

Interact directly with SEI's version of JIRA, whereby apps can check or request status of tickets, add comments or attachments, and search across ticket filters.

TAGS: aml, backOffice, middleOffice, workflow, JIRA,

ENDPOINT: Create Task

ENDPOINT DESCRIPTION:

Creates a new Task along with sample task attachment for reference.

PROD URL: https://test.api.seic.com/api/v1/workdesk/ticket

TEST URL: https://test.api.seic.com/v2/workdesk/ticket

TLS URL*: https://mtls.api.seic.com/v2/workdesk/ticket

*The TLS endpoint is only used if Server Based Application was selected during the App Registration process.

STEP 1

OAuth Token

STEP 2

Pass the OAuth Token and the App key to invoke this endpoint for a particular application. Note: 1 to 5 API endpoints are sequenced and remaining can be randomly invoked. This endpoint will create a new ticket in a particular project. .

REQUEST PARAMETERS:

Name	Description	Data Type	Parameter Type
Authorization	The access_token value obtained from the Token endpoint. This is passed as "Bearer access_token".	string	header
AppKey	This is the Consumer Key provided during the App Registration process and is used to identify the user. It is a form of user credential generated and provisioned by the SEI Developer Portal for applications associated with individual developers.	string	header
Content-Type	Request content type. The valid values are multipart/form-data; boundary=123456789.	string	header

BODY PARAMETERS:

Name	Description	Data Type	Parameter Type
ticketModel		object	body

SAMPLE REQUEST:

```
curl -X POST \
  https://api.seic.com/v2/workdesk/ticket \
  -H 'appkey: gsedgaerhDSHGRSH' \
  -H 'authorization: Bearer ADsgdnnkjnfIKJN' \
  -H 'content-type: multipart/form-data; boundary=123456789' \
  -d '--123456789--
Content-Disposition: form-data; name="ticketModel"
```

```
{
  "fields" :{
    "project" : { "id" : "23233" },
    "issuetype" : {"name" : "Document Review"},
    "summary": "Description Test",
    "customfield_23440": "Fund",
    "customfield_23450": "ABCDE1234",
    "customfield_12450": "2021-8-8"
  }
}
--123456789--
```

REQUEST HEADER:

```
Authorization: Bearer ADsgdnnkjnfIKJN
AppKey: gsedgaerhDSHGRSH
```

RESPONSE PAYLOAD MODEL:

Name	Description	Data Type(L)
Id	Project Id.	string
Key	Task number.	string

RESPONSE PAYLOAD MODEL:

Name	Description	Data Type(L)
id		string
key		string
self		string

RESPONSE ERRORS:

Name	Description
200	OK- The request has succeeded and the response has been returned.
400	Bad request - The server cannot or will not process the request due to something that is perceived to be an error on the client application side.
401	Unauthorized - Invalid authentication details have been provided. Also useful to trigger an authorization pop up if the API is used from a browser.
500	Internal Server Error - The server was unable to fulfill the request due to an unknown condition.

SAMPLE RESPONSE:

```
{
  "id": "1000000",
  "key": "TC-12345"
}
```

RESPONSE HEADER:

```
Content-Type: application/json
Status: 200 OK
```

