Memento Validator: A toolset for Memento compliance testing

Bhanuka Mahanama*
bhanuka@cs.odu.edu
Old Dominion University
Norfolk, VA, USA

Lyudmila Balakireva
Los Alamos National Laboratory
Los Alamos, NM, USA
ludab@lanl.gov

Sampath Jayarathna
Old Dominion University
Norfolk, VA, USA
sampath@cs.odu.edu

Michael Nelson
Old Dominion University
Norfolk, VA, USA
mln@cs.odu.edu

Martin Klein
Los Alamos National Laboratory
Los Alamos, NM, USA
mklein@lanl.gov

ABSTRACT

Web archiving is serving the task of knowledge preservation for the ever changing state of the web. The Memento protocol provides a unified approach to access versions of web resources across heterogeneous archives and repositories. The discovery of archived content relies on data providers' correct implementation of the Memento protocol, which extends the HTTP protocol with content negotiation over the dimension of time. Implementation inconsistencies can impede the overall time-based content negotiation process and render the resources not usable. We introduce a novel tool set, an API, and a web interface that allow data providers to test their Memento protocol compliance and improve the time-travel experience for users. We offer all our tools as open source to help adoption by the web archiving community.

CCS CONCEPTS

• **Information systems → Digital libraries and archives**; • **Networks → Protocol testing and verification**; • **Software and its engineering → Software testing and debugging**.

KEYWORDS

Web archiving, Memento, Compliance Testing, Memento Infrastructure

ACM Reference Format:
Bhanuka Mahanama, Lyudmila Balakireva, Sampath Jayarathna, Michael Nelson, and Martin Klein. 2022. Memento Validator: A toolset for Memento compliance testing. In *The ACM/IEEE Joint Conference on Digital Libraries in 2022 (JCDL '22), June 20–24, 2022, Cologne, Germany*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3529372.3533297

1 INTRODUCTION

The increase of web archiving efforts by libraries, archives, and other institutional organizations leads to a variety of new implementations of past web repositories. Therefore, supporting a common access protocol such as Memento is of vital importance. The validation of Memento compliance should be the task of data providers and not end-users.

We refer to an archived snapshot of a web resource as a memento. A memento encapsulates a state of a resource at a particular point in time. The Memento protocol, on the other hand, serves as a framework for time-based content negotiation over the Hyper Text Transfer Protocol (HTTP) [3]. The protocol therefore offers a uniform and machine-friendly way to access versions of web resources (mementos) across a heterogeneous landscape of web archives, content management systems, and repositories. The protocol specifies four different types of resources as [2],

1. Original (URI-R): A resource for which prior state access is required [1]
2. Memento (URI-M): Web resource that encapsulates a prior state [2]
3. TimeGate (URI-G): Resource that redirects to a prior state of an original resource [3]
4. TimeMap (URI-T): A resource comprising of serialization of URI-Ms that correspond to a single URI-R [4]

Furthermore, the protocol defines the interaction patterns for the time-based content negotiation using HTTP headers and redirection. A failure to present the client with appropriate headers or responses can render resources inaccessible, despite their existence. This can affect the outcomes of a user studying or browsing the past web. Therefore, identifying issues in a Memento implementation plays a pivotal role in ensuring compliance with the protocol.

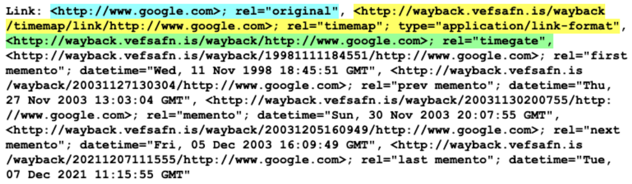
Despite the importance of Memento protocol compliance testing and validation, only two validation approaches have been available. Firstly, we can manually test for compliance by analyzing the responses [1], which is tedious, time-consuming, and undesirable. Alternatively, we can use the Memento Validator [5], the only publicly available validation tool, which analyzes the responses and indicates issues with the implementation. However, since the introduction of the Memento validator, it has offered compliance testing only in the form of a web application. As a result, it lacks desirable properties, such as test automation, multi-level APIs and libraries, and local deployment capabilities.

*This work has been conducted while visiting Los Alamos National Laboratory.

JCDL '22, June 20–24, 2022, Cologne, Germany
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9345-4/22/06.
https://doi.org/10.1145/3529372.3533297

[1] https://datatracker.ietf.org/doc/html/rfc7089#section-2.2.1
[2] https://datatracker.ietf.org/doc/html/rfc7089#section-2.2.4
[3] https://datatracker.ietf.org/doc/html/rfc7089#section-2.2.2
[4] https://datatracker.ietf.org/doc/html/rfc7089#section-2.2.3
[5] http://mementoweb.org/tools/validator/

In this paper, we introduce a novel Memento validator tool set to address these issues. Further, we explain the tool set's capabilities and how each feature can be incorporated into a validation workflow. Finally, we discuss our experience and drawn conclusions from developing the tool set.



```
Link: <http://www.google.com>; rel="original", <http://wayback.vefsafn.is/wayback
/timemap/link/http://www.google.com>; rel="timemap"; type="application/link-format",
<http://wayback.vefsafn.is/wayback/http://www.google.com>; rel="timegate",
<http://wayback.vefsafn.is/wayback/19981111184551/http://www.google.com>; rel="first
memento"; datetime="Wed, 11 Nov 1998 18:45:51 GMT", <http://wayback.vefsafn.is
/wayback/20031127130304/http://www.google.com>; rel="prev memento"; datetime="Thu,
27 Nov 2003 13:03:04 GMT", <http://wayback.vefsafn.is/wayback/20031130200755/http:
//www.google.com>; rel="memento"; datetime="Sun, 30 Nov 2003 20:07:55 GMT",
<http://wayback.vefsafn.is/wayback/20031205160949/http://www.google.com>; rel="next
memento"; datetime="Fri, 05 Dec 2003 16:09:49 GMT", <http://wayback.vefsafn.is
/wayback/20211207111555/http://www.google.com>; rel="last memento"; datetime="Tue,
07 Dec 2021 11:15:55 GMT"
```

Figure 1: HTTP link headers for a URI-M response. Highlighted links refer to other types of resources associated with the resource (URI-R, URI-T, and URI-G).

2 MEMENTO VALIDATOR

During the implementation, we focus on improving the automating capabilities presented in our tool set while addressing a broader range of users. In this section, we discuss how we try to achieve these goals by detailing the tool set architecture with sample usage scenarios.

2.1 Architecture

We implemented the tool set using a layered architecture (see Figure 2) to implement the Memento Validator comprising two layers:

(1) Core library and
(2) Applications.

The core library includes the logical processes for validating a given type of memento. We organize the library into tests and pipelines. Tests define the procedure to validate a specific property of a resource (e.g., presence of a link for a TimeGate in the HTTP response headers when requesting a memento) and pipelines that use tests to validate the given resource's compliance (e.g., validating the compliance of a TimeGate response).

We use the core library to develop user applications in the application layer. In the initial phase, we introduce four applications:

(1) Web Validator: A web application for validating Memento compliance (see Figure 3).
(2) HTTP API: An interface that uses HTTP for communication with JSON serialized responses (see Figure 3).
(3) Validator Reporter: A configurable reporting application that can periodically check and report compliance.
(4) Validator Command Line Interface (CLI): An application with minimal functionalities for compliance testing.

2.2 Usage

Users can use our tool set in the core library or application layer. Since we use Python 3 for implementation, the tool set offers portability across platforms. Firstly, they can use the core library directly. Here, users can automate their tests by encompassing their test cases with Memento Validator tests or pipelines. For instance, a user can include the Memento Validator link header validation test in a test case of a unit test. In this manner, users can automate unit, integration, and system tests. Further, the user can also create customized test cases and pipelines by extending the included components.

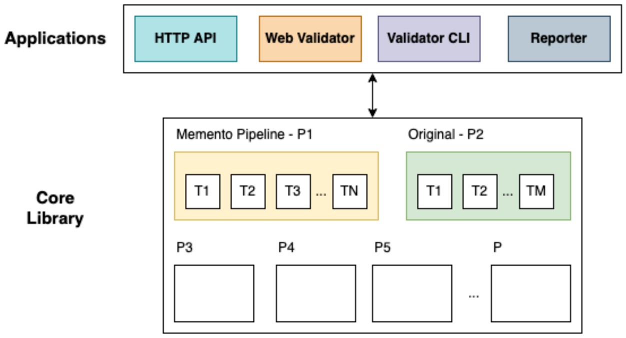


Figure 2: Architecture of the tool set. Note the layered architecture and composition within the core library with pipelines containing multiple tests.

Another approach to use our tool set is to use the included applications. In addition to using our hosted Memento Validator web application and HTTP API [7], users can utilize a self-hosted web application by using the included containerization with a Docker image[8], which can be helpful for compliance testing in private networks. Moreover, users can automate the testing process by using an API automation tool (e.g., SoapUI[9] or Postman[10]) and the HTTP API. The users also can use the command-line application to perform tests over remote connections and automate testing through scripts. Alternatively, users can establish the validator reporter as a periodic job on their machine, which generates a compliance report and sends email notifications.

An important characteristic of our tool set is that our tool set allows automated testing regardless of the underlying technology of the Memento protocol implementation. As a result, our tool set can be used with any existing archive for compliance testing.

3 CONCLUSIONS AND FUTURE WORK

The most important contribution of our work is that we lay the seminal work for an efficient and effective tool set for Memento validation solving an often overlooked problem. With the ever-increasing interest in archival data and the amount of archival data, our tool set facilitates archives to test their compliance conveniently. Moreover, the wide range of tools we offer can cater to users with different levels of technical expertise. As a result, we expect the tool set to alleviate the burden of compliance testing of the providers.

However, we did not gather or study comments from web-archivists at large in this study. As a result, we anticipate modifications and additions to the tool set based on the input from potential users.

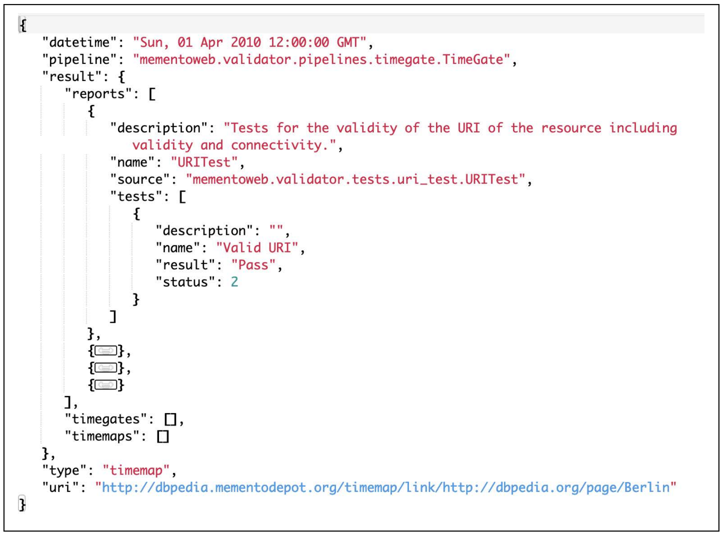
[6] http://dbpedia.mementodepot.org/timemap/link/http://dbpedia.org/page/Berlin

[7] http://labs.mementoweb.org/validator/api

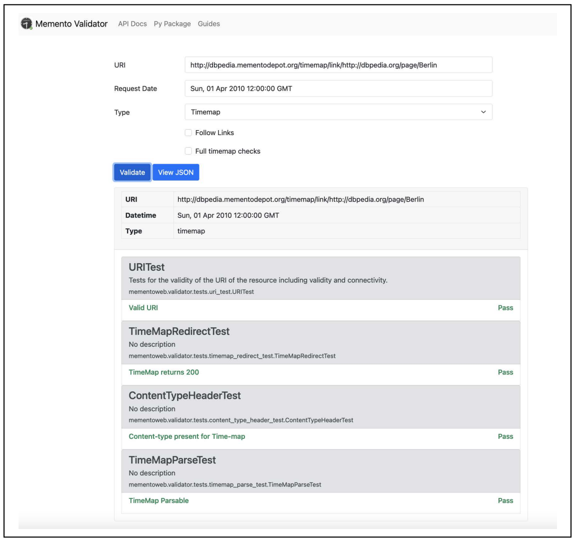
[8] https://www.docker.com/

[9] https://www.soapui.org/

[10] https://www.postman.com/



```
{
  "datetime": "Sun, 01 Apr 2010 12:00:00 GMT",
  "pipeline": "mementoweb.validator.pipelines.timegate.TimeGate",
  "result": {
    "reports": [
      {
        "description": "Tests for the validity of the URI of the resource including
          validity and connectivity.",
        "name": "URITest",
        "source": "mementoweb.validator.tests.uri_test.URITest",
        "tests": [
          {
            "description": "",
            "name": "Valid URI",
            "result": "Pass",
            "status": 2
          }
        ]
      },
      {…},
      {…},
      {…}
    ],
    "timegates": [],
    "timemaps": []
  },
  "type": "timemap",
  "uri": "http://dbpedia.mementodepot.org/timemap/link/http://dbpedia.org/page/Berlin"
}
```



URI http://dbpedia.mementodepot.org/timemap/link/http://dbpedia.org/page/Berlin

Request Date Sun, 01 Apr 2010 12:00:00 GMT

Type Timemap

□ Follow Links

□ Full timemap checks

URI	http://dbpedia.mementodepot.org/timemap/link/http://dbpedia.org/page/Berlin
Datetime	Sun, 01 Apr 2010 12:00:00 GMT
Type	timemap

URITest
Tests for the validity of the URI of the resource including validity and connectivity.
mementoweb.validator.tests.uri_test.URITest
Valid URI Pass

TimeMapRedirectTest
No description
mementoweb.validator.tests.timemap_redirect_test.TimeMapRedirectTest
TimeMap returns 200 Pass

ContentTypeHeaderTest
No description
mementoweb.validator.tests.content_type_header_test.ContentTypeHeaderTest
Content-type present for Time-map Pass

TimeMapParseTest
No description
mementoweb.validator.tests.timemap_parse_test.TimeMapParseTest
TimeMap Parsable Pass

Figure 3: TimeMap tests on a DBpedia article [6]. Left: JSON response from the HTTP API (sections collapsed for brevity), Right: Response from the Web Validator.

We also made the code publicly available[11], enabling access to our tool set to the community at large.

REFERENCES

[1] Sawood Alam. 2020. 2020-03-26: Memento Compliance Audit of PyWB. https://ws-dl.blogspot.com/2020/03/2020-03-26-memento-compliance-audit-of.html

[2] Herber Van de Sompel, Michael Nelson, and Robert Sanderson. 2013. RFC 7089-HTTP framework for time-based access to resource states-Memento. *Internet Engineering Task Force (IETF), RFC* (2013).

[3] Herbert Van de Sompel, Michael L Nelson, Robert Sanderson, Lyudmila L Balakireva, Scott Ainsworth, and Harihar Shankar. 2009. Memento: Time travel for the web. *arXiv preprint arXiv:0911.1112* (2009).

[11] https://github.com/lanl/memento-validator