

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:** Project Issues

**ENDPOINT DESCRIPTION:**

Returns a list of issues in a project by project Id.

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

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

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

\*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.

**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
projectId	A valid Jira Project Id.	string	query

**SAMPLE REQUEST:**

```
curl -X GET \
  https://api.seic.com/v2/workdesk/projectissuetype?projectId=23233\
  -H 'appkey: gsedgaerhDSHGRSH' \
  -H 'authorization: Bearer ADsgdnnkjnfIKJN' \
```

**REQUEST HEADER:**

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

**RESPONSE PAYLOAD MODEL:**

Name	Description	Data Type(L)
Id	Project Id.	string
Name	Project Name.	string
IsSubTask	Defines whether the issue is a subtask or not.	string

**RESPONSE PAYLOAD MODEL:**

Name	Description	Data Type(L)
Id		string
Name		string
IsSubTask		boolean

**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": 10000,
    "Name": "Maintenance",
    "IsSubTask": false
  },
  {
    "Id": 12003,
    "Name": "Test Update",
    "IsSubTask": false
  },
  {
    "Id": 13005,
    "Name": "Subscription",
    "IsSubTask": false
  },
  {
    "Id": 19007,
    "Name": "Document Review",
    "IsSubTask": false
  }
]
```

**RESPONSE HEADER:**

```
Content-Type: application/json
```

Status: 200 OK  
requesttrackingid: 67e1ff68-164f-03ad-0f2d-5cbbfda56ec9