MeritWiki API Specification

Version	Date	Author	Description
0.1	12-Nov-2013	SKircher	Initial draft

Table of Contents

Methods

I. User

1. POST user/register

Create a new MeritWiki user account

Request

Method	URL
POST	/meritapi/user/register/

Request Parameters

Name	Type	Description
user_id	string	User name to uniquely identify this account
password	string	Password to user to secure access to this account
name	string	Name of the user to associated with this account
email	string	Email address to associate with this account

Response Object

Field	Type	Description
success	bool	Registration successful indication
error (optional)	string	Description of error condition that occurred during account registration

Response Format

Field	Type	Description

200	{
	    "success": true
	}	

400	{"success": false, "error": "Password not sufficiently secure."}	

401	{"success": false, "error": "Username already taken."}	
500	{"success": false, error": "Something went wrong. Please try again later."}	

2. POST user/login

Authenticate the user with the system

Request

Method	URL
POST	/meritapi/user/login/

Request Parameters

Name	Type	Description
user_id	string	Unique identifier for the user’s account
password	string	Password to user to secure access to this account

Response Object

Field	Type	Description
auth_key	string	Authentication token to be used in subsequent API method calls.
error (optional)	string	Description of error condition that occurred during login

Response Format

Field	Type	Description

200	{
	"auth_key": <auth_key>
	}	

400	{"error":"Please provide username."}	
400	{"error":"Please provide password."}	

403	{"error”: “Authorization key is missing."}	
401	{"error":"Invalid authorization key."}	
401	{"error":"Incorrect username or password."}	
500	{"error":"Something went wrong. Please try again later."}	

3. GET user/profile

Returns user profile information

Request

Method	URL
GET	/meritapi/user/profile/

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
user_id	string	Unique identifier for the user’s account

Response Object

Field	Type	Description
name	string	Registered name of the user
email	string	Registered email address of the user
status	string	State indicating the operational state of the user’s account

Response Format

Code	Data
200	{ "name": “William Franklin” "email": “wfranklin@hotmail.com” "status": “ACTIVE” }

4. GET user/score

Returns calculated rating score of tabulated ratings from all wiki revision entries made by the user

Request

Method	URL
GET	/meritapi/user/score/

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
user_id	string	Unique identifier for the user’s account

Response Object

Field	Type	Description
rating_score	integer	Calculated totals from all ratings of user’s wiki revision entries

Response Format

Code	Data

200	{
	    "rating_score": 55
	}

5. GET user/tags

Returns collection of tags associated with the user

Request

Method	URL
GET	/meritapi/user/tags/

Request Parameters

Name	Type	Description

auth_key	string	Authentication token that was given in response to user/login
user_id	string	Unique identifier for the user’s account

Response

Field	Type	Description
tags	Collection of Tag	Collection of tag details
tag	Tag	Categorization information applied to users interests, pages, sections, and revisions
tag_name	string	Unique identifier for the tag
tag_desc	string	Description of the tag and its intended usage
tag_status (optional)	string	State indicating the operational state this tag within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

<table>
<tr><th>Code</th><th>Data</th></tr>
<tr><td>200</td><td><pre>{
    "tags": [
        “tag”: [{
             “tag_name”: “Mobile_A/R”
             "tag_desc": “Related to Augmented Reality”
        }]
    ]
}</pre></td></tr>
</table>

II. Tag

1. GET tag/detail

Returns information details about the requested tag

Request

Method	URL
GET	/meritapi/tag/detail

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
tag_name	string	Unique name associated with the tag

Response

Field	Type	Description
tag	Tag	Categorization information applied to users interests, pages, sections, and revisions
tag_name	string	Unique identifier for the tag
tag_desc	string	Description of the tag and its intended usage
tag_status (optional)	string	State indicating the operational state this tag within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	    “tag”: [{
	        “tag_name”: “Mobile_A/R”
	        "tag_desc": “Related to Augmented Reality integration into mobile devices”
	        "tag_status": “ACTIVE”
	    }] }

2. GET tag/tags_by_status

Returns information details about the requested tag

Request

Method	URL
GET	/meritapi/tag/tags_by_status

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
tag_status	string	Status indicator to retrieve list of tags

Response

Field	Type	Description
tags	Collection of Tag	Collection of tag details
tag	Tag	Categorization information applied to users interests, pages, sections, and revisions
tag_name	string	Unique identifier for the tag
tag_desc	string	Description of the tag and its intended usage
tag_status (optional)	string	State indicating the operational state this tag within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	    "tags": [
	        “tag”: [{

	“tag_name”: “Mobile_A/R” "tag_desc": “Related to Augmented Reality integration into mobile devices” "tag_status": “ACTIVE” }]] }

3. POST tag/add

Creates a new tag for use within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding tags.

Request

Method	URL
POST	/meritapi/tag/add

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
tag_name	string	Unique identifier for the tag
tag_desc	string	Description of the tag and its intended usage

Response

Field	Type	Description
success	bool	Tag creation successful indication
error (optional)	string	Description of error condition that occurred during tag creation

Response Format

Code	Data

200	{
	"success": true
	}

401	{"success": false, "error": "Tag already exists, name not unique."}

4. POST tag/update_desc

Updates the description for an existing tag within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating system tags.

Request

Method	URL
POST	/meritapi/tag/update_desc

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
tag_name	string	Unique identifier for the tag
tag_desc	string	New description of the tag and its intended usage

Response

Field	Type	Description
success	bool	Tag creation successful indication
error (optional)	string	Description of error condition that occurred during tag creation

Response Format

Code	Data

200	{
	    "success": true
	}

401	{"success": false, "error": "Tag already exists, name not unique."}

5. POST tag/update_status

Updates the status for an existing tag within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating system tag status.

Request

Method	URL
POST	/meritapi/tag/update_status

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
tag_name	string	Unique identifier for the tag
tag_status	string	New proposed status for the tag (i.e. *active*, *deprecated*, *pending)*

Response

Field	Type	Description
success	bool	Tag status successfully updated indication
error (optional)	string	Description of error condition that occurred during updating tag status

Response Format

Code	Data

200	{
	    "success": true
	}

401	{"success": false, "error": "Tag already exists, name not unique."}

III. Page

1. GET page/detail

Returns information details about the requested page

Request

Method	URL
GET	/meritapi/page/detail

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page

Response

Field	Type	Description
page	Page	Top level container for entries into the MeritWiki system
page_name	string	Unique name associated with the page
page_desc	string	Description of the page and its intended usage
page_url	string	URL suffix associated with the page
page_status (optional)	string	State indicating the operational state this page within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	    “page”: [{
	        “page_name”: “VELCRO_AR”
	        "page_desc": “The new, better WebEx”
	        "page_url": “\velcro_ar”
	        "page_status": “ACTIVE”
	    }]
	}

2. GET page/pages_by_status

Returns collection of page information details for requested pages

Request

Method	URL
GET	/meritapi/page/pages_by_status

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_status	string	Status indicator to retrieve collection of pages

Response

Field	Type	Description
pages	Collection of Page	Collection of page details
page	Page	Top level container for entries into the MeritWiki system
page_name	string	Unique name associated with the page
page_desc	string	Description of the page and its intended usage
page_url	string	URL suffix associated with the page
page_status (optional)	string	State indicating the operational state this page within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	    "pages": [
	        "page": [{
	            "page_name": "VELCRO_AR"
	            "page_desc": "The new, better WebEx"
	            "page_url": "\velcro_ar"
	            "page_status": "ACTIVE"
	        }]
	    ]
	}

3. POST page/add

Creates a new page for use within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding pages.

Request

Method	URL
POST	/meritapi/page/add

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
page_desc	string	Description of the page and its intended usage
page_url	string	URL suffix associated with the page

Response

Field	Type	Description
success	bool	Page creation successful indication
error (optional)	string	Description of error condition that occurred during page creation

Response Format

Code	Data

200	{
	    "success": true
	}

401	{"success": false, "error": "Page already exists, name not unique."}

4. POST page/update

Update basic information for an existing page within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating a page.

Request

Method	URL
POST	/meritapi/page/update

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
page_desc	string	Description of the page and its intended usage
page_url	string	URL suffix associated with the page

Response

Field	Type	Description
success	bool	Page update successful indication
error (optional)	string	Description of error condition that occurred during updating page

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Page not updated, page not found."}
403	{"success": false, "error": "Page not updated, rule minimum threshold not met."}

5. POST page/update_status

Updates the status for an existing page within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating page status.

Request

Method	URL
POST	/meritapi/page/update_status

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
page_status	string	State indicating the operational state this page within the system (i.e. *active*, *deprecated*, *pending)*

Response

Field	Type	Description
success	bool	Page status update successful indication
error (optional)	string	Description of error condition that occurred during updating page status

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Status not updated, page not found."}
403	{"success": false, "error": " Status not updated, not a valid status change."}
403	{"success": false, "error": "Status not updated, rule minimum threshol not met."}

6. POST page/tag

Associates a tag to an existing page within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding a tag to a page. Both the page and the tag must already have been created. Pages with a page_status of “DEPRECATED” may not be tagged.

Request

Method	URL
POST	/meritapi/page/tag

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with an existing page
tag_name	string	Unique identifier for an existing tag

Response

Field	Type	Description
success	bool	Associating tag with the page status successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the page

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Tag not associated, page not found."}
400	{"success": false, "error": “Tag not associated, tag not found."}
403	{"success": false, "error": "“Tag not associated, page status does not allow tagging."}
403	{"success": false, "error": "“Tag not associated, rule minimum threshold not met."}

7. POST page/untag

Dissociate a tag with an existing page within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for deleting a tag association with a page.

Request

Method	URL
POST	/meritapi/page/untag

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with an existing page
tag_name	string	Unique identifier for an existing tag

Response

Field	Type	Description
success	bool	Associating tag with the page status successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the page

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": “Tag not dissociated, tag not associated."}
403	{"success": false, "error": "“Tag not associated, rule minimum threshold not met."}

IV. Section

1. GET section/detail

Returns information details about the requested section for a page

Request

Method	URL
GET	/meritapi/section/detail

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page

Response

Field	Type	Description
section	Section	Container for entries for a page in the MeritWiki system
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
section_desc	string	Description of the section and its intended usage
section_type	string	Enumeration values for valid types of sections (see SectionType)
section_order	integer	Sequential ordering value for sections within a page
section_status (optional)	string	State indicating the operational state this section within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	    “section”: [{
	        “page_name”: “VELCRO_AR”
	        “section_name”: “Title”
	        "section_desc": “VELCRO: Visual & Voice Enhanced Layout for Conference Room Overlay”
	        "section_type": “Title”
	        "section_order": 0
	        "section_status": “PENDING”
	    }]
	}

2. GET section/sections_by_page

Returns information details about the requested tag

Request

Method	URL
GET	/meritapi/section/sections_by_page

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Status indicator to retrieve collection of pages
section_status (optional)	string	State indicating the operational state this section within the system (i.e. *active*, *deprecated*, *pending).* If not provided, sections will be returned regardless of status.

Response

Field	Type	Description
sections	Collection of Section	Collection of page details

section	Section	Container for entries for a page in the MeritWiki system
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
section_desc	string	Description of the section and its intended usage
section_type	string	Enumeration values for valid types of sections (see SectionType)
section_order	integer	Sequential ordering value for sections within a page
section_status (optional)	string	State indicating the operational state this section within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	"**sections**": [
	“**section**”: [{
	“**page_name**”: “VELCRO_AR”
	“**section_name**”: “Title”
	"**section_desc**": “VELCRO: Visual & Voice Enhanced Layout for Conference Room Overlay”
	"**section_type**": “Title”
	"**section_order**": 0
	"**section_status**": “PENDING”
	}]
	]
	}

3. POST section/add

Creates a new section for use within the MeritWiki system. A section with the section_type = “TITLE” is added when the page is created with a section_order = 0. No other section with section_type = “TITLE” nor section_order = 0 may be added. If collisions occur on the section_order value with an existing section for the page the section being added will be inserted at the requested order and other sections will be reorder in sequential order. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding a section. There are separate rules for adding sections to pages created by the current user and pages created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/section/add

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
section_desc	string	Description of the section and its intended usage
section_type	string	Enumeration values for valid types of sections (see SectionType)
section_order	integer	Sequential ordering value for sections within a page

Response

Field	Type	Description
success	bool	Adding new section successful indication
error (optional)	string	Description of error condition that occurred during adding a new section

Response Format

Code	Data

200	{
	    "success": true
	}

401	{"success": false, "error": "Section already exists, name not unique."}
403	{"success": false, "error": " Section not added, name not unique."}
403	{"success": false, "error": "Section not added, rule minimum

	threshold not met."}

4. POST section/update

Updates an existing section for use within the MeritWiki system. A section with the section_type = “TITLE” is added when the page is created with a section_order = 0. The section_type may not be updated after creation. A section may not be updated for the section_type = “TITLE”. A section may not be updated with a section_order = 0. If collisions occur on the section_order value with an existing section for the page the section being added will be inserted at the requested order and other sections will be reorder in sequential order. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating a section. There are separate rules for adding sections to pages created by the current user and pages created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/section/update

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
section_desc (optional)	string	Description of the section and its intended usage.
section_order (optional)	integer	Sequential ordering value for sections within a page

Response

Field	Type	Description
success	bool	Section update successful indication
error (optional)	string	Description of error condition that occurred during updating section

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Section not updated, section not found."}
403	{"success": false, "error": "Section not updated, rule minimum threshold not met."}

5. POST section/update_status

Updates the status for an existing section within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating status of a section. There are separate rules for updating status for section created by the current user and sections created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/section/update_status

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
section_status	string	State indicating the operational state this section within the system (i.e. *active*, *deprecated*, *pending)*

Response

Field	Type	Description
success	bool	Section status update successful indication

error (optional)	string	Description of error condition that occurred during updating section status

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Status not updated, section not found."}
403	{"success": false, "error": " Status not updated, not a valid status change."}
403	{"success": false, "error": "Status not updated, rule minimum threshold not met."}

6. POST section/tag

Associates a tag to an existing section within the MeritWiki system. Both the section and the tag must already have been created. Sections with a section_status of “DEPRECATED” may not be tagged. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding a tag to a section. There are separate rules for tagging sections created by the current user and sections created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/section/tag

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique name associated with an existing page
section_name	string	Name associated with the section, unique within a page
tag_name	string	Unique identifier for an existing tag

Response

Field	Type	Description
success	bool	Associating tag with the page status successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the page

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Tag not associated, section not found."}
400	{"success": false, "error": “Tag not associated, tag not found."}
403	{"success": false, "error": "“Tag not associated, section status does not allow tagging."}
403	{"success": false, "error": "“Tag not associated, rule minimum threshold not met."}

7. POST section/untag

Dissociate a tag with an existing section within the MeritWiki system. Both the section and the tag must already have been created. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for deleting a tag association with a section. There are separate rules for tagging for section created by the current user and sections created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/section/untag

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login

page_name	string	Unique name associated with an existing page
section_name	string	Name associated with the section, unique within a page
tag_name	string	Unique identifier for an existing tag

Response

Field	Type	Description
success	bool	Associating tag with the section status successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the section

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": “Tag not dissociated, tag not associated."}
403	{"success": false, "error": "“Tag not associated, rule minimum threshold not met."}

V. Revision

1. GET revision/detail

Returns information details about the requested revision of a section for a page

Request

Method	URL
GET	/meritapi/section/detail

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
revision_id	string	Unique id associated with a revision assigned by system when a revision is added

Response

Field	Type	Description
revision	Revision	Container for entries for a page in the MeritWiki system
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
user_id	string	Unique identifier for the user's account
body	string	Text contents of the detailed information for this section of the page
timestamp	timestamp	Date and time stamp of when this revision was modified
revision_status (optional)	string	State indicating the operational state this revision within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	“revision”: [{
	“revision_id”: “1234567891011”
	“page_name”: “VELCRO_AR”
	“section_name”: “Title”
	"user_id": “VELCRO: Visual & Voice Enhanced Layout for Conference Room Overlay”
	"body": “Title”
	"timestamp": 0
	"revision_status": “PENDING”
	}]
	}

2. GET revision/latest_revisions

Returns collection of information details for the latest revisions for a requested for a page. If section_name is provided only the latest revision for that section is returned.

Request

Method	URL
GET	/meritapi/section/latest_revisions

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique identifier for page to retrieve collection of revisions
section_name (optional)	string	Name associated with the section, unique within a page. If not provided only the latest revision for that section is returned.
section_status (optional)	string	State indicating the operational state this revision within the system (i.e. *active*, *deprecated*, *pending).* If not provided, revisions will be returned regardless of status.

Response

Field	Type	Description
revisions	Collection of Revision	Collection of revision details
revision	Revision	Container for revision details of entries for a page in the MeritWiki system
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
user_id	string	Unique identifier for the user’s account
body	string	Text contents of the detailed information for this section of the page
timestamp	timestamp	Date and time stamp of when this revision was modified
revision_status (optional)	string	State indicating the operational state this revision within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	"revisions": [
	“revision”: [{
	“revision_id”: “1234567891011”
	“page_name”: “VELCRO_AR”
	“section_name”: “Title”
	"user_id": “VELCRO: Visual & Voice Enhanced Layout for Conference Room Overlay”
	"body": “Title”
	"timestamp": 0
	"revision_status": “PENDING”
	}]
	]
	}

3. GET revision/highest_rated_revisions

Returns collection of information details for the highest rated revisions for a requested for a page. If section_name is provided only the highest rated revision for that section is returned.

Request

Method	URL
GET	/meritapi/section/highest_rated_revisions

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique identifier for page to retrieve collection of revisions
section_name (optional)	string	Name associated with the section, unique within a page. If not provided only the highest rated revision for that section is returned.
section_status (optional)	string	State indicating the operational state this revision within the system (i.e. *active*, *deprecated*, *pending).* If not provided, revisions will be returned regardless of status.

Response

Field	Type	Description
revisions	Collection of Revision	Collection of revision details
revision	Revision	Container for revision details of entries for a page in the MeritWiki system
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
page_name	string	Unique name associated with the page
section_name	string	Name associated with the section, unique within a page
user_id	string	Unique identifier for the user's account

body	string	Text contents of the detailed information for this section of the page
timestamp	timestamp	Date and time stamp of when this revision was modified
revision_status (optional)	string	State indicating the operational state this revision within the system (i.e. *active*, *deprecated*, *pending)*

Response Format

Code	Data

200	{
	    "revisions": [
	        “revision”: [{
	            “revision_id”: “1234567891011”
	            “page_name”: “VELCRO_AR”
	            “section_name”: “Title”
	            "user_id": “VELCRO: Visual & Voice Enhanced Layout for Conference Room Overlay”
	            "body": “Title”
	            "timestamp": 0
	            "revision_status": “PENDING”
	        }]
	    ]
	}

4. POST revision/add

Creates a new revision for use within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding a revision. There are separate rules for adding revisions for pages created by the current user and pages created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/revision/add

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
page_name	string	Unique identifier for page to retrieve collection of revisions
section_name	string	Name associated with the section, unique within a page. If not provided only the highest rated revision for that section is returned.
body	string	Text contents of the detailed information for this section of the page

Response

Field	Type	Description
success	bool	Revision creation successful indication
revision_id {optional)	string	Unique id associated with a revision assigned by system when a revision is added
error (optional)	string	Description of error condition that occurred during adding a new revision

Response Format

Code	Data

200	{
	    "success": true
	     “revision_id”: “1234567891011”
	}

400	{"success": false, "error": "Revision not added, section not found."}
403	{"success": false, "error": " Revision not added, rule minimum threshold not met."}

5. POST revision/update

Updates an existing revision for use within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating a revision. There are separate rules for updating revision to pages created by the current user and pages created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/revision/update

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
body	string	Text contents of the detailed information for this section of the page

Response

Field	Type	Description
success	bool	Revision update successful indication
error (optional)	string	Description of error condition that occurred during updating a revision

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Revision not updated, revision not found."}
403	{"success": false, "error": "Section not updated, rule minimum threshold not met."}

6. POST revision/update_status

Updates the status for an existing revision within the MeritWiki system. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for updating status of a revision. There are separate rules for updating status for revision created by the current user and revisions created by other users with distinct minimum_threshold values.

Request

Method	URL
`POST`	`/meritapi/revision/update_status`

Request Parameters

Name	Type	Description
`auth_key`	`string`	Authentication token that was given in response to user/login
`revision_id`	`string`	Unique id associated with a revision assigned by system when a revision is added
`revision_status`	`string`	State indicating the operational state this revision within the system (i.e. *active*, *deprecated*, *pending)*

Response

Field	Type	Description
`success`	`bool`	Revision status update successful indication

`error`
`(optional)`
	`string`	Description of error condition that occurred during updating revision status

Response Format

Code	Data
`200`	`{ "success": true }`
`400`	`{"success": false, "error": "Status not updated, revision not found."}`
`403`	`{"success": false, "error": " Status not updated, not a valid status change."}`
`403`	`{"success": false, "error": "Status not updated, rule minimum threshold not met."}`

7. POST section/tag

Associates a tag to an existing revision within the MeritWiki system. Both the revision and the tag must already have been created. Revisions with a revision_status of “DEPRECATED” may not be tagged. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for adding a tag to a revision. There are separate rules for tagging revisions created by the current user and revisions created by other users with distinct minimum_threshold values.

Request

Method	URL
`POST`	`/meritapi/section/tag`

Request Parameters

Name	Type	Description
`auth_key`	`string`	Authentication token that was given in response to user/login
`revision_id`	`string`	Unique id associated with a revision assigned by system when a revision is added
`tag_name`	`string`	Unique identifier for an existing tag

Response

Field	Type	Description
`success`	`bool`	Associating tag with the revision status successful indication

`error`
`(optional)`
	`string`	Description of error condition that occurred during associating tag with the revision

Response Format

Code	Data

`200`	`{`
	`    "success": true`
	`}`

`400`	`{"success": false, "error": "Tag not associated, revision not found."}`

400	{"success": false, "error": “Tag not associated, tag not found."}
403	{"success": false, "error": "“Tag not associated, revision status does not allow tagging."}
403	{"success": false, "error": "“Tag not associated, rule minimum threshold not met."}

8. POST revision/untag

Dissociate a tag with an existing revision within the MeritWiki system. Both the revision and the tag must already have been created. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for deleting a tag association with a revision. There are separate rules for untagging for revisions created by the current user and revisions created by other users with distinct minimum_threshold values.

Request

Method	URL
POST	/meritapi/revision/untag

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
tag_name	string	Unique identifier for an existing tag

Response

Field	Type	Description
success	bool	Associating tag with the revision successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the revision

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": “Tag not dissociated, tag not associated."}
403	{"success": false, "error": "“Tag not associated, rule minimum threshold not met."}

VI. Rating

1. POST rating/like

Apply a positive rating for a revision within the MeritWiki system. Revisions with a revision_status of “DEPRECATED” may not be rated. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for liking a revision.

Request

Method	URL
`POST`	`/meritapi/rating/like`

Request Parameters

Name	Type	Description
`auth_key`	`string`	Authentication token that was given in response to user/login
`revision_id`	`string`	Unique id associated with a revision assigned by system when a revision is added
`user_id`	`string`	Unique identifier for the user’s account

Response

Field	Type	Description
`success`	`bool`	Associating tag with the revision status successful indication
`error` `(optional)`	`string`	Description of error condition that occurred during associating tag with the revision

Response Format

Code	Data

`200`	`{`
	`    "success": true`
	`}`

`400`	`{"success": false, "error": "Revision not liked, revision not found."}`
`403`	`{"success": false, "error": "“Revision not liked, revision status`

	does not allow rating."}
403	{"success": false, "error": "“ Revision not liked, rule minimum threshold not met."}

2. POST rating/dislike

Apply a negative rating for a revision within the MeritWiki system. Revisions with a revision_status of “DEPRECATED” may not be rated. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for disliking a revision.

Request

Method	URL
POST	/meritapi/rating/dislike

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
user_id	string	Unique identifier for the user’s account

Response

Field	Type	Description
success	bool	Associating tag with the revision status successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the revision

Response Format

Code	Data

200	{
	    "success": true
	}

400	{"success": false, "error": "Revision not disliked, revision not

	found."}
403	{"success": false, "error": "“Revision not disliked, revision status does not allow rating."}
403	{"success": false, "error": "“ Revision not disliked, rule minimum threshold not met."}

3. POST rating/report

Report a revision within the MeritWiki system as suspect. Revisions with a revision_status of “DEPRECATED” may not be rated. The authenticated user must have a rating_score that meets or exceeds the minimum_threshold for the associated rule for reporting a revision.

Request

Method	URL
POST	/meritapi/rating/dislike

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
revision_id	string	Unique id associated with a revision assigned by system when a revision is added
user_id	string	Unique identifier for the user’s account

Response

Field	Type	Description
success	bool	Associating tag with the revision status successful indication
error (optional)	string	Description of error condition that occurred during associating tag with the revision

Response Format

Code	Data

200	{
	    "success": true

	}
400	{"success": false, "error": "Revision not reported, revision not found."}
403	{"success": false, "error": "“Revision not reported, revision status does not allow rating."}
403	{"success": false, "error": "“ Revision not reported, rule minimum threshold not met."}

VII. Rule

1. GET rule/detail

Returns detailed rule information. If the rule_name is provided only the details associated with the specified rule are returned.

Request

Method	URL
GET	/meritapi/rule/detail

Request Parameters

Name	Type	Description
auth_key	string	Authentication token that was given in response to user/login
rule_name (optional)	string	Unique identifier for a rule.

Response

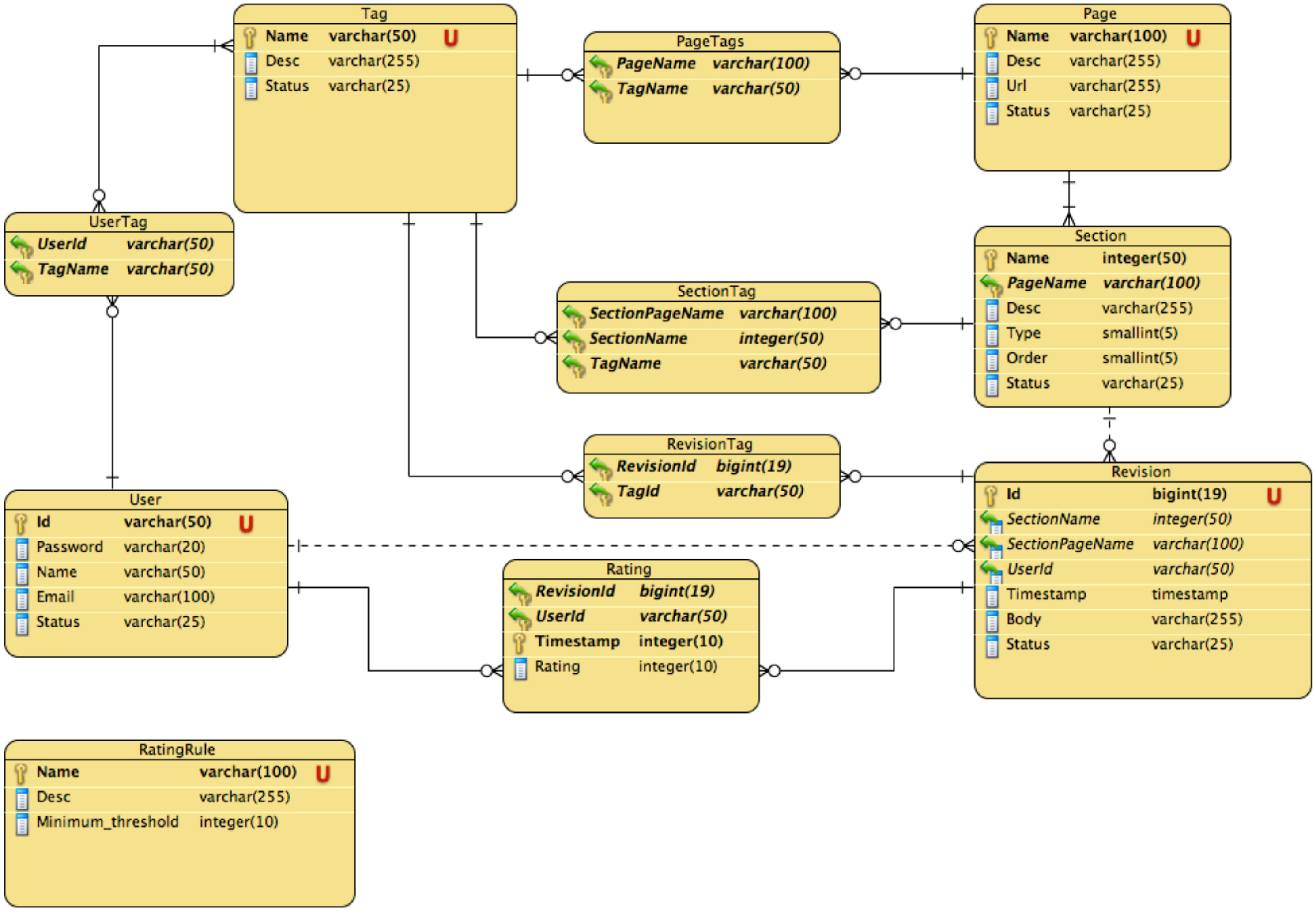
Field	Type	Description
rule_name	string	Unique identifier for a rule
rule_desc	string	Description of the tag and its intended usage
minimum_threshold	integer	Value of a users overall score required to satisfy permission for this rule

Response Format

Code	Data

200	{
	"rules": [
	“rule”: [{
	“rule_name”: “ADD_OWN_SECTION”
	"rule_desc": “Ability for a user to add a section to a page they
	created”
	"minimum_threshold": 5
	}]
	] }

Name	MeritWiki API ERD
Author	Shawn Kircher


Tag
Name varchar(50) U
Desc varchar(255)
Status varchar(25)
PageTags
PageName varchar(100)
TagName varchar(50)
Page
Name varchar(100) U
Desc varchar(255)
Url varchar(255)
Status varchar(25)
UserTag
UserId varchar(50)
TagName varchar(50)
Section
Name integer(50)
PageName varchar(100)
Desc varchar(255)
Type smallint(5)
Order smallint(5)
Status varchar(25)
SectionTag
SectionPageName varchar(100)
SectionName integer(50)
TagName varchar(50)
RevisionTag
RevisionId bigint(19)
TagId varchar(50)
Revision
Id bigint(19) U
SectionName integer(50)
SectionPageName varchar(100)
UserId varchar(50)
Timestamp timestamp
Body varchar(255)
Status varchar(25)
User
Id varchar(50) U
Password varchar(20)
Name varchar(50)
Email varchar(100)
Status varchar(25)
Rating
RevisionId bigint(19)
UserId varchar(50)
Timestamp integer(10)
Rating integer(10)
RatingRule
Name varchar(100) U
Desc varchar(255)
Minimum_threshold integer(10)