

**webMethods CloudStreams Provider for DIGITE SwiftKanban**

Version 10.11 | 30 January 2022

Contents

[1 Document Change History 3](#_Toc92778028)

[2 What is webMethods CloudStreams Provider for Digite SwiftKanban? 4](#_Toc92778029)

[3 Steps to create the connector connection 5](#_Toc92778030)

[4 webMethods CloudStreams Provider for Digite SwiftKanban Connector 6](#_Toc92778031)

[4.1 Connector Details 6](#_Toc92778032)

[4.1.1 Supported Resources 6](#_Toc92778033)

[4.1.2 Usage Notes 6](#_Toc92778034)

# Document Change History

|  |  |
| --- | --- |
| **Document revision date** | **Summary of changes** |
| 30 January 2022 | First release of this document. |

# What is webMethods CloudStreams Provider for SwiftKanban?

SwiftKanban is a Work Management Tool for helping you manage your work effectively and improve continuously. SwiftKanban leverages the simple yet powerful principles of the Kanban Method. It combines highly Configurable and Visual Workflow modelling, powerful and actionable Flow Metrics and ease of use to give you the best Kanban, Scrum or Scrumban tool.

Whether you are a single team (Marketing, HR, Procurement, Legal, Software, IT, DevOps – pretty much ANY function), or an Enterprise looking to accelerate your Lean/ Agile Transformation Initiative or your Portfolio and Risk Management implementation, SwiftKanban is for you!

Below is the snapshot of SwiftKanban homepage.

Graphical user interface, application

Description automatically generated

* SwiftKanban provider can be used to add and update sprints, backlogs, tasks, comments, attachments, relations in new workspace ( called Board in SwiftKanban).
* It can fetch all the activity logs mentioned in cards, backlogs and relations.
* We can also fetch the details of sprints, cards, backlogs, relations or any comments.
* This provider can also perform delete opeartion on any card, sprints, backlogs, tasks, comments, attachment.
* We can get all the Team members’ details for your account in SwiftKanban.
* We can fetch the Active Master Lists details for your account at an Org level.
* The connector can be integrated to automate all the above-mentioned activities.
* It enables better management of workflow and gives higher degree of transparency amongst teams.

For more information about SwitKanban, click [here](https://www.digite.com/knowledge-base/swiftkanban/learn-swiftkanban/).

For more information about how to configure and use CloudStreams connectors with webMethods CloudStreams, see the Administering webMethods CloudStreams document available in the **webMethods** section of the [Software AG Documentation](http://documentation.softwareag.com) web page.

# Steps to create the connector connection

Follow the below steps to create Connector connection.

* Create a connection for a published Connector.  
  Go to **Integration Server Admin page > Solutions > CloudStreams > Providers > Digite > SwiftKanban from the connectors list > Configure New Connection.**

Graphical user interface, text, application

Description automatically generated

* To Configure connection, we need Authorization Token.

The authentication method for the REST APIs is JWT token-based authentication with no expiry of token. Follow the below steps to retrieve a user’s API authorization token from SwiftKanban home page.

1. Login to your SwiftKanban web application.

Graphical user interface, application

Description automatically generated

1. Click on the profile picture in the upper right corner.

Graphical user interface, application

Description automatically generated

1. Next, click the username to open the profie view window.

Graphical user interface, application

Description automatically generated

1. Now, click the Authorization Token on the Side toolbar at the left to access your authorization token. Graphical user interface, application

   Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

* Once the Authorization token is generated, in the Basic view provide the connection details by selecting the IS Package, Connection Type, Folder name, Connection name, Server URL, and Authorization token.

Graphical user interface, application

Description automatically generated

* Enable the connection after creating the connection.

Graphical user interface, text, application

Description automatically generated

# webMethods CloudStreams Provider for SwiftKanban Connector

## Connector Details

The connector details include:

* **SaaS Provider**: SwiftKanban
* **API Version:** 3.0.1
* **API Type**: REST
* **Developer**: Chethan BG, Rekha Kumari
* **Group**: Digite
* **CloudStreams Minimum Version Compatibility**: 10.11
* **Provider Package Name**: WmSwiftKanbanProvider

### Supported Resources

List down resources supported by the provider:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service: Bundle** | | | | |
| S.No | **Name** | **Path** | **Method** | **Description** |
| 1 | AccountService | /account-operations/accounts | GET | Fetches Account details |
| 2 | ActivityLogService | /activity-log/boards/{boardId}/todos/{id} | GET | Fetches activity log of ToDo |
| /activity-log/relations | GET | Fetches activity log of all the relation |
| /activity-log/boards/{boardId}/cards | GET | Fetches activity log of cards |
| 3 | AttachmentService | /attachment-operations/boards/{boardId}/cards/{cardId}/attachments | GET | Gets all the attachments of the card |
| /attachment-operations/boards/{boardId}/cards/{cardId}/attachments | POST | Adds attachment to the card |
| /attachment-operations/boards/{boardId}/cards/{cardId}/attachments | DELETE | Deletes multiple attachment from the card |
| 4 | RelationService | /relation-operations/boards/{boardId}/cards/relations | GET | Fetches the list of all the relations among all the cards of the specified board |
| /relation-operations/boards/{boardId}/cards/{id}/relations/delete | PUT | Deletes relation between cards |
| /relation-operations/boards/{boardId}/cards/{id}/relations | GET | Fetches the list of all the relations with the card |
| /relation-operations/boards/{boardId}/cards/{id}/relations | POST | Adds a relation between cards |
| 5 | CardService | ​/card-operations​/boards​/{boardId}​/cards | GET | Fetches cards details |
| ​/card-operations​/boards​/{boardId}​/cards | PUT | Updates cards |
| ​/card-operations​/boards​/{boardId}​/cards | POST | Add cards in the Backlog or the custom column as defined in the Board Policies |
| ​/card-operations​/boards​/{boardId}​/cards | DELETE | Deletes cards |
| ​/card-operations​/boards​/{boardId}​/card-types​/{cardType}​/startDate​/{startDate}​/cards | GET | Fetches cards from specified date |
| ​/card-operations​/boards​/cards​/ext-creator-id​/{emailid} | GET | Fetches cards basic details across boards by using ext\_creator\_id |
| ​/card-operations​/boards​/{boardId}​/cards​/discard-abort | PUT | Discards/Aborts the cards |
| ​/card-operations​/boards​/{boardId}​/cards​/transfer | PUT | Transfers the cards |
| ​/card-operations​/boards​/{boardId}​/cards​/convert | PUT | Converts cards |
| ​/card-operations​/boards​/{boardId}​/cards​/move​/backlog | PUT | Moves cards to backlog |
| ​/card-operations​/boards​/{boardId}​/cards​/flag | PUT | Adds Flags on cards |
| ​/card-operations​/boards​/{boardId}​/cards​/archive | PUT | Archives the cards |
| ​/card-operations​/boards​/{boardId}​/cards​/{id} | GET | Fetches card details |
| ​/card-operations​/boards​/{boardId}​/cards​/{id} | DELETE | Deletes card |
| ​/card-operations​/boards​/{boardId}​/cards​/move​/board | PUT | Moves cards on board |
| ​/card-operations​/boards​/{boardId}​/cards​/move​/ready | PUT | Moves cards to ready |
| ​/card-operations​/boards​/{boardId}​/cards​/unflag | PUT | Removes Flags from cards |
| ​/card-operations​/boards​/{boardId}​/cards​/{id}​/new​/reference | GET | Fetches new reference of converted and transfered card |
| 6 | CommentService | ​/comment-operations​/boards​/{boardId}​/cards​/{id}​/comments | GET | Fetches list of comments added to card along with comment details |
| ​/comment-operations​/boards​/{boardId}​/cards​/{id}​/comments | PUT | Update comment |
| ​/comment-operations​/boards​/{boardId}​/cards​/{id}​/comments | POST | Adds comment to the card |
| ​/comment-operations​/boards​/{boardId}​/cards​/comments | GET | Fetches the list of comments from multiple cards based on the filter criteria. |
| ​/comment-operations​/boards​/{boardId}​/cards​/{id}​/comments​/reply | POST | Replies on comment |
| 7 | MetadataService | ​/metadata-operations​/masterlist | GET | Fetches master list details for account |
| ​/metadata-operations​/items​/{itemType}​/metaData | GET | Fetches Meta Data details along with content |
| ​/metadata-operations​/relation-types | GET | Fetches the list of all the relation types |
| ​/metadata-operations​/masterlist​/{id} | GET | Fetches masterlist details |
| ​/metadata-operations​/boards​/{boardId}​/item-types​/{itemType}​/dck​/{dckKey}​/lovData | GET | Fetches list of items allowed in the dynamic data list |
| ​/metadata-operations​/card-types​/{cardType}​/relations | GET | Fetches a list of all the card types which have parent-child or other links with another card type |
| ​/metadata-operations​/boards​/{boardId}​/cardTypes | GET | Fetches all the card types available in a board |
| 8 | BoardService | ​/board-operations​/boards​/{boardId}​/lanes | GET | Fetches smartLane details along with content |
| ​/board-operations​/boards​/{boardId}​/policies | GET | Fetches all policies for particular board |
| ​/board-operations​/boards​/{boardId}​/dynamicDataList | GET | Fetches all dynamic data list and their list of items respectively for particular board |
| ​/board-operations​/boards | GET | Fetches board details along with content |
| ​/board-operations​/boards​/{boardId}​/lanes​/{laneId}​/cards | GET | Fetches smartlane data with queues and cards |
| ​/board-operations​/boards​/{boardId}​/config | GET | Fetches Lane details along with queues and subqueues |
| ​/board-operations​/boards​/{id} | GET | Fetches board detail for a given boardId along with content |
| 9 | ReleaseService | ​/release-operations​/boards​/{id}​/releases | PUT | Updates selected fields of a release passed in request |
| ​/release-operations​/boards​/{id}​/releases | POST | Creates a new release in a board |
| ​/release-operations​/boards​/{boardId}​/releases​/{id}​/tag | PUT | Tags cards to a release |
| ​/release-operations​/boards​/{boardId}​/releases​/{id}​/untag | PUT | Untags cards from a release |
| ​/release-operations​/boards​/{boardId}​/releases​/{id}​/delete | DELETE | Deletes a release from a board |
| ​/release-operations​/boards​/{boardId}​/releases​/{id} | GET | Fetches the release present in a board |
| ​/release-operations​/boards​/{boardId}​/releases​/{id}​/close | PUT | Closes a release from a board |
| ​/release-operations​/boards​/{boardId}​/releases | GET | Fetches list of all the releases present in a board |
| ​/release-operations​/boards​/{boardId}​/releases​/{id}​/cards | GET | Fetches details of all the cards tagged to the release |
| 10 | SprintService | ​/sprint-operations​/boards​/{boardId}​/releases​/{id}​/sprints | GET | Fetches details of all the sprints in a release |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints​/{id}​/delete | DELETE | Deletes a sprint from a board |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints​/{id}​/cards | GET | Fetches details of all the cards tagged to the sprint |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints | GET | Fetches the sprints present in a board |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints | PUT | Updates selected fields of a release passed in request |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints | POST | Creates a new sprints in a board |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints​/{id}​/close | PUT | Closes a sprint from a board |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints​/{id} | GET | Fetches the sprint present in a board |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints​/{id}​/tag | PUT | Tags a card to an sprint |
|  |  | ​/sprint-operations​/boards​/{boardId}​/sprints​/{id}​/untag | PUT | Untags a card from an sprint |
| 11 | TeamMemberService | ​/team-member-operations​/boards​/team-members | GET | Fetches board-wise team members details |
| ​/team-member-operations​/boards​/{boardId}​/team-members | GET | Fetches Team Members details in a board |
| ​/team-member-operations​/boards​/{boardId}​/team-members | POST | Invites team members in board |
| 12 | TeamService | /team-operations/teams | GET | Fetches Teams details for an account |
| 13 | ToDoService | ​/todo-operations​/boards​/{boardId}​/cards​/{cardId}​/todos | GET | Modifies ToDo on card |
| ​/todo-operations​/boards​/{boardId}​/cards​/{cardId}​/todos | PUT | Create ToDo on card |
| ​/todo-operations​/boards​/{boardId}​/cards​/{cardId}​/todos | POST | Deletes ToDos of a card |
| ​/todo-operations​/boards​/{boardId}​/cards​/{cardId}​/todos | DELETE | Fetches details of todos in a given time interval. |
| ​/todo-operations​/boards​/{boardId}​/cardType​/{cardType}​/todos​/startDate​/{startDate} | GET | Deletes ToDo of a card |
| ​/todo-operations​/boards​/{boardId}​/cards​/{cardId}​/todos​/{todoId} | DELETE | Gets ToDo from card |
| ​/todo-operations​/boards​/{boardId}​/cards​/{cardId}​/todos​/{id} | GET | Modifies ToDo on card |
| 14 | TokenService | /secured/auth | POST | Generated AuthorizationToken |
| 15 | UserService | ​/user-operations​/users​/{id} | GET | Fetches user and other user information |
|  |  | ​/user-operations​/users​/{id} | DELETE | Deletes user |
|  |  | ​/user-operations​/users | GET | Fetches all users details |
|  |  | ​/user-operations​/users | POST | Adds users |
|  |  | ​/user-operations​/users | DELETE | Deletes multiple users |
|  |  | ​/user-operations​/users​/token | GET | Fetches user details information |

For more REST APIs related information, click [here](https://login.swiftkanban.com/swift-api-doc/#/).

### Usage Notes

Any special notes