

What is the WebAPI

The WebAPI is an interface used to access to handbook from external.
This WebAPI is available from Handbook Studio 3.0.16.

--Remark: when you post data, the data encoding should be utf-8.

connect

connect to handbook

Request

Method	Request
GET	http://<SERVER>/eapi/connect

Parameters

Parameters	Require	Description
username	Yes	only groupAdmin; begin with a letter; only include:letter,number,"-" and "_"; size:4-30
password	Yes	size:6-50

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
data	Object	return data information

data

Key	Type	Description
-----	------	-------------

id	Int	editor id
username	String	
first_name	String	
last_name	String	
email	String	
organization	String	organization name

If the response if failed, see common error response.

Example Request

`http://<SERVER>/eapi/connect?username=username&password=password`

Example Response

```
{
  "success": true,
  "data": {
    "id": "id",
    "username": "username",
    "first_name": "first name",
    "last_name": "last name",
    "email": "email",
    "organization": "organization"
  }
}
```

disconnect

disconnect from handbook

Request

Method	Request
GET	<code>http://<SERVER>/eapi/disconnect</code>

Parameters

This API requires no parameters.

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false

If the response if failed, see common error response.

Example Request

`http://<SERVER>/eapi/disconnect`

Example Response

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

uploadHandbook

Upload handbook

Request

Method	Request
POST	<code>http://<SERVER>/eapi/uploadHandbook</code>

Parameters

Parameters	Require	Description
filename	Yes	the upload file name

Request Head

Content-Type: multipart/form-data; boundary=-----XXXXXXXX
Content-Length: YYYYYY

Request Body

-----XXXXXX

Content-Disposition: form-data; name="filename"; filename="xxxxxx.zip(or
xxxxxx.hbz)"

Content-Type: application/x-zip(or application/octet-stream)

.....
... <the upload file>Binary data ...

.....
-----XXXXXX--

remark:

Only allow zipped handbook,the filename extension is "zip" or "hbz".

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
data	Object	return data information

data

Key	Type	Description
id	Int	handbook id
name	String	handbook name
keycode	String	the handbook keycode
cover	String	when add/edit, it post the attachment id to server side, and it will be convert to path when get next time; rule:32 md5+ext ; for web side, can use this path to get the picture file
icon	String	the same logic as cover
num	Int	the handbook number
username	String	

If the response if failed, see common error response.

Example Request

URL: http://<SERVER>/eapi/uploadHandbook

The request head is as below:

Content-Type: multipart/form-data; boundary=-----XXXXXX
Content-Length: YYYYYY

The request body is as below:

-----XXXXXX
Content-Disposition: form-data; name="filename"; filename="handbook.zip"
Content-Type: application/x-zip

.....
... <the upload file binary data> ...
.....

-----XXXXXX--

Example Response

```
{"success":true,"data":  
{"books":[{"id":"id","name":"handbookName","keycode":"keycode","cover":  
"cover","icon":"icon","num":"number","username":"username"}]}}
```

downloadHandbookByTitle

download handbook by title

Request

Method	Request
GET	http://<SERVER>/eapi/downloadHandbookByTitle

Parameters

Parameters	Require	Description
handbook	Yes	the be downloaded handbook name

Request Body

The API requires no request body.

Responses

zipped handbook file.

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/downloadHandbookByTitle?  
handbook=HandbookName
```

downloadScores

download the user scores information.

Request

Method	Request
GET	http://<SERVER>/eapi/downloadScores

Parameters

Parameters	Require	Description
start	Yes	the time when start whole quiz/exam, GMT, convert to GMT in client before post, and back to locale time after get. Format such as:2011-09-09
end	Yes	the time when finish whole quiz/exam, GMT, convert to GMT in client before post, and convert to locale time after get.

		Format such as:2011-09-09
encoding	No	shift-jis or utf-8; default: utf-8

Request Body

The API requires no request body.

Responses

csv file. content as follows:

Start Time | End Time | User | Handbook Title | Chapter Title | Section Title |
Total Scores | Question Serial number | Questions Title| Scores

If the response if failed, see common error response.

Example Request

<http://<SERVER>/eapi/downloadScores?start=2011-09-09&end=2011-12-31&encoding=utf-8>

downloadResult

download user questionnaire result information.

Request

Method	Request
GET	<a href="http://<SERVER>/eapi/downloadResult">http://<SERVER>/eapi/downloadResult

Parameters

Parameters	Require	Description
start	Yes	the time when start whole quiz/exam, GMT, convert to GMT in client before post, and back to locale time after get. Format such as:2011-09-09
end	Yes	the time when finish whole quiz/exam, GMT, convert to GMT in client before post, and convert to locale time after get.

		Format such as:2011-09-09
encoding	No	shift-jis or utf-8; default:utf-8

Request Body

The API requires no request body.

Responses

csv file. content as follows:

Handbook Title | Chapter Title | Section Title | Start Time | End Time | Questions
| User | Answer

If the response if failed, see common error response.

Example Request

`http://<SERVER>/eapi/downloadResult?start=2011-09-09&end=2011-12-31&encoding=utf-8`

downloadUserLog

download user log information.

Request

Method	Request
GET	<code>http://<SERVER>/eapi/downloadUserLog</code>

Parameters

The API requires no parameters.

Request Body

The API requires no request body.

Responses

zipped log file.

If the response if failed, see common error response.

Example Request

http://<SERVER>/eapi/downloadUserLog

createHandbook

create handbook

Request

Method	Request
GET	http://<SERVER>/eapi/createHandbook

Parameters

Parameters	Require	Description
handbook	Yes	the new handbook name
keycode	Yes	handbook keycode

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	handbook id
data	Object	return data information
disksize	String	the organization used disk size

data

Key	Type	Description
permlink	String	permanent link, a customize url format for client to identify a special section and jump to that page, for example, 'handbook3://<server_name>/H235343454'

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/createHandbook?handbook=handbookName
&keycode=keycode
```

Example Response

```
{"success":true,"id":"id","data":
{"permlink":"handbook3:\V<server_name>\H880586113"},
"disksize":"disksize"}
```

createChapter

create chapter

Request

Method	Request
GET	http://<SERVER>/eapi/createChapter

Parameters

Parameters	Require	Description
handbook	Yes	the handbook name
chapter	Yes	the new chapter name
keycode	No	chapter keycode; default:handbook keycode

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	chapter id
data	Object	return data information
disksize	String	the organization used disk size

data

Key	Type	Description
permlink	String	permanent link, a customize url format for client to identify a special section and jump to that page, for example, 'handbook3://<server_name>/H235343454S235632563'

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/createChapter?  
handbook=handbookName&chapter=chapterName&keycode=keycode
```

Example Response

```
{"success":true,"id":"id","data":  
{"permlink":"handbook3:\V<server_name>\H880586113S346743564"},  
"disksize":"disksize"}
```

createSection

create section

Request

Method	Request
POST/GET	http://<SERVER>/eapi/createSection

Parameters

Parameters	Require	Description
handbook	Yes	the handbook name
section	Yes	the new section name

Request Head

Content-Type: multipart/form-data; boundary=-----XXXXXXXX
Content-Length: YYYYYY

Request Body

-----XXXXXXXX

Content-Disposition: form-data; name="handbook"
Content-Type: text/plain

<the handbook title>

-----XXXXXXXX

Content-Disposition: form-data; name="section"
Content-Type: text/plain

<the section title>

-----XXXXXXXX

Content-Disposition: form-data; name="filename"; filename="<picture.jpg>"
Content-Type: image/jpg

.....
... <the upload file binary data> ...

.....
-----XXXXXXXX--

remark:

the upload file allow types:

jpg,jpeg,png,bmp,gif,mp4,mov,m4v,pdf,key,ppt,pptx,numbers,xls,
xlsx,pages,doc,docx,wav,aac,mp3,m4a,aif,aiff,hbs

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	section id
data	Object	return data information
disksize	String	the organization used disk size

data

Key	Type	Description
permlink	String	permanent link, a customize url format for client to identify a special section and jump to that page, for example, 'handbook3://<server_name>/H235343454S232144246'

If the response if failed, see common error response.

Example Request

GET

http://<SERVER>/eapi/createSection?

handbook=handbookName§ion=sectionName

ps: This example use 'GET' method, not upload file.

POST

URL: http://<SERVER>/eapi/createSection

The request head is as below:

Content-Type: multipart/form-data; boundary=-----XXXXXXXX

Content-Length: YYYYYY

The request body is as below:

-----XXXXXXXX

Content-Disposition: form-data; name="handbook"

Content-Type: text/plain

<the handbook title>

-----XXXXXXXX

Content-Disposition: form-data; name="section"
Content-Type: text/plain

<the section title>

-----XXXXXX

Content-Disposition: form-data; name="filename"; filename="<picture.jpg>"
Content-Type: image/jpg

.....
... <the upload file binary data> ...
.....

-----XXXXXX--

Example Response

```
{"success":true,"id":"id","data":  
{"permlink":"handbook3:V<server_name>VH880586113S233532575"},  
"disksize":"disksize"}
```

updateHandbookByTitle

update handbook by title. P.S. If there are two books with the same name, the selected book will be the first book according to the order of the num asc.

Request

Method	Request
GET	http://<SERVER>/eapi/updateHandbookByTitle

Parameters

Parameters	Require	Description
handbook	Yes	the old handbook title
newhandbook	No	the new handbook title; default:the old handbook title

keycode	No	default:the original handbook keycode
---------	----	---------------------------------------

Request Body

The API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
disksize	String	the organization used disk size

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/updateHandbookByTitle?handbook=handbookName
&newhandbook=newHandbookName&keycode=keycode
```

Example Response

```
{"success":true,"disksize":"disksize"}
```

updateChapterByTitle

update chapter by title. P.S. If there are two chapters with the same name, the selected chapter will be the first chapter according to the order of the num asc.

Request

Method	Request
GET	http://<SERVER>/eapi/updateChapterByTitle

Parameters

Parameters	Require	Description
------------	---------	-------------

handbook	Yes	the handbook title
chapter	Yes	the old chapter title
newchapter	No	the new chapter title; default:the old chapter title
keycode	No	default:the old chapter keycode

Request Body

The API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
disksize	String	the organization used disk size

If the response if failed, see common error response.

Example Request

`http://<SERVER>/eapi/updateChapterByTitle?handbook=handbookName &chapter=chapterName&newchapter=newChapterName&keycode=keycode`

Example Response

`{"success":true,"disksize":"disksize"}`

updateSectionByTitle

update section by title. P.S. If there are two sections with the same name, the selected section will be the first section according to the order of the num asc.

Request

Method	Request
POST/GET	<code>http://<SERVER>/eapi/updateSectionByTitle</code>

Parameters

Parameters	Require	Description
handbook	Yes	the handbook title
section	Yes	the old section title
newsection	No	the new section title; default:the old section title

Request Head

Content-Type: multipart/form-data; boundary=-----XXXXXX
Content-Length: YYYYYY

Request Body

-----XXXXXX

Content-Disposition: form-data; name="handbook"
Content-Type: text/plain

<the handbook title>

-----XXXXXX

Content-Disposition: form-data; name="section"
Content-Type: text/plain

<the old section title>

-----XXXXXX

Content-Disposition: form-data; name="newsection"
Content-Type: text/plain

<the new section title>

-----XXXXXX

Content-Disposition: form-data; name="filename"; filename="<picture.jpg>"
Content-Type: image/jpg

.....
... <the upload file binary data> ...

.....
-----XXXXXX--

remark:

the upload file allow types:

jpg,jpeg,png,bmp,gif,mp4,mov,m4v,pdf,key,ppt,pptx,numbers,xls,
xlsx,pages,doc,docx,wav,aac,mp3,m4a,aif,aiff,hbs

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
permlink	String	permanent link, a customize url format for client to identify a special section and jump to that page, for example, 'handbook3://<server_name>/H235343454S232144246'
disksize	String	the organization used disk size

If the response if failed, see common error response.

Example Request

GET:

http://<SERVER>/eapi/updateSectionByTitle?
handbook=handbookName§ion=sectionName
&newsection=newSectionName
ps: This example use 'GET' method, not upload file.

POST:

URL: http://<SERVER>/eapi/updateSectionByTitle

The request head is as below:

Content-Type: multipart/form-data; boundary=-----XXXXXXXX
Content-Length: YYYYYY

The request body is as below:

-----XXXXXXXX
Content-Disposition: form-data; name="handbook"
Content-Type: text/plain

<the handbook title>
-----XXXXXXXX
Content-Disposition: form-data; name="section"
Content-Type: text/plain

<the old section title>
-----XXXXXXXX
Content-Disposition: form-data; name="newsection"
Content-Type: text/plain

<the new section title>

-----XXXXXX

Content-Disposition: form-data; name="filename"; filename="<picture.jpg>"

Content-Type: image/jpg

.....
... <the upload file>Binary data ...
.....

-----XXXXXX--

Example Response

```
{"success":true,"data":  
{"permlink":"handbook3:V<server_name>VH880586113S233532575"},  
"disksize":"disksize"}
```

removeHandbookByTitle

remove handbook by title. P.S. If there are two books with the same name, the selected book will be the first book according to the order of the num asc.

Request

Method	Request
GET	http://<SERVER>/eapi/removeHandbookByTitle

Parameters

Parameters	Require	Description
handbook	Yes	the handbook title

Request Body

The API requires no request body.

Responses

This API returns json

Key	Type	Description
-----	------	-------------

success	Boolean	true or false
---------	---------	---------------

If the response if failed, see common error response.

Example Request

`http://<SERVER>/eapi/removeHandbookByTitle?handbook=handbookName`

Example Response

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

removeChapterByTitle

remove chapter by title. P.S. If there are two chapters with the same name, the selected chapter will be the first chapter according to the order of the num asc.

Request

Method	Request
GET	<code>http://<SERVER>/eapi/removeChapterByTitle</code>

Parameters

Parameters	Require	Description
handbook	Yes	the handbook title
chapter	Yes	the chapter title

Request Body

The API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/removeChapterByTitle?  
handbook=handbookName&chapter=chapterName
```

Example Response

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

removeSectionByTitle

remove section by title. P.S. If there are two sections with the same name, the selected section will be the first section according to the order of the num asc.

Request

Method	Request
GET	http://<SERVER>/eapi/removeSectionByTitle

Parameters

Parameters	Require	Description
handbook	Yes	the handbook title
section	Yes	the section title

Request Body

The API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/removeSectionByTitle?  
handbook=handbook&section=section
```

Example Response

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

createHandbookUser

create handbook user

Request

Method	Request
GET	http://<SERVER>/eapi/createHandbookUser

Parameters

Parameters	Require	Description
username	Yes	begin with a letter; only include: letter, number, "-" and "_"; size:4-30
firstname	Yes	size:1~30
lastname	Yes	size:1~30
email	Yes	maxSize:100
password	Yes	size:6-50
keycode	No	size:0~50;split by ",";only include: letter or number
deviceid	No	user's device ids, split by ",", " or ";"
oplog	No	whether recode the operation records; int[0/1]; if it is a not '0' string, set to '1'; default:0

division	No	maxSize:50
----------	----	------------

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	user id
data	Object	return data information

data

Key	Type	Description
track	Int[0/1]	whether recode the operation records
keycodes	String	keycodes

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/createHandbookUser?  
username=username&firstname=firstname&lastname=lastname&email=email  
&password=password&keycode=keycode&deviceid=deviceid&oplog=oplog
```

Example Response

```
{"success":true,"id":"id","data":{"track":"track","keycodes":"keycodes"}}
```

createEditorUser

create editor user

Request

Method	Request
GET	http://<SERVER>/eapi/createEditorUser

Parameters

Parameters	Require	Description
username	Yes	begin with a letter; only include: letter, number,"-" and "_"; size:4-30
firstname	Yes	size:1~30
lastname	Yes	size:1~30
email	Yes	maxSize:100
password	Yes	size:6-50

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	editor id

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/createEditorUser?  
username=username&firstname=firstname&lastname=lastname&email=email  
&password=password
```

Example Response

```
{"success":true,"id":"editor id"}
```


removeHandbookUser

remove handbook user

Request

Method	Request
GET	http://<SERVER>/eapi/removeHandbookUser

Parameters

Parameters	Require	Description
username	Yes	begin with a letter; only include: letter, number, "-" and "_"; size:4-30

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false

If the response if failed, see common error response.

Example Request

http://<SERVER>/eapi/removeHandbookUser?username=username

Example Response

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

removeEditorUser

remove editor user

Request

Method	Request
GET	http://<SERVER>/eapi/removeEditorUser

Parameters

Parameters	Require	Description
username	Yes	begin with a letter; only include: letter, number, "-" and "_"; size:4-30

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/removeEditorUser?username=editorName
```

Example Response

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

updateHandbookUser

update handbook user

Request

Method	Request
GET	http://<SERVER>/eapi/updateHandbookUser

Parameters

Parameters	Require	Description
username	Yes	begin with a letter; only include: letter, number, "-" and "_"; size:4-30
firstname	No	size:1~30
lastname	No	size:1~30
email	No	maxSize:100
password	No	size:6-50
keycode	No	size:0~50; split by ","; only include: letter or number
deviceid	No	user's device ids; split by "," or ";"; only include: letter or number
disable	No	int[0/1]; disable=1, means this user is disable; default:0;
oplog	No	whether recode the operation records; int[0/1]; if it is a not '0' string, set to '1'; default:0
division	No	maxSize:50

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	handbook user id

If the response if failed, see common error response.

Example Request

```
http://<SERVER>/eapi/updateHandbookUser?  
username=username&firstname=firstname&lastname=lastname&email=email  
&password=password&keycode=keycode&deviceid=deviceid&oplog=oplog  
&disable=disable
```

Example Response

```
{"success":true,"id":"id"}
```

updateEditorUser

update editor user

Request

Method	Request
GET	http://<SERVER>/eapi/updateEditorUser

Parameters

Parameters	Require	Description
username	Yes	begin with a letter; only include:letter,number,"-" and "_"; size:4-30

firstname	No	size:1~30
lastname	No	size:1~30
email	No	maxSize:100
password	No	size:6-50
disable	No	int[0/1]; disable=1, means this user is disable; default:0;
admin	No	only: editor or groupAdmin; default:editor

Request Body

This API requires no request body.

Responses

This API returns json

Key	Type	Description
success	Boolean	true or false
id	Int	editor id

If the response if failed, see common error response.

Example Request

http://<SERVER>/eapi/updateEditorUser?
 username=username&firstname=firstname&lastname=lastname&email=email
 &password=password&disable=disable&admin=groupAdmin

Example Response

```
{"success":true,"id":"id"}
```

Common Error Response

the common error response.

Key	Type	Description
success	Boolean	false
code	Int	error code
error	String	error information

Example Response

```
{"success":false,"code":"code","error":"error"}
```

the error code and error information table

error code	error information
100	PERMISSION_DENIED
101	NO_RECORD
102	ACCOUNT_DISABLE
103	ACCOUNT_NOT_ACTIVATED
104	PASSWORD_NOT_CORRECT
105	PARAMETERS_ERROR
106	NAME_TOO_LONG
107	CATEGORY_ERROR
108	TAG_NULL_OR_TOO_LONG
109	ATTACHMENT_NAME_TOO_LOG
110	LENGTH_TOO_LOGN
111	TAG_TOO_LONG
112	ATTACHMENT_TYPE_ERROR
113	ATTACHMENT_TIME_IS_NUMBER
114	UPLOAD_FILE_TOO_LARGE
115	PART_WAS_UPLOADED

116	NO_FILE_WAS_UPLOADED
117	TEMPORARY_FILE_NOT_FOND
118	FILE_WRITE_FAILURE
119	PHP_EXTENSION_ERROR
120	CONVERT_THUMBNAIL_FAIL
121	ADD_WATERMARK_FAIL
122	DELETE_FILE_FAILED
123	VIDEO_CAPTURE_FAILED
124	Add_WATERMARK_TO_VIDEO_CAPTURE_FAILED
125	FAILURE_TO_SET_THE_WATERMARK_IAMGE
126	THIS_IMAGE_HAS_A_WATERMARK
127	FILE_USER_NOT_DELETE
128	DESCRIPTION_TOO_LONG
129	LANGUAGE_TYPE_ERRORS
130	KEYCODE_TOO_LONG
131	COPYRIGHT_TOO_LONG
132	COPYRIGHT_LAYOUT_ERRORS
133	CATETORY_TOO_LOGN
134	TIPS_NULL_OR_LENTH_MORE_THAN
135	FAILURE_OF_THE_EXPORT_FILE
136	FOLDER_NAME_ALREADY_EXISTS
137	ID_EMPTY_OR_NOT_NUM
138	INCORRECT_EMAIL
139	USERNAME_EXIST

140	USERNAME_NO_EXIST
141	URL_TOO_LONG
142	KEYCODE_ALREADY_EXISTS
143	FILE_NOT_EXIST
144	FILE_OPERATION_FAIL
145	CAN_NOT_DELETE
146	ATTACHMENT_IS_BEING_USED
147	CSV_FILE_FIELD_COUNT_INVALID
148	CSV_FILE_FIELD_NAME_INVALID
149	QUESTION_MODE_ERROR
150	QUESTION_TYPE_ERROR
151	ACCOUNT_NOT_EXIST
152	EMAIL_EXIST
153	ORGANIZATION_EXIST
154	NO_LOG_RECORDS_ARE_FOUND
155	INVALID_USERNAME_OR_PASSWORD
156	SECTION_LAYOUT_ERRORS
157	FILE_CANNOT_CONVERT_TO_HANDBOOK
158	ROOT_DIR_CANNOT_CONVERT_TO_HANDBOOK
159	NULL_BLANK_TAG
160	HAS_THE_SAME_USERNAME
200	SAVE_FAIL
201	SELECT_FAIL
202	DELETE_FAIL

203	Abnormal_Operation
204	MOVE_FAIL
205	CLONE_FAIL
310	UNZIP_FAIL
311	EDITOR_DELETE_CHECK
312	ZIP_DECOMPRESSION_FAIL
314	XML_PARSE_FAIL
315	XML_STRUCTURE_INCORRECT
316	MISS_REQUIRED_ATTACHMENT
317	XML_NOT_FOUND
318	ID_NOT_FOUND
319	FILE_TYPE_DOES_NOT_MATCH
320	MISS_REQUIRED_FIELD
400	LOGIN_FALSE