

Web Experience Management Framework

Version 1.1

REST API Resource Reference



FATWIRE CORPORATION PROVIDES THIS SOFTWARE AND DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. In no event shall FatWire be liable for any direct, indirect, incidental, special, exemplary, or consequential damages of any kind including loss of profits, loss of business, loss of use of data, interruption of business, however caused and on any theory of liability, whether in contract, strict liability or tort (including negligence or otherwise) arising in any way out of the use of this software or the documentation even if FatWire has been advised of the possibility of such damages arising from this publication. FatWire may revise this publication from time to time without notice. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Copyright © 2012 FatWire Corporation. All rights reserved.

The release described in this document may be protected by one or more U.S. patents, foreign patents or pending applications.

FatWire, FatWire Content Server, FatWire Engage, FatWire Satellite Server, CS-Desktop, CS-DocLink, Content Server Explorer, Content Server Direct, Content Server Direct Advantage, FatWire InSite, FatWire Analytics, FatWire TeamUp, FatWire Content Integration Platform, FatWire Community Server and FatWire Gadget Server are trademarks or registered trademarks of FatWire, Inc. in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. AIX, AIX 5L, WebSphere, IBM, DB2, Tivoli and other IBM products referenced herein are trademarks or registered trademarks of IBM Corporation. Microsoft, Windows, Windows Server, Active Directory, Internet Explorer, SQL Server and other Microsoft products referenced herein are trademarks or registered trademarks of Microsoft Corporation. Red Hat, Red Hat Enterprise Linux, and JBoss are registered trademarks of Red Hat, Inc. in the U.S. and other countries. Linux is a registered trademark of Linus Torvalds. SUSE and openSUSE are registered trademarks of Novell, Inc., in the United States and other countries. XenServer and Xen are trademarks or registered trademarks of Citrix in the United States and/or other countries. VMware is a registered trademark of VMware, Inc. in the United States and/or various jurisdictions. Firefox is a registered trademark of the Mozilla Foundation. UNIX is a registered trademark of The Open Group in the United States and other countries. Any other trademarks and product names used herein may be the trademarks of their respective owners.

This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved.

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>).

The OpenSymphony Group license is derived and fully compatible with the Apache Software License; see <http://www.apache.org/LICENSE.txt>.

Copyright (c) 2001-2004 The OpenSymphony Group. All rights reserved.

You may not download or otherwise export or reexport this Program, its Documentation, or any underlying information or technology except in full compliance with all United States and other applicable laws and regulations, including without limitations the United States Export Administration Act, the Trading with the Enemy Act, the International Emergency Economic Powers Act and any regulations thereunder. Any transfer of technical data outside the United States by any means, including the Internet, is an export control requirement under U.S. law. In particular, but without limitation, none of the Program, its Documentation, or underlying information of technology may be downloaded or otherwise exported or reexported (i) into (or to a national or resident, wherever located, of) any other country to which the U.S. prohibits exports of goods or technical data; or (ii) to anyone on the U.S. Treasury Department's Specially Designated Nationals List or the Table of Denial Orders issued by the Department of Commerce. By downloading or using the Program or its Documentation, you are agreeing to the foregoing and you are representing and warranting that you are not located in, under the control of, or a national or resident of any such country or on any such list or table. In addition, if the Program or Documentation is identified as Domestic Only or Not-for-Export (for example, on the box, media, in the installation process, during the download process, or in the Documentation), then except for export to Canada for use in Canada by Canadian citizens, the Program, Documentation, and any underlying information or technology may not be exported outside the United States or to any foreign entity or "foreign person" as defined by U.S. Government regulations, including without limitation, anyone who is not a citizen, national, or lawful permanent resident of the United States. By using this Program and Documentation, you are agreeing to the foregoing and you are representing and warranting that you are not a "foreign person" or under the control of a "foreign person."

FatWire Web Experience Management Framework

Document Revision Date: Jan. 31, 2012

Product Version: 1.1

FatWire Headquarters

FatWire Corporation
330 Old Country Road
Suite 303
Mineola, NY 11501

Table of Contents

About This Guide	7
Who Should Use This Guide	7
Related Documents	7
Conventions	7
Third-Party Libraries	7
REST API	9
Overview	10
REST Resources	10
<i>Asset Type</i>	11
All Asset Types	11
/types	
Selected Asset Type	11
/types/{assettype}	
All Subtypes	11
/types/{assettype}/subtypes	
Selected Subtype	11
/types/{assettype}/subtypes/{subtype}	
Site-Enabled Asset Types	12
/sites/{sitename}/types	
Site-Specific Asset Type Search	12
/sites/{sitename}/types/{assettype}/search	
Global Asset Type Search	12
/types/{assettype}/search	
<i>Asset</i>	14
Selected Asset, Selected Site	14
/sites/{sitename}/types/{assettype}/assets/{id}	

All Asset Associations, Selected Site	14
/sites/{sitename}/types/{assettype}/assets/{id}/associations	
Selected Asset Association and Site	14
/sites/{sitename}/types/{assettype}/assets/{id}/associations/{assocName}	
Site-Specific Assets Search	15
/sites/{sitename}/search	
Global Assets Search	15
/search	
<i>Index</i>	17
Indexing Configurations	17
/indexes	
Indexing Configuration	17
/indexes/{source}	
<i>Site</i>	18
All Sites	18
/sites	
Selected Site.	18
/sites/{sitename}	
<i>Site Plan Navigation</i>	18
Publication Node	18
/sites/{sitename}/navigation	
Page Node	19
/sites/{sitename}/navigation/{pageid}	
<i>Role</i>	20
All Roles	20
/roles	
Selected Role	20
/roles/{rolename}	
<i>Application</i>	20
All Applications.	20
/applications	
Selected Application	20
/applications/{applicationid}	
<i>User</i>	21
All Users	21
/users	
Selected User	21
/users/{username}	
All Site Users.	22
/sites/{sitename}/users	

Selected Site User	22
/sites/{sitename}/users/{username}	
<i>User Locale</i>	23
User Locales	23
/userlocales	
<i>User Definition</i>	23
User Def.	23
/userdef	
<i>ACLs</i>	24
ACLs	24
/acls	
<i>Groups</i>	24
All Groups	24
/groups	
Selected Group.	24
/groups/{groupname}	
<i>Geographical Location.</i>	24
Time Zone	24
/timezone	

About This Guide

This guide provides information about FatWire Content Server's Representational State Transfer (REST) resources and methods.

Who Should Use This Guide

This guide is intended for developers and administrators who are responsible for configuring users' access to REST resources when the users work with applications that are integrated with FatWire Content Server via the FatWire Web Experience Management Framework.

Related Documents

See the following documents in the FatWire documentation set:

- *FatWire Web Experience Management Developer's Guide*
- *FatWire Web Experience Management Administrator's Guide*
- *FatWire Content Server Administrator's Guide*

Conventions

The following text conventions are used in this guide:

- **Boldface** type indicates graphical user interface elements that you select.
- *Italic* type indicates book titles, emphasis, or variables for which you supply particular values.
- `Monospace` type indicates file names, URLs, sample code, or text that appears on the screen.
- `Monospace bold` type indicates a command.

Third-Party Libraries

FatWire Content Server 7.6 patch 2 and its applications include third-party libraries. For additional information, see *FatWire Content Server 7.6 Patch 2: Third-Party Licenses*.

REST API

- [Overview](#)
- [REST Resources](#)

Overview

FatWire Web Experience Management (WEM) Framework provides a Representational State Transfer (REST) API for accessing Content Server's data. REST services are supported on both FatWire Content Server and FatWire Satellite Server to leverage the Satellite Server cache.

The default location of REST resources is under Content Server's BaseURI:

```
<protocol>://<hostname:port>/<servletPath>/REST/
```

For example, on a local installation the BaseURI could be the following:

```
http://localhost:8080/cs/REST/
```

The `<BaseURI>/application.wadl` file lists the resources described in this reference.

REST Resources

This section lists supported REST resources and methods, including error codes that can be returned if a resource does not exist or unforeseen errors occur.

Each resource has two types of views: list and detail. The list view lists resources of a given type (`users` for example) and for each resource, an `href` pointing to the detailed view. For example, the resource `<BaseURI>/users` returns a list view naming all users in the system, among them `fwadmin`.

```
<users>
```

```
  <href>http://localhost:8080/cs/REST/users/fwadmin</href>  
  <name>fwadmin</name>
```

```
</users>
```

The `fwadmin` entry contains an `<href>` that specifies the URL of the detail view where more information about `fwadmin` can be found.

Except for `/timezone` and `/userlocales`, all other REST resources are protected. For information about REST resource security, see the WEM Framework developer and administrator guides.

Note

By default, if an authorization failure occurs, then the login page for Central Authentication Service (CAS) is displayed. If you want to receive a 500 error instead, add `auth-redirect=false` to the URL when making the request.

Asset Type

All Asset Types <code>/types</code>	
Description	This resource reads the list of asset types in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Asset Type <code>/types/{assettype}</code>	
Description	This resource displays detailed information about the specified asset type.
REST Methods	GET, PUT, DELETE
Possible Error Codes	404 if asset type does not exist 500 if unforeseen error occurred

All Subtypes <code>/types/{assettype}/subtypes</code>	
Description	This resource reads all subtypes of the specified asset type.
REST Methods	GET
Possible Error Codes	404 if asset type does not exist 500 if unforeseen error occurred

Selected Subtype <code>/types/{assettype}/subtypes/{subtype}</code>	
Description	This resource reads the specified subtype of the specified asset type.
REST Methods	GET
Possible Error Codes	404 if asset type or subtype does not exist 500 if unforeseen error occurred

Site-Enabled Asset Types`/sites/{sitename}/types`

Description	This resource lists all asset types on the specified site.
REST Methods	GET
Possible Error Codes	404 if site does not exist 500 if unforeseen error occurred

Site-Specific Asset Type Search`/sites/{sitename}/types/{assettype}/search`

Description	<p>This resource supports searches on the specified asset type on the specified site.</p> <p>Note: Access to any search is provided by granting <code>LIST</code> privileges on an <code>Asset</code> resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).</p>
REST Methods	GET
Possible Error Codes	404 if site does not exist

Global Asset Type Search`/types/{assettype}/search`

Description	<p>This resource supports searches on the specified asset type.</p> <p>For this resource to work, the asset type index must be configured in Content Server (otherwise, an exception is thrown and the 404 error code is displayed to the user).</p> <p>Note: Access to any search is provided by granting <code>LIST</code> privileges on an <code>Asset</code> resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).</p>
Query Parameters	<p>q</p> <p>Specify a query String, the search engine-specific query that was specified during the index configuration. (The default search engine is Lucene.)</p>

Global Asset Type Search *(continued)*`/types/{assettype}/search`

	<p>field:[fieldname]:[Operation] Specify a query condition for a field of [fieldname]. [Operation] is optional and can be one of equals, range, startswith, phrase, wildcard, similar, or contains; the default is contains. For the range operation, the value's format is {lower bound}:{upper bound}, e.g., 1:10. Only the first value specified will be used in the search.</p> <p>startindex Specify the starting index from where the results will be returned.</p> <p>count Specify the total number of items to be returned.</p> <p>sortfield:[fieldname]:[asc/des] Specify the field on which the results would be sorted.</p> <p>Note: For Lucene, sorting is possible only fields that were indexed as "not tokenized".</p> <p>Example: <code>http://localhost:8080/cs/REST/types/Content_C/search?field:name:contains=FSII&startindex=5&count=3</code></p>
REST Methods	GET
Possible Error Codes	404 if the indexing configuration does not exist 500 if unforeseen errors occurred

Asset

Selected Asset, Selected Site

`/sites/{sitename}/types/{assettype}/assets/{id}`

Description	This resource provides developers with Create, Edit, Update, Delete (CRUD) functions for operating on the specified asset. {id} is in the long asset id format.
REST Methods	HEAD, GET, POST, PUT, DELETE
Possible Error Codes	404 if site does not exist 404 if asset type does not exist 404 if asset does not exist 500 if unforeseen error occurred

All Asset Associations, Selected Site

`/sites/{sitename}/types/{assettype}/assets/{id}/associations`

Description	This resource lists all asset associations on the specified site.
REST Methods	GET
Possible Error Codes	404 if site does not exist 404 if asset type does not exist 404 if asset does not exist 500 if unforeseen error occurred

Selected Asset Association and Site

`/sites/{sitename}/types/{assettype}/assets/{id}/associations/{assocName}`

Description	This resource lists the specified asset association on the specified site.
REST Methods	GET
Possible Error Codes	404 if site does not exist 404 if asset type does not exist 404 if asset does not exist 500 if unforeseen error occurred

Site-Specific Assets Search

/sites/{sitename}/search

Description	<p>This resource supports searches on all assets on the specified site.</p> <p>Note: Access to any search is provided by granting <code>LIST</code> privileges on an <code>Asset</code> resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).</p>
REST Methods	GET
Possible Error Codes	404 if site does not exist

Global Assets Search

/search

Description	<p>This resource supports global searches on assets.</p> <p>Note: REST does not restrict the types of assets that are returned. As a result, assets of a certain type can be returned to a user who may otherwise not have permission to access the asset type.</p> <p>Note: Access to any search is provided by granting <code>LIST</code> privileges on an <code>Asset</code> resource. In this case, search access can be configured per site or for all sites, but not per asset type. All users with access to the search service will be able to search assets of all asset types (in a particular site, if one is specified).</p>
Query Parameters	<p>q Specify a query String, the search engine-specific query that was specified during the index configuration. (The default search engine is Lucene.)</p> <p>field:[fieldname]:[Operation] Specify a query condition for field of [fieldname] with the specified operation. [Operation] is optional and can be one of equals, range, startswith, phrase, wildcard, similar, or contains; the default is contains. For the range operation, the value's format is {lower bound}:{upper bound}, e.g., 1:10.</p> <p>startindex Specify the starting index from where the results will be returned.</p> <p>count Specify the total number of items to be returned.</p>

Global Assets Search *(continued)*

/search

	sortfield:[fieldname]:[asc/des] Specify the field on which the results would be sorted. Note: For Lucene, sorting is possible only fields that were indexed as “not tokenized”.
	Example: <code>http://localhost:8080/cs/REST/types/Content_C/ search?field:name:contains=FSII&startindex= 5&count=3</code>
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Index

Indexing Configurations

/indexes

Description	This resource lists configurations for global and asset type-based indexing. Global indexing produces a single index of asset types. Asset type indexing produces one index per asset type.
REST Methods	GET, PUT, POST, DELETE
Possible Error Codes	404 if indexing configuration does not exist 500 if unforeseen error occurred

Indexing Configuration

/indexes/{source}

Description	<p>This resource supports the configuration of global indexing and provides CRUD functions for operating on global and per asset type indexing configurations. Global indexing produces a single index of asset types. Asset type indexing produces one index per asset type.</p> <p>Note: In this resource's URL, source is either <code>Global</code> (for the global index) or the name of the asset type (for each asset type-based index). For asset type-based indexing, the indexing process will be also started, stopped, paused, or resumed automatically depending on your choice of HTTP method (<code>PUT</code>, <code>POST</code>, <code>DELETE</code>) and index status (<code>ENABLED</code>, <code>PAUSED</code>). Global indexing is enabled for all asset types by default, although the configuration may not exist.</p>
REST Methods	GET, POST, PUT, DELETE
Possible Error Codes	404 if indexing configuration does not exist 500 if unforeseen error occurred

Site

All Sites	
<code>/sites</code>	
Description	This resource lists all sites in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Site	
<code>/sites/{sitename}</code>	
Description	<p>This resource provides developers with CRUD functions for operating on the specified site.</p> <p>To perform CRUD operations on a site, the user must have either the GeneralAdmin or SiteAdmin role on that site in addition to the privileges granted via groups. To manage AdminSite, a user must have the GeneralAdmin role.</p>
REST Methods	HEAD, GET, POST, PUT, DELETE
Possible Error Codes	404 if site does not exist 500 if unforeseen error occurred

Site Plan Navigation

Publication Node	
<code>/sites/{sitename}/navigation</code>	
Description	<p>This resource reads the Site Plan tree for the site specified by <code>{sitename}</code>. Accessing this resource requires having READ permission on the corresponding site resource. Query parameters defined for the service are as follows:</p> <p><code>depth</code>: Controls the depth to which the Site Plan tree can be read. Default value: 1 Accepted values: Integers or the string <code>all</code> (case-insensitive) to read up to the maximum depth.</p> <p><code>code</code>: Controls the type of site plan nodes (placed or unplaced) to be read. By default, both placed and unplaced pages are read back. The value for this parameter is case-insensitive.</p> <p>For more information about the Site Plan tree, see the <i>Content Server Developer's Guide</i>.</p>

Publication Node *(continued)*

/sites/{sitename}/navigation

REST Methods	GET
Possible Error Codes	<p>400 Exception in parsing depth. Depth is less than or equal to 0.</p> <p>403 User does not have permission.</p> <p>404 Site does not exist.</p> <p>500 No site plan publication node for the site. Any other unforeseen error.</p>

Page Node

/sites/{sitename}/navigation/{pageid}

Description	<p>This resource reads the Site Plan sub-tree for the site specified by {sitename} and page specified by {pageid} as the root. Accessing this resource requires having READ permission to at least the page asset resource that is specified by pageid. Query parameters defined for the service are as follows:</p> <p>depth: Controls the depth to which the Site Plan sub-tree can be read.</p> <p>Default value: 1</p> <p>Accepted values: Integers or the string all (case-insensitive) to read up to the maximum depth.</p> <p>For more information about the Site Plan tree, see the <i>Content Server Developer's Guide</i>.</p>
REST Methods	GET
Possible Error Codes	<p>400 Page asset type is not enabled on site. Exception in parsing depth. Depth is less than or equal to 0.</p> <p>403 User does not have permission.</p> <p>404 Either the site and/or page asset does not exist.</p> <p>500 Node path is null for page. No node elements exist in node path for page. Root node is not a publication node for page. Inconsistent child node. Any other unforeseen error.</p>

Role

All Roles <code>/roles</code>	
Description	This resource reads the list of roles that are stored in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Role <code>/roles/{rolename}</code>	
Description	This resource provides CRUD functions for operating on the specified role.
REST Methods	GET, POST, PUT, DELETE
Possible Error Codes	404 if role does not exist 500 if unforeseen error occurred

Application

All Applications <code>/applications</code>	
Description	This resource lists registered applications.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Application <code>/applications/{applicationid}</code>	
Description	This resource registers the specified application by performing CRUD operations on the application's <code>FW_View</code> and <code>FW_Application</code> assets. More information about application registration is available in the <i>WEM Framework Developer's Guide</i> .
REST Methods	GET, PUT, POST, DELETE
Possible Error Codes	404 if application does not exist 500 if unforeseen error occurred

User

All Users	
<code>/users</code>	
Description	This resource lists all users that are defined in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected User	
<code>/users/{username}</code>	
Description	<p>This resource provides CRUD functions for operating on the specified user. If the user profile contains an image, a 90 x 90 pixel thumbnail image is returned by the service. If the thumbnail image is required in a different size, the request can be made with the following parameters: <code>imageHeight</code> and <code>imageWidth</code> (in pixels); the thumbnail image is then returned at the specified size. The actual uploaded image is always returned in its original size when user information is requested. To restrict the user service to return only the thumbnail image and not the actual image, set the <code>thumbOnly</code> parameter to <code>true</code>.</p> <p>The “otherAttributes” field provides placeholders for custom user attributes, which you define when creating a Content Server user either in LDAP or in Content Server’s database. (Note that for LDAP, each user will be a descendant of the value that is specified for the <code>peopleParent</code> property in <code>dir.ini</code>. The <code>peopleParent</code> property value is set during integration with LDAP.)</p> <p>Regardless of where custom user attributes are created, to take effect in the Content Server application they need to be specified in Content Server’s <code>dir.ini</code> file, in the <code>requiredPeopleAttrs</code> property as a key-value pair in the following format:</p> <pre>requiredPeopleAttrs= attribute1=description1&attribute2=description2</pre> <p>The key is the attribute name as it appears in Content Server’s database (or in LDAP), and the value is the attribute’s description as it appears in the Content Server interface. The property will be looked up by both the Advanced interface and REST API in defining user attributes.</p> <p>For example, if you want to specify two attributes called “Phone number” and “Fax” for all users, do the following:</p> <ol style="list-style-type: none"> 1. Define two attributes called “Phone number” and “Fax”. If you are using Content Server, define them as user attributes (they are stored in the <code>SystemUserAttr</code> table).

Selected User *(continued)*`/users/{username}`

Description	<p>2. In Content Server's <code>dir.ini</code> file, provide the attribute references in the <code>requiredPeopleAttrs</code> property as follows: <code>requiredPeopleAttrs=phone=Phone number&fax=Fax</code></p> <p>The above configuration allows <code>Phone number</code> and <code>Fax</code> to be displayed as attributes in the user account form in Content Server's Advanced interface and the WEM Admin interface.</p> <p>3. If you create a user with <code>Phone number:12345678</code> and <code>Fax:23456789</code>, invoking a REST request on the user URI <code>/users/{username}</code> provides the following information for the "otherAttributes" field:</p> <pre><otherAttributes> <name>phone</name> <value>12345678</value> </otherAttributes> <otherAttributes> <name>fax</name> <value>23456789</value> </otherAttributes></pre>
REST Methods	HEAD, GET, POST, PUT, DELETE
Possible Error Codes	404 if user does not exist 500 if unforeseen error occurred

All Site Users`/sites/{sitename}/users`

Description	This resource lists the roles of all users on the specified site.
REST Methods	GET
Possible Error Codes	404 if site does not exist 500 if unforeseen error occurred

Selected Site User`/sites/{sitename}/users/{username}`

Description	This resource lists the roles of the specified user on the specified site and supports the REST methods listed below.
REST Methods	GET
Possible Error Codes	404 if site does not exist 404 if user does not exist 500 if unforeseen error occurred

User Locale

The `/userlocales` resource is not protected

User Locales <code>/userlocales</code>	
Description	This resource lists all locales that are defined in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

User Definition

User Def <code>/userdef</code>	
Description	This resource reads attributes in the user profile. When the database is used for authentication, only the Content Server predefined user attributes are returned. If LDAP is used, the service returns the attributes that are defined in the <code>requiredPeopleAttrs</code> property of the <code>dir.ini</code> file, along with the predefined Content Server attributes.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

ACLs

ACLs <code>/acls</code>	
Description	This resource lists all ACLs that are defined in Content Server.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Groups

All Groups <code>/groups</code>	
Description	This resource lists all REST security groups in the system.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred

Selected Group <code>/groups/{groupname}</code>	
Description	This resource provides the security privilege configuration of the specified group.
REST Methods	GET
Possible Error Codes	404 if requested group is not found

Geographical Location

The `/timezone` resource is not protected.

Time Zone <code>/timezone</code>	
Description	This resource returns the server's time zone.
REST Methods	GET
Possible Error Codes	500 if unforeseen error occurred