



Image Annotation

Workflow API

V1.1

Content

Workflow API	1
1. User Login	3
2. User Logout	4
3. Get Current User Information	4
4. Change Password	5
5. Get Project List	6
6. Get Project	7
7. Get Tasks of Project	8
8. Get Task List	9
9. Create Task	10
10. Import Tasks	12
11. Get Task	13
12. Update Task	14
13. Remove Task	15
14. Get Task Annotation	16
15. Update Task Annotation	20
16. Delete Task Annotation	21
18. Get Task Data	22



19.	Get Task Meta Data	23
20.	Export Task Annotation	25
21.	Get Server Information	25
22.	Get Supported Annotation Formats	26



This API is based on the HTTPS protocol with token exchange. The client needs to call the login method to get the token returned from the service. All subsequent calls will use the token as the authentication parameter.

1. User Login

- **API Endpoint: /interface/wf_login.do**

This API uses the HTTP form POST method to login. Authorization information is to be put in HTTP headers as below:

HTTP Header Fields:

Field Name	Type	Description
authorization	string	Fixed value: AIDIANNO
identity	string	API user identity
password	string	API user password

Returned JSON:

Field Name	Type	Description
userid	long	User ID
token	string	Token string returned from server. The user's session token is used by other requests to ensure session security. The session timeout will be set to 5 minutes to guarantee session integrity



2. User Logout

- **API Endpoint: /interface/wf_logout.do**

This API uses the HTTP form POST method to logout and delete the Token object assigned to the current User object.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Session ID

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any

3. Get Current User Information

API Endpoint: /interface/wf_userinfo.do

This API uses the HTTP form POST method to get information about the current User.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Session ID



Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any
user	object	User Object with the following fields
user.id	int	ID of the user
user.username	string	Username of the user
user.firstname	string	First name of the user
user.lastname	string	Last name of the user
user.email	string	Email of the user
user.roles	string	User Roles of the user
user.groups	string	User Groups of the user

4. Change Password

- **API Endpoint: /interface/wf_changepwd.do**

This API uses the HTTP form POST method to change the password of the current account:

FORM fields:

Field Name	Type	Description
userid	long	User ID



token	string	Token
newpassword	string	New password

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Description message

5. Get Project List

- **API Endpoint: /interface/wf_searchproject.do**

This API uses the HTTP form POST method to return a paginated list of projects according to the query parameters (10 projects per page).

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
name	string	Find all projects where name contains a parameter value
ordering	string	Which field to use when ordering the results
page	int	A page number within the paginated result set



page_size	int	Number of results to return per page
-----------	-----	--------------------------------------

Returned JSON:

Field Name	Type	Description
total	int	Total number of projects.
projects	Project[]	Project object array

6. Get Project

- **API Endpoint: /interface/wf_getproject.do**

This API uses the HTTP form POST method to get information about a specific project.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any.
project	object	The project object with the



		following fields
project.id	int	ID of the project
project.name	string	Name of the project
project.owner	int	Owner id of the project
project.created_date	Datetime	Creation date
project.updated_date	Datetime	Date of last update

7. Get Tasks of Project

- **API Endpoint: /interface/wf_getprojecttasks.do**

This API uses the HTTP form POST method to get the tasks from a specific project.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any.
tasks	Task[]	Task object array



8. Get Task List

- **API Endpoint: /interface/wf_search.do**

This API uses the HTTP form POST method to identify tasks based on given search conditions

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the Project
name	string	Find all tasks where name contains a parameter value
owner	int	Find all tasks with the specified owner id
mode	string	Find all tasks with a specific mode: annotation, interpolation
assignee	int	Find all tasks with the specified assignee id
status	string	The status of task: annotation, validation, completed
ordering	string	Which field to use when determining order of results
page	int	A page number within the paginated



		result set
page_size	int	Number of results to return per page

Returned JSON:

Field Name	Type	Description
total	int	Total number of tasks.
tasks	Task[]	Task object array

9. Create Task

- **API Endpoint: /interface/wf_createtask.do**

This API uses the HTTP form POST method to create a single video/image annotation task in the workflow system.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Session ID
task	object	Task Object

Task Object contains an array of the following fields:

Field Name	Type	Description
projectid	string	ID of the project



id	int	Task ID, 0 for creation
url	string	Video/Image URL
name	string	Task name
mode	string	Task Mode: annotation, interpolation
owner	int	Owner ID of the task, 0 for creation
assignee	int	Assignee ID of the task, 0 for creation
language	string	The language of the task
batch	string	The batch number
annotatorCount	int	Number of Annotators per Task, Default is 1
created_date	Datetime	Created date, not required for creation
updated_date	Datetime	Updated date, not required for creation
z_order	boolean	Enables order for shapes. Useful for segmentation tasks
image_quality	number	Image compression level % (0-1.0)



status	string	Status of the task: annotation, validation, completed
labels	Label[]	Labels of this task with the following structure
label.id	int	ID of the label
label.name	string	Name of the label
Label. attributes	Attribute[]	Attributes of the label

The Attributes object has the following JSON formats

```
{
  "id": 0
  "name":"name of the attribute",
  "mutable": Boolean,
  "input_type":"checkbox, radio, number, text, select",
  "default_value":"string",
  "values":"string"
}
```

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any.
task	object	The task object been created

10. Import Tasks

- **API Endpoint:** /interface/wf_importtask.do



This API uses the HTTP form POST method to create a list of video/image annotation tasks in the workflow system.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Session ID
tasks	Task[]	Array of Tasks

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any
tasks	Task[]	The tasks that have been created

11. Get Task

- **API Endpoint: /interface/wf_gettask.do**

This API uses the HTTP form POST method to get information about a specific task.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project



taskid	string	ID of the task
--------	--------	----------------

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any
task	object	The task object

12. Update Task

- **API Endpoint: /interface/wf_importtask.do**

This API uses the HTTP form POST method to update a task.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Session ID
projectid	string	ID of the project
taskid	string	ID of the task
status	string	Set status field if provided
assignee	int	Set assignee field if provided

Returned JSON:



Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any
tasks	object	The updated task object

13. Remove Task

- **API Endpoint: /interface/wf_remove.do**

This API uses the HTTP form POST method to remove a created task.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	int	ID of the project
taskid	int	ID of the task to be removed

Returned JSON:

Field Name	Type	Description
success	boolean	True/False Tasks will not be able to be removed via API if the tasks have already been worked on by annotators.



msg	string	Error message if any
-----	--------	----------------------

14. Get Task Annotation

- **API Endpoint: /interface/wf_gettaskannotation.do**

This API uses the HTTP form POST method to get a task annotation result.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project
taskid	string	ID of the task
format	String	Annotation format to be exported

- Returned JSON:

Field Name	Type	Description
success	boolean	True
msg	string	Description message
annotation	Annotation	Annotation object in the following format:



- **The Annotation Object**

Field Name	Type	Description
version	int	Version
tags	LabeledImage[]	Array of labeled image objects
shapes	LabeledShape[]	Array of labeled shape objects
tracks	LabeledTracks[]	Array of labeled track objects

- **Labeled Image**

Field Name	Type	Description
id	int	ID
frame	int	Frame ID it belongs to
label_id	int	Label ID
group	int	Group ID it belongs to
attributes	AttrVal[]	Attributes array with the following structure: <pre>{ spec_id:int,</pre>



		value:string }
--	--	-------------------

• **Labeled Shape**

Field Name	Type	Description
id	int	ID
frame	int	Frame ID it belongs to
label_id	int	Label ID
group	int	Group ID it belongs to
z_order	int	Z order
occluded	boolean	Occluded
points	[number]	Array of points
attributes	AttrVal[]	Attributes array with the following structure: { spec_id:int, value:string }



- **Labeled Track**

Field Name	Type	Description
id	int	ID
frame	int	Frame ID it belongs to
label_id	int	Label ID
group	int	Group ID it belongs to
shapes	TrackedShape[]	Array of Tracked Shapes
attributes	AttrVal[]	Attributes array with the following structure: <pre>{ spec_id:int, value:string }</pre>

- **Tracked Shape**

Field Name	Type	Description
id	int	ID
type	string	Shape type:



		rectangle, polygon, polyline, points, cuboid
occluded	boolean	Occluded
z_order	int	Z order
points	[number]	Array of points
frame	int	Frame ID it belongs to
outside	boolean	
attributes	AttrVal[]	Attributes array with the following structure: <pre>{ spec_id:int, value:string }</pre>

15. Update Task Annotation

- **API Endpoint: /interface/wf_updatetaskannotation.do**

This API uses the HTTP form POST method to update a task annotation.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token



projectid	string	ID of the project
taskid	string	ID of the task
annotation	object	Annotation Object to be updated

- Returned JSON:

Field Name	Type	Description
success	boolean	True
msg	string	Description message

16. Delete Task Annotation

- API Endpoint: `/interface/wf_removetaskannotation.do`

This API uses the HTTP form POST method to remove all annotations of a task.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project
taskid	string	ID of the task

- 17. Returned JSON:



Field Name	Type	Description
success	boolean	True
msg	string	Description message

18. Get Task Data

- **API Endpoint: /interface/wf_gettaskdata.do**

This API uses the HTTP form POST method to get task specific data.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project
taskid	string	ID of the task
type	string	Specifies the type of the requested data: chunk frame preview
quality	string	Specifies the quality level of the requested data, doesn't matter for 'preview' type: compressed original
number	number	A unique number value identifying chunk or frame, doesn't matter for



		'preview' type
--	--	----------------

- Returned:

The actual image/data has been returned

19. Get Task Meta Data

- API Endpoint: /interface/wf_gettaskmeta.do**

This API uses the HTTP form POST method to get a task metadata.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project
taskid	string	ID of the task

- Returned JSON:

Field Name	Type	Description
success	boolean	True
msg	string	Description message
meta	object	Meta Data object in the following format



- **The Meta Data Object**

Field Name	Type	Description
chunk_size	int	
size	int	
image_quality*	int	
start_frame	int	
stop_frame	int	
frame_filter	string	
frames	Object[]	Array of Frame Object

- **The Frame Object**

Field Name	Type	Description
width	int	
height	int	
name	string	



20. Export Task Annotation

- **API Endpoint: /interface/wf_exporttask.do**

This API uses the HTTP form POST method to export a task annotation.

FORM fields:

Field Name	Type	Description
userid	long	User ID
token	string	Token
projectid	string	ID of the project
taskid	string	ID of the task
format	string	Desired output format name

- Returned:

Exported file

21. Get Server Information

- **API Endpoint: /interface/wf_getserverinfo.do**

This API uses HTTP GET method to get workflow server information.

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any.



info	string	Server information
------	--------	--------------------

22. Get Supported Annotation Formats

- **API Endpoint: /interface/wf_getannotationformats.do**

This API uses HTTP GET method to get the list of supported annotations formats.

Returned JSON:

Field Name	Type	Description
success	boolean	True/False
msg	string	Error message if any
DatasetFormats	object	Supported annotation formats. Object defined as below:
DatasetFormats. importers	DatasetFormat[]	Array of supported importers
DatasetFormats. exporters	DatasetFormat[]	Array of supported exporters

The DatasetFormat object has the following JSON formats

```
{
  "name": "name of the format",
  "ext": "file extention",
  "version": "version",
  "enabled": true
}
```