing the requested resource that may require one or more scoped roles for access to be permitted). Example of multiple roles asserted in this header: *roleA@domainOne.xxx|roleB@domainTwo* [‡]

[†] The value of X-Bamboo-BPID is set to the identifier for the application itself (X-Bamboo-AppID) when a client application calls the Person Service to *create* a new Bamboo Person Identifier; or to *retrieve* the BPId for a user based on the identifier of the IdP with which she has logged in and an SHA-256 hash of that IdP's user identifier for the logged-in person.

[‡] Policies and policy evaluation are described on the page *Authorization and Policy*. Also see *Conventions for representing Identity Providers in the Bamboo Trust Federation*.

Schema

${**REPOSITORY_ROOT**} is http://svn.code.sf.net/p/projectbamboo/code/

- EngineListRepresentation schema: ${REPOSITORY_ROOT}/platform-services/morphology-service/trunk/morphology-service/resource/src/main/resources/EngineListRepresentation.xsd
- TextRepositoryListRepresentation schema: ${REPOSITORY_ROOT}/platform-services/morphology-service/trunk/morphology-service/resource/src/main/resources/TextRepositoryListRepresentation.xsd
- alpheios lexicon schema

Morphology Engine

A morphological engine available for use by the service when producing morphological analyses.

List Engine instances

Lists the set of morphology engines configured for this instance of the service.

Calling Method and Arguments

Invoked as an HTTP GET method. Send an HTTP request of the form:

```
GET /morphologyservice/engine
```

Parameter	Supported Values
Accept Header	application/xml,text/xml,application/json

Response

On success, a response with a "202 Accepted" HTTP status code will be returned:

Parameter	Meaning
HTTP Body	An instance of an EngineListRepresentation XML document

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the Get request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested document does not exist and/or the requested analysis returned no results
500	Internal Server Error	A service error occurred

List an Engine

Lists a specific morphology engine configured for this instance of the service.

Calling Method and Arguments

Invoked as an HTTP GET method. Send an HTTP request of the form:

```
GET /morphologyservice/engine/{EngineId}
```

Parameter	Supported Values
Accept Header	application/xml,text/xml,application/json

Example:

```
GET /morphologyservice/engine/morpheus HTTP/1.1
```

Response

On success, a response with a "202 Accepted" HTTP status code will be returned:

Parameter	Meaning
HTTP Body	An instance of an EngineListRepresentation XML document

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the Get request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested document does not exist and/or the requested analysis returned no results
500	Internal Server Error	A service error occurred

Text Repository

A repository from which texts/documents can be retrieved for analysis. The 1.1.0 version of the services supports both CTS repositories and repositories which can respond to URI based requests.

List Repository instances

List the text repositories configured for this instance of the service.

Calling Method and Arguments

Invoked as an HTTP GET method. Send an HTTP request of the form:

```
GET /morphologyservice/repository
```

Parameter	Supported Values
Accept Header	application/xml,text/xml,application/json

Response

On success, a response with a "202 Accepted" HTTP status code will be returned:

Parameter	Meaning

HTTP Body	An instance of an TextRepositoryListRepresentation XML document

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the Get request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested document does not exist and/or the requested analysis returned no results
500	Internal Server Error	A service error occurred

List a Repository

List a specific text repository configured for this instance of the service.

Calling Method and Arguments

Invoked as an HTTP GET method. Send an HTTP request of the form:

```
GET /morphologyservice/repository/{RepoId}
```

Parameter	Supported Values
Accept Header	application/xml,text/xml,application/json

Example:

```
GET /morphologyservice/repository/xmlRepository HTTP/1.1
```

Response

On success, a response with a "202 Accepted" HTTP status code will be returned:

Parameter	Meaning
HTTP Body	An instance of an TextRepositoryListRepresentation XML document

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the Get request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested document does not exist and/or the requested analysis returned no results
500	Internal Server Error	A service error occurred

Analysis (Word)

Produce a morphological analyses of a supplied word or list of words.

Create a morphological analysis of a word or list of words

Calling Method and Arguments

Invoked as an HTTP GET or HTTP POST. Send an HTTP request of the form:

```
GET /bsp/morphologyservice/analysis/word HTTP/1.1
POST /bsp/morphologyservice/analysis/word HTTP/1.1
```

parameter	accepted values	cardinality	notes
word	word to be looked up (may include spaces if compound word)	1	
word_uri	uri for the word	0..1	
lang	iso 639-2 three-letter code	1	
engine	morphology engine code	1	
option	morphology engine option string	0..*	if not supplied will use defaults for engine
encoding	encoding of supplied word	0..1	if not supplied will use default for language
output_schema	OAC annotations, with morphology as body	0 ..1	1.0 release supports the alpheios lexicon schema for annotation body. future releases may support others
Accept Header	application/xml,text/xml,application/json		

- This request is always synchronous (i.e. response returns and is not cached).

Example:

```
POST /bsp/morphologyservice/analysis/word HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 63
word=&lang=grc&engine=morpheus
```

Response

On success, a response with a "201 Created" HTTP status code will be returned:

A response with HTTP 201 (CREATED) status will be returned.

Parameter	Meaning
HTTP Body	one or more OAC Annotations with the morphological analysis results as the annotation body(ies) (analysis output adheres to the schema identified in the output_schema request parameter (for release 1.0 this will be the alpheios lexicon schema).The morphology engine and version used to produce the output will be supplied as the foaf:Agent of the dcterms:creator element of the OAC Annotation.

Example: morph_response_word.xml

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested requested analysis returned no results
500	Internal Server Error	A service error occurred

Parameter	Meaning
HTTP Body	XML document with MorphologyServiceError as root element containing a descriptive message.

Analysis (Document)

Retrieve a requested document from a repository, tokenize it, and produce morphological analyses of the words contained within it.

Create morphological analyses of words in a document.

Calling Method and Arguments

Invoked as an HTTP GET or HTTP POST. Send an HTTP request of the form:

```
GET /bsp/morphologyservice/analysis/document HTTP/1.1
POST /bsp/morphologyservice/analysis/document HTTP/1.1
```

parameter	accepted values	cardinality	notes
document_id	document id (cts urn, cmis object id)	1	
lang	iso 639-2 three-letter code	1	(in future versions, if not supplied,should try to obtain from text)
engine	morphology engine code	1	
repos_type	repository type (e.g. cts, cmis)	0 ..1	
repos_uri	base uri for the repository	0 ..1	
option	morphology engine option string	0..*	if not supplied will use default for engine
output_schema	OAC annotations, with morphology as body	0 ..1	1.0 release supports the alpheios lexicon schema for annotation body. future releases may support others
wait	true or false	0..1	if not supplied, default=true (request will be submitted synchronously)
notify	email address or bamboo person id for notification upon completion	0..1	
Accept Header	application/xml,text/xml,application/json		

Example:

```
POST /bsp/morphologyservice/analysis/document HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 158
lang=grc&engine=morpheus&document_id=http://www.perseus.tufts.edu/hopper/xmlchunk?doc=Perseus:text:1999.01.0025:
card=1553&repos_type=xml
```

Response

On success, a response with a "201 Created" HTTP status code will be returned:

A response with HTTP 201 (CREATED) status will be returned.

Parameter	Meaning
HTTP Body	For synchronous requests (param wait=true) A response with HTTP 201 (CREATED) status will be returned. The response data will contain the OAC Annotations with morphological analyses as annotation body(s), adhering to the schema identified in the output_schema request parameter (for release 1.0 this will be the alpheios lexicon schema).

For asynchronous requests (param wait=false) a response will with HTTP 202 (ACCEPTED) status will be returned. The response data will contain single annotation which represents a summary annotation for the document. The rdf:about value for the Annotation itself will be set to a urn representing a unique identifer for the request. The rdf:about value for the Annotation Body will be the url identifying the location the results will be stored at when complete. |

Example response where wait=0: morph_response_doc_wait.xml
Example of annotation after waiting (truncated to only first two words of document): morph_response_doc.xml

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested document was not found or the analysis returned no results

415	Unsupported Media Type	Returned by the analysis/text request if the mime type of the supplied text is not supported
500	Internal Server Error	A service error occurred

Parameter	Meaning
HTTP Body	XML document with MorphologyServiceError as root element containing a descriptive message. For asynchronous requests that fail, the error message will be stored in cached result document.

Analysis (Text)

Tokenize the supplied text and produce morphological analyses of the words contained within it.

Create morphological analyses of the words in the supplied text.

Calling Method and Arguments

Invoked as an HTTP GET or HTTP POST. Send an HTTP request of the form:

```
GET /bsp/morphologyservice/analysis/text HTTP/1.1
POST /bsp/morphologyservice/analysis/text HTTP/1.1
```

parameter	accepted values	cardinality	notes
text	text to be analyzed	0..1	one of text or text_uri must be supplied
text_uri	a uri from which to retrieve text to be analyzed	0..1	one of text or text_uri must be supplied
mime_type	mime-type of the supplied text (text/html, text/xml, or text/plain)	1	
lang	iso 639-2 three-letter code	1	
engine	morphology engine code	0..*	if not supplied will use default for language
option	morphology engine option string	0..*	if not supplied will use default for engine
wait	true or false	0..1	if not supplied, default=true (request will be submitted synchronously)
notify	email address or bamboo person id for notification upon completion	0..1	
output_schema	OAC annotations, with morphology as body	0..1	1.0 release supports the alpheios lexicon schema for annotation body. future releases may support others
Accept Header	application/xml,text/xml,application/json		

Example:

```
POST /bsp/morphologyservice/analysis/text HTTP/1.1
Accept: application/json, text/javascript, */*
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
Content-Length: 71
lang=lat&engine=morpheus&mime_type=text/plain&text=veni vidi vici
```

Response

On success, a response with a "201 Created" HTTP status code will be returned:

A response with HTTP 201 (CREATED) status will be returned.

Parameter	Meaning
HTTP Body	For synchronous requests (param wait=true) A response with HTTP 201 (CREATED) status will be returned. The response data will contain the OAC Annotations with morphological analyses as annotation body(s), adhering to the schema identified in the output_schema request parameter (for release 1.0 this will be the alpheios lexicon schema). If a text_uri was supplied, that will be the uri of the annotation target (token identification in URI TBD). If a text_uri was not supplied, a uri supplied by the service will be the uri of the annotation target (e.g. to a location in the result set cache at which the supplied text was stored).

For asynchronous requests (param wait=false) a response will with HTTP 202 (ACCEPTED) status will be returned. The response data will contain single annotation which represents a summary annotation for the document. The rdf:about value for the Annotation itself will be set to a urn representing a unique identifer for the request. The rdf:about value for the Annotation Body will be the url identifying the location the results will be stored at when complete. |

Exceptions

If an error occurred, some non-2xx code will be returned. Check the HTTP Status Code that is returned in the response's HTTP headers for the specific error. The following errors may be returned in response to the request:

Error (Status Code)	Meaning	Returned When
400	Bad Request	Request is missing required parameters
404	Not Found	The requested document was not found or the analysis returned no results
415	Unsupported Media Type	Returned by the analysis/text request if the mime type of the supplied text is not supported
500	Internal Server Error	A service error occurred

Parameter	Meaning
HTTP Body	XML document with MorphologyServiceError as root element containing a descriptive message. For asynchronous requests that fail, the error message will be stored in cached result document.

SOA Layer API

See Javadoc.

Notifications

If the notify parameter has been supplied and set to an email address an email will be sent to that address upon completion of the request. If the notify parameter is set to a bamboo person id, notifications will be sent according to user preferences. The body of the notification messages will contain the unique identifier for the request (as returned in the rdf:about value for the annotation returned by the requested), the current status, and the location at which the results can be retrieved.

Tests

See Tests at ${REPOSITORY_ROOT}/platform-services/morphology-service/trunk/morphology-service/resource/src/test/test.html