A REST Proposal for XQuery

Abstract

In this document, we propose a set of HTTP functions in order to invoke REST services in XQuery. All functions are implemented in the Zorba XQuery Processor and, hence, contained in the namespace `http://www.zorba-xquery.org/rest`.

1 Rest Functions

The namespace URI of all REST functions, parameter, and result is http://www.zorba-xquery.org/rest. The prefix is zorba-rest. The following functions should be declared *nondeterministic* as in the upcoming XQuery 1.1 proposal and *sequential* as in the upcoming XQuery Scripting proposal, i.e. `declare nondeterministic sequential function ...`.

Name	Parameters	Result
get	uri as xs:anyURI headers as restHeaders?	restResult
head	uri as xs:anyURI headers as restHeaders?	resultResult
post	uri as xs:anyURI	restResult
post	uri as xs:anyURI payload as restPayload headers as restHeaders?	resultResult
put	uri as xs:anyURI payload as restPayload headers as restHeaders?	resultResult
delete	uri as xs:anyURI headers as restHeaders?	resultResult

2 Parameter Description

URI The first parameter to all functions is the URI of the REST service to invoke. The parameter is of type xs:anyURI and, hence, may be an arbitrary URI including URI parameters. The parameters must be url encoded. Therefore, a processor may offer convenient functions in order to create a valid URI.

Payload HTTP post and put request may send an arbitrary payload to the endpoint. The payload passed to any of these functions can be of arbitrary type (e.g. text/plain, text/xml, or image/jpeg). In order to tell the REST service the type of the payload, HTTP requires the content-type header to be set accordingly in the request headers. The payload to the functions above can be passed as follows:

```
<zorba-rest:payload type="content-type">
</zorba-rest:payload>
```

Header Besides passing information encoded in the URI of the request or send as payload, an HTTP/REST request may invoke the endpoint by passing additional parameters (name-value pairs) as headers of the HTTP request. Such header data is of the form:

```
<zorba-rest:headers>
<header>
<name></name>
<value></value>
</header>
</zorba-rest:headers>
```

If any of the name of any of these pairs is `content-type` this header is overridden by the content-type parameter passed with the payload (if any is passed to a post or put request). If the request is a post or put request and the headers do not contain a content-length header, the content-length is added automatically by buffering the input and counting it's size.

3 Result Description

The result of every (successful) request is of the following form:

```
<zorba-rest:result>
<status-code>2XX</status-code>
<headers>
  <header><name/><value/></header>
</headers>
<payload type="content-type">
</payload>
<zorba:result>
```

The status-code is the code that is returned by the HTTP response. For every successful request this status code is of the form 2xx. The `headers` and the `payload` element are of the same form as described in the sections above (see 2 and 2 respectively).

If the content-type of the response is of type

text/xml or text/xhtml , the content is parsed.

text/plain, the content is of type xs:string.

For all other content-types the type of the payload is xs:base64Binary.

Errors For a request that was returned a status-code other then one of the 2xx status codes, an error is raised.

The URI of any of these errors is http://www.zorba-xquery.org/rest/error. The error object consists of the content that is returned by the request (if it could be send) and is a `zorba-rest:result` element of the form described above (see 3). If the request could not be send, the error object is empty. The localname is one of the following error codes:

- InvalidURL
- InvalidHeaderData
- InvalidPayloadData
- A status-code other then 200