



Sakai Kernel Bundle

Service Manual – Requests

Date: November 9, 2005
Version: 2

Table of Contents

1	Overview.....	2
1.1	Request Definition.....	2
1.2	Persisting Sessions.....	2
2	Application Programming Interface.....	3
3	Default Sakai Implementation.....	3
4	Utility Objects.....	3
4.1	Request Filter.....	3
4.2	Wrapped Request.....	3
4.3	Wrapped Response.....	3
4.4	Wrapped Request File Upload.....	3
4.5	Request Attributes.....	4
4.6	Configuration Parameters.....	4
4.7	Other Properties.....	5
5	Recommended Practices.....	6
6	References.....	6
7	Document History.....	7

1 Overview

This document describes how requests are handled by Sakai.

Requirements for this service are defined in *Sakai Kernel Bundle Requirements* [1]. This document is based, in part, on an earlier description of tools by Glenn Golden [5].

1.1 Request Definition

A *request* is an HTTP event incoming to a web server to provide information or content. The Sakai Request Filter provides support needed to maintain the tool and user session across request events.

1.2 Persisting Sessions

In general, web transactions are stateless. Sakai needs to maintain a user session across many transactions, so a cookie is used to record information like the current remote user, the current HTTP session, and context [including tool placement?].

2 Application Programming Interface

There is no request manager at this time. A Request Filter provides some conditioning and maintains session information via a cookie.

3 Default Sakai Implementation

See below for notes on the Request Filter.

4 Utility Objects

4.1 Request Filter

The Request Filter is implemented as a utility object in the request module-part of the kernel module. RequestFilter implements javax.servlet.Filter and contains a few inner classes: WrappedRequest, WrappedResponse, WrappedRequestFileUpload, [other?]

The public methods for RequestFilter include:

Filter a Request or Response

```
public void doFilter(ServletRequest requestObj, ServletResponse responseObj, FilterChain chain) throws IOException, ServletException;
```

Filter a request or response.

Initialize the Request Filter

```
public void init(FilterConfig filterConfig) throws ServletException;
```

Initialize the filter by setting the HTTP session type, setting the remote user, setting the tool placement, setting the context id, setting the character encoding, and setting variables associated with file upload including enabled, max size, and temp directory.

4.2 Wrapped Request

This class wraps a request object so that standard behavior can be overridden.

4.3 Wrapped Response

This class wraps a response object so that standard behavior can be overridden.

4.4 Wrapped Request File Upload

This class wraps a file upload request object so that standard behavior can be overridden.

4.5 Request Attributes

The following request attributes are defined by the Request Filter.

Constant	Value	Notes
ATTR_SESSION	sakai.session	The request attribute name used to store the Sakai session.
ATTR_FILTERED	sakai.filtered	The request attribute name (and value) used to indicated that the request has been filtered.
ATTR_UPLOADS_DONE	sakai.uploads.done	The request attribute name (and value) used to indicated that file uploads have been parsed.
ATTR_CHARACTER_ENCODING_DONE	sakai.character.encoding.done	The request attribute name (and value) used to indicated that character encoding has been set.
ATTR_REDIRECT	sakai.redirect	The request attribute name used to indicated that the <u>response</u> has been redirected.
PARAM_AUTO	auto	The request parameter name used to indicate that the request is automatic, not from a user action.

4.6 Configuration Parameters

The following configuration parameters are defined by the Request Filter.

Constant	Value	Notes
CONFIG_SESSION	http.session	parameter to control http session handling.
CONFIG_REMOTE_USER	remote.user	Config parameter to control remote user handling.
CONFIG_TOOL_PLACEMENT	tool.placement	Config parameter to control tool placement URL en/de-coding.
CONFIG_CHARACTER_ENCODING_ENABLED	encoding.enabled	Config parameter to control whether to set the character encoding on the request. Default is true.
CONFIG_CHARACTER_ENCODING	encoding	Config parameter which to control character encoding to apply to the request. Default is UTF-8.
CONFIG_UPLOAD_ENABLED	upload.enabled	Config parameter to control whether the request filter parses file uploads. Default is true. If false, the tool will need to provide its own upload filter that executes BEFORE the Sakai request filter.

CONFIG_UPLOAD_MAX	upload.max	Config parameter to control the maximum allowed upload size (in bytes) from the browser. If defined on the filter, overrides the system property. Default is 1 MB (1048576 bytes). This is an aggregate limit on the sum of all files included in a single request.
CONFIG_UPLOAD_THRESHOLD	upload.threshold	Config parameter (in bytes) to control the threshold at which to store uploaded files on-disk (temporarily) instead of in-memory. Default is 1024 bytes.
CONFIG_UPLOAD_DIR	upload.dir	Config parameter that specifies the absolute path of a temporary directory in which to store file uploads. Default is the servlet container temporary directory. Note that this is TRANSIENT storage, used by the commons-fileupload API. The files must be renamed or otherwise processed (by the tool through the commons-fileupload API) in order for the data to become permanent.
CONFIG_CONTEXT	context	Config parameter to set the servlet context for context based session (overriding the servlet's context name).

4.7 Other Properties

Constant	Value	Notes
SYSTEM_UPLOAD_MAX	sakai.content.upload.max	System property to control the maximum allowed upload size (in MEGABYTES) from the browser. Default is 1 (one megabyte). This is an aggregate limit on the sum of all files included in a single request.
SESSION_COOKIE	JSESSIONID	name of the cookie we use to keep Sakai session.

5 Recommended Practices

[to be written]

6 References

- [1.1] Sakai Kernel Bundle – Overview, Mark J. Norton, Sept. 2005 [URL Here]
- [1.2] Sakai Kernel Bundle Requirements, Mark J. Norton, Sept. 2005 [URL Here]
- [1.3] SKB Manual – Components, v1, Oct. 9, 2005
- [1.4] SKB Manual – Tools, [this document]
- [1.5] SKB Manual – Sessions, [to be written]
- [1.6] SKB Manual – Requests, [to be written]
- [1.7] SKB Manual – Configuration, [to be written]
- [1.8] SKB Manual - Context and Thread Safety, [to be written]
- [1.9] SKB Manual – Logging and Debugging, [to be written]
- [2.1] Sakai’s Component Manager API, Component Packaging, and the Underlying Spring Implementation, Glenn R. Golden, March 23, 2005 [URL here]
- [2.2] Sakai Tools, Glenn R. Golden, July 12, 2005 [URL here]
- [2.3] Sakai Request Processing, Glenn R. Golden, July 12, 2005 [URL here]
- [2.4] How to Configure Sakai, Glenn R. Golden, July 19, 2005 [URL here]
- [2.5] Sakai Sessions, Glenn R. Golden, March 23, 2005 [URL here]

7 Document History

This document was created by Mark Norton. It represents research into the Request Manager (and related) software developed by Glenn Golden of the University of Michigan and includes many of the concepts initially described him in [1].

Date	Version	Who	Work
Oct. 12, 2005	1	Mark Norton	Initial version of the document.
11/9/05	2		Terminology changes; edits.