

# **Tencent Integration Platform**

## **Product Introduction**

### **Product Documentation**



## Copyright Notice

©2013-2023 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

## Product Introduction

Overview

Concepts

Integration Apps

Messages

Components

Strengths

Use Cases

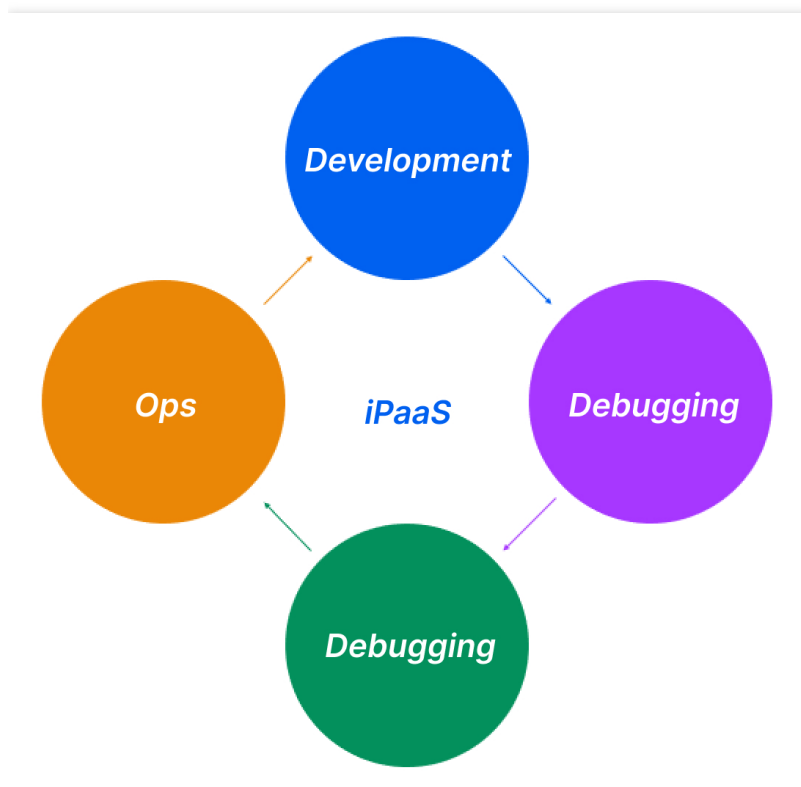
# Product Introduction

## Overview

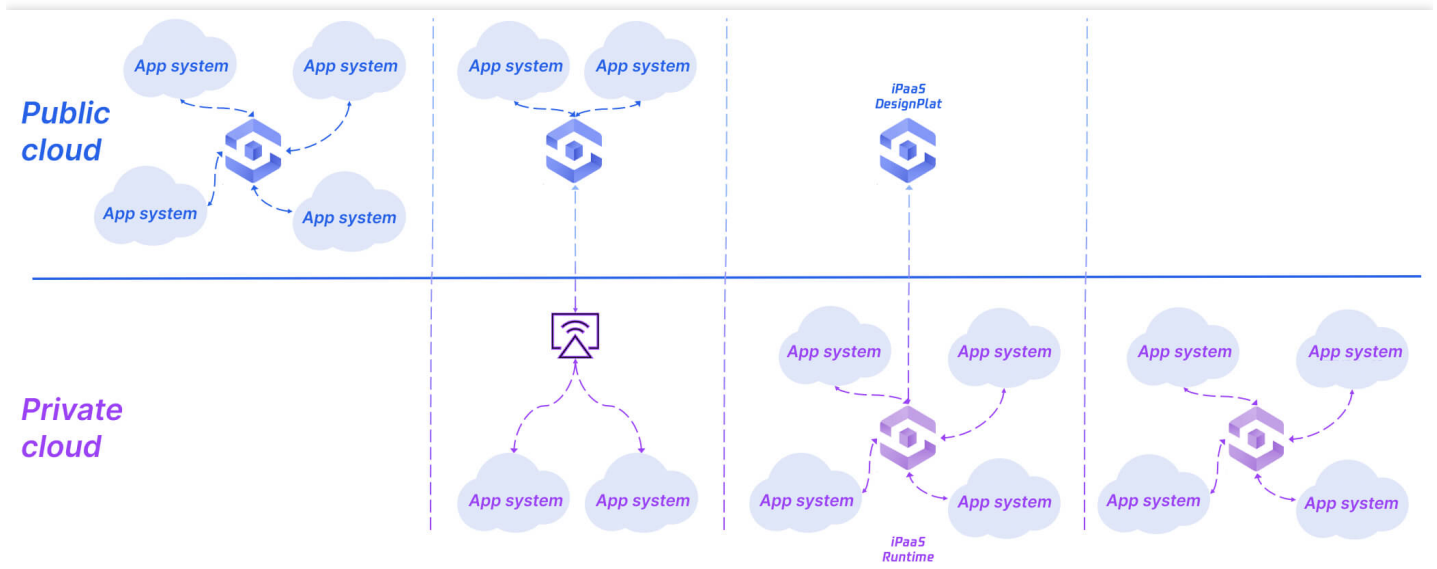
Last updated : 2023-08-04 09:46:04

### Overview

As a new type of cloud integration service, Tencent Integration Platform(iPaaS) fully utilizes elastic scaling, container, resource management, and self-service capabilities to connect different systems and applications in and outside your organization to the same platform through little to no code. With best integration practices and the ability to quickly build system integration models, it offers various features, including resource integration, data orchestration, and business connection between systems, making it a lightweight, comprehensive, and flexible integration system. It also provides rich application management and Ops capabilities as well as visual monitoring and log pages to further reduce your operational costs. iPaaS offers a full range of integration capabilities that combine development, debugging, publishing, and Ops, greatly improving your integration efficiency and reducing the late Ops costs of integration tasks.



As enterprises' internal and external systems have complicated network topologies and highly sensitive data and are managed in multiple or hybrid clouds, iPaaS provides two solutions: in-cloud and off-cloud business system connection through a security gateway and on-premises platform deployment. It breaks through the barriers of different regions and architectures so that data can flow throughout the entire domain, while guaranteeing the security of business data and the reliability of integrations.



## Features

### Communication protocol connector

iPaaS offers mainstream communication protocol connectors such as FTP, HTTP, Advanced Message Queuing Protocol (AMQP), and Applicability Statement (AS) 1/2/3/4 connectors.

### SaaS application connector

As SaaS applications gain popularity, connectivity of out-of-the-box applications is becoming increasingly important. Through the SaaS application connector, you can simply connect different services together. iPaaS has published connectors for over 200 applications, including WeCom, Tencent Meeting, Tencent QiDian, and Tencent Lexiang. In addition, it also opens up an application connector ecosystem to empower more SaaS vendors to publish their own application connector components based on the SDK.

### Data mapping and transformation

As the core features of iPaaS, data transformation and orchestration eliminate the technological differences between heterogeneous applications, making different application services communicate and collaborate with each other smoothly.

After connecting your data to iPaaS, you can orchestrate and convert the formats and structures of the input data to those of the target system; for example:

- Data format conversion: XML to JSON.
- Data mapping: Data structure conversion and field mapping.
- Content modification: Parameter modification and variable assignment.

## Message routing

In many scenarios, such as data splitting, aggregation, sorting, and distribution, the message direction in a flow needs to be controlled. iPaaS provides diverse components for controlling the flow of messages. For example, the Choice component can act like if-else branch logic in program design to first match and judge the first if condition. If the preorder condition is hit, the component will process the nodes under the condition; otherwise, it will continue matching the subsequent if conditions.

## Versioning

iPaaS allows you to publish different versions of your integration apps to cope with dynamic requirements such as business changes and version upgrades.

## Integration app log

iPaaS offers real-time and historical logs of your integration apps to help you with security analysis, compliance auditing, resource tracking, and problem locating.

## Secure transfer

iPaaS offers multiple data security mechanisms, including data security (sensitive data encryption), system security, network security (firewall), and business security (tenant isolation).

## Development capabilities

Dataway is an expressive script language for custom data conversion and processing in iPaaS. It is integrated in the iPaaS runtime service and plays a key role in implementing iPaaS extensibility.

iPaaS's many built-in components can provide Dataway script-based customization capabilities to dynamically process connector events.

# Concepts

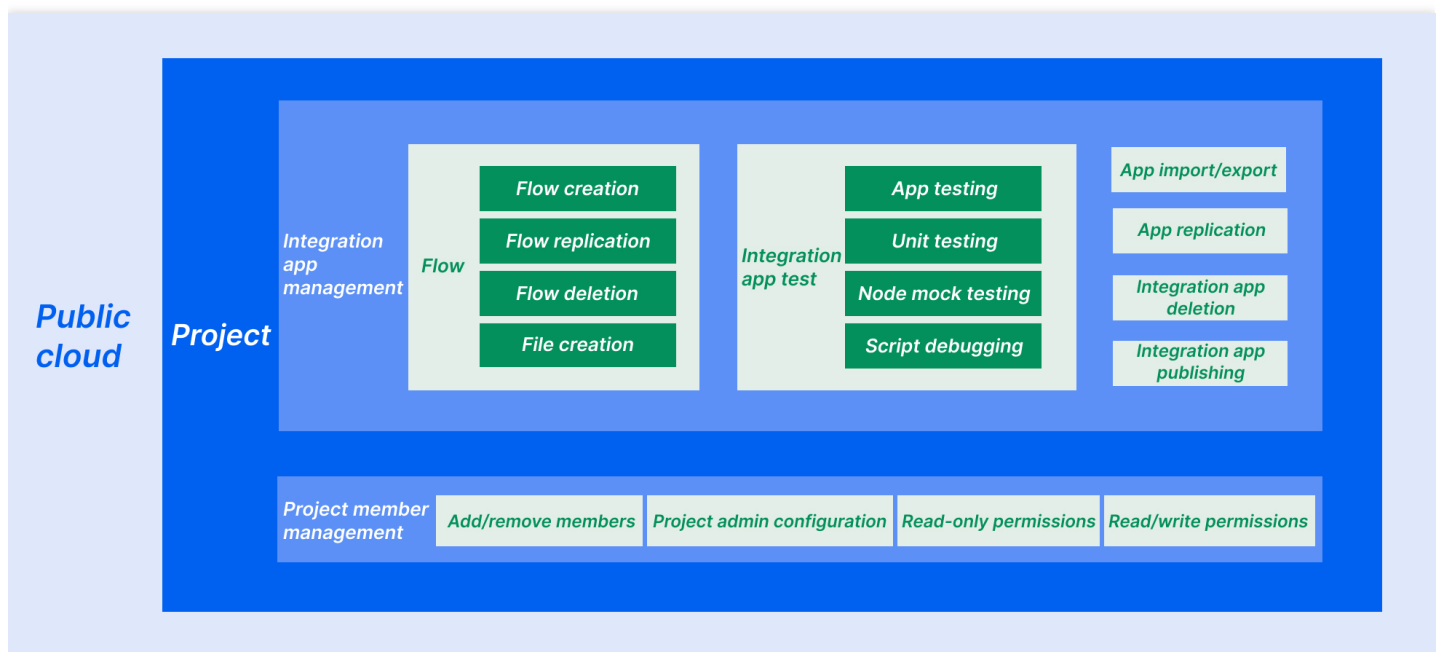
## Integration Apps

Last updated : 2023-08-03 16:59:24

Key concepts of iPaaS include integration apps, messages, components, and security gateways. This document describes integration apps.

### Overview

Integration apps are an important capability of iPaaS that make it easier to cope with complex system integrations. For example, a company's internal business systems, including OA, HR, ERP, CRM, and proprietary systems, have independent architectures, which prevents their business logic and data from flowing together easily and makes business processing slow and inefficient. For these complex scenarios, an integration app can be created to interconnect the multiple business systems. You can create multiple integration apps within one project to handle specific integration scenarios. In the above example, an integration app can be created to integrate the OA and HR systems or interconnect the data among the ERP, CRM, and proprietary systems.



### Components

#### Integration apps

An integration app is a way to interconnect systems. You can create an integration app to interconnect data from multiple systems through the use of connectors. Each integration app corresponds to a specific integration scenario. In the above example, the company's large-scale integration includes two integration apps; one is for the OA and HR systems, and the other is for the ERP, CRM, and proprietary systems.

## Flows

A flow is one of the features used to integrate a single application in a system. An integration app can have one or multiple flows, depending on the business complexity. Flows can be divided into main flows and subflows. The starting node of a main flow must be a trigger [component](#) such as HTTP Listener or Scheduler. The starting node of a subflow isn't necessarily a trigger, and a subflow can be referenced by another flow on a node as a branch to complete a series of data interconnection operations. For example, in the above example, the company's large-scale integration has an integration app for the ERP, CRM, and proprietary systems. The ERP information can be synced to the proprietary systems as the main flow, and sync calls of information between the CRM and ERP systems can run as subflows.

You can orchestrate flows freely on the canvas. You can add different [components](#) on each node as needed to call business system APIs, adapt to protocols, convert data, and implement logics like loop, choice, and concurrency. During flow orchestration, you can also use various debugging capabilities, including app testing, unit testing, and node mock testing, to publish the integration app to the sandbox environment and check whether the input and output of each node are as expected. After debugging, you can publish the integration app to the production environment to start the integration.

## Connector roles

iPaaS is a product in the Tencent Cloud console. It can be logged in to with a Tencent Cloud account and supports access management through [CAM](#). You can be redirected to CAM from the iPaaS console to add sub-accounts under your Tencent Cloud root account, and iPaaS will automatically pull all sub-accounts under the root account to the member list. For example, ordinary member A has read/write permissions for the integration app for the OA and HR systems, while ordinary member B has the read-only permission for the same integration app.

The system admin and project admin can add a user to a project so the user can participate in co-development of integration apps. The system admin can grant an ordinary user the project admin privileges or revoke the privileges. The system admin and project admin can both grant ordinary users read-only permission or read/write permission for specified integration apps within the project.

## System admin

A system admin has the highest privileges of iPaaS. A Tencent Cloud root account automatically has system admin privileges which cannot be revoked. A system admin can grant any member the system admin and project admin privileges or revoke the privileges. They can also grant ordinary users in a project the read-only permission or read/write permissions for a specified integration app within the project.



**Project admin**

A project admin can grant or revoke the project admin privileges and grant any users in the integration project the read-only permission or read/write permissions for specified integration apps.

**Ordinary member**

All sub-accounts under a Tencent Cloud root account will automatically become ordinary members in iPaaS.

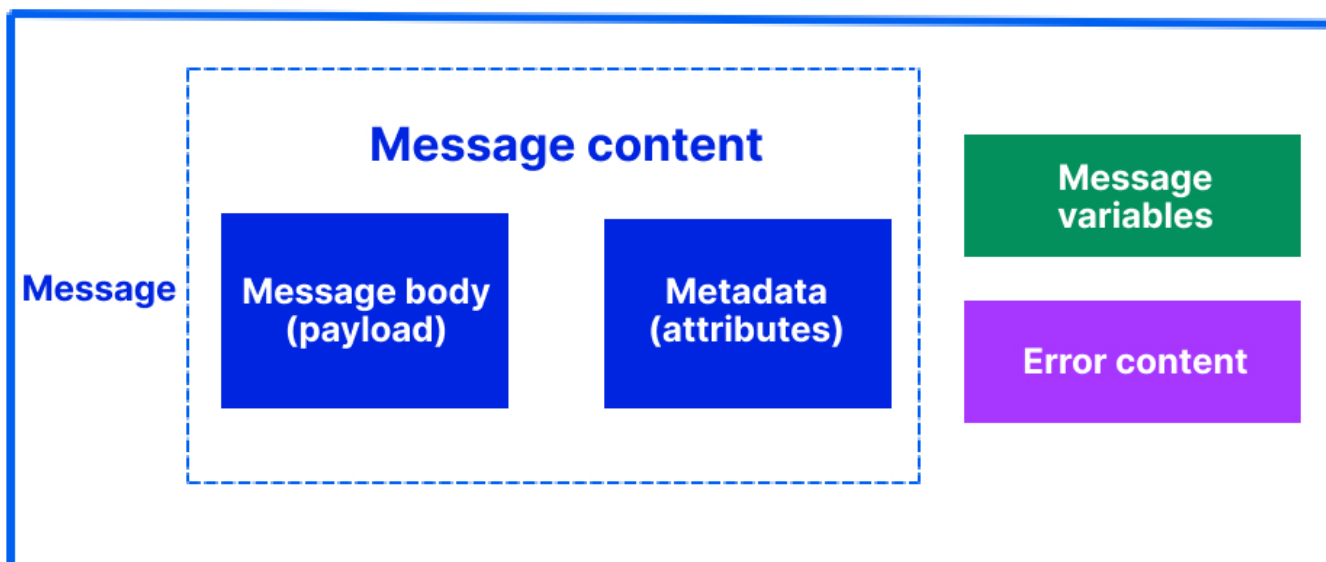
# Messages

Last updated : 2023-08-03 16:59:24

Key concepts of iPaaS include integration apps, messages, components, and security gateways. This document describes messages.

A message is a carrier of runtime data such as metadata in iPaaS. Messages are transferred between components in a flow according to the predefined logic. When a trigger begins the execution of a flow, it will encapsulate the raw input data into a message and pass it in to a component as parameters. After processing the message, the component will return a new message, which will be the input to the next component, and so on until the last component in the flow ends its execution and returns a message with the returned values of the entire flow. A message consists of the following parts:

- **Message content:** Contains the body of the message ( `payload` ) and metadata ( `attribute` ). The former is the core information, while the latter is auxiliary information.
- **Message variable:** Indicates the intermediate value stored temporarily during message transfer in the flow.
- **Error content:** An error.



# Components

Last updated : 2023-08-03 16:59:24

Key concepts of iPaaS include integration apps, messages, components, and security gateways. This document describes components.

Common connectors	Application connectors				Logical components

## Overview

Components are the minimum units in flow orchestration and include logical components and connectors. Each node in a flow is a component. You can enter parameters or use other methods to configure the component as needed.

## Components

### Logical components

Logical components encapsulate common coding logic, such as Set Variable (variable creation) and Choice (conditional branch execution). If you want to add a logic to a business process, you only need to add the corresponding component on the node and configure it. For example, after adding the Set Variable component, you only need to enter the variable name and value; after adding the Choice component, you can add and delete conditional branches and configure the execution condition for each branch.

Currently, logical components cover five logic scenarios: flow control, data processing, errors and logs, parallelism and concurrency, and functionality.

Logical Component	Description
Async	<b>Async</b> is a flow control component, where you can configure a subflow to execute an async task, which is similar to creating a thread.
Break	Break needs to be used together with the <b>For Each</b> or <b>While</b> component to break a loop.
Cache	<b>Cache</b> is used to cache the intermediate data generated during flow execution. Different flows in the same application can share the data cached in the <b>Cache</b> component.
Choice	<b>Choice</b> is a branch selection statement. Similar to if-else, it is used to execute actions by condition.
Continue	Similar to the <b>Break</b> component, <b>Continue</b> also needs to be used together with the <b>For Each</b> or <b>While</b> components. It is used to jump out of the current loop and execute the next loop.
Flow Reference	<b>Flow Reference</b> is used to reference other flows in the current application. Different from <b>Async</b> , <b>Flow Reference</b> is a sync action. It will continue to execute the next action only after the execution of the referenced flow is completed.
For Each	<b>For Each</b> is a loop control structure, which is similar to for/foreach in programming languages. In <b>For Each</b> , you can configure a subflow to process the specified data set in a loop but cannot specify the number of loops.
Logger	<b>Logger</b> is used to output logs in the console. Currently, it supports four log levels: <b>DEBUG</b> , <b>INFO</b> , <b>WARN</b> , and <b>ERROR</b> .
Parallel Foreach	<b>Parallel Foreach</b> is used to execute parallel tasks. It executes the same processing on data set elements in parallel. The number of parallel elements is subject to the configured maximum number of parallel elements. If the number of elements is less than or equal to the maximum number of parallel elements, the number of parallel elements is equal to the number of elements; otherwise, it is the configured maximum number.
Raise Error	<b>Raise Error</b> is used to report exceptions and stop flow execution. This component can be used alone or together with the <b>Try</b> component.

Logical Component	Description
Remove Variable	The opposite of the <b>Set Variable</b> component, <b>Remove Variable</b> is used to delete the specified variable in <code>message</code> .
Scatter Gather	<b>Scatter Gather</b> can execute multiple tasks in parallel. This component has two core configuration items: the number of parallel tasks (2–8 currently) and worker. In each worker, a task to be executed can be configured.
Scheduler	<b>Scheduler</b> is used to trigger a flow at the scheduled time according to the configured rule. Its configuration contains one or more cron rules. When any cron rule matches the current time, the flow where the <b>Scheduler</b> component resides will be triggered.
Set Payload	<b>Set Payload</b> is used to set the <code>payload</code> attribute in <code>message</code> . It supports expressions and literal values. To enter a literal value, select the data type and enter the literal value in the input box. To enter an expression, select the <code>any</code> type and write an expression.
Set Variable	<b>Set Variable</b> is used to declare a variable and save it in <code>variables</code> in <code>message</code> , so that subsequent nodes can import it in the form of <code>msg.vars.get('name')</code> .
Synchronized	<b>Synchronized</b> is a distributed lock used to lock and protect concurrent requests. It can be used in scenarios with conflicting resource access requests.
Transform	<b>Transform</b> is used to orchestrate and convert the format of the data in <code>message</code> . It can modify <code>payload</code> , <code>attribute</code> , and <code>variable</code> .
Try	<b>Try</b> is used to capture errors. It can capture errors reported by a subflow executed in <b>Try</b> as well as system errors. It can also be used together with the <b>Raise Error</b> component to capture custom errors.
Until Successful	In <b>Until Successful</b> , you can configure a subflow to retry the subflow execution. It supports three retry modes: unconditional retry, retry based on success condition, and retry based on failure condition.
While	<b>While</b> is a loop component, which supports up to 10,000 loops. Unlike the <b>For Each</b> component, the <b>While</b> component doesn't rely on a data set for loop but uses conditional statements to determine whether the loop should continue. If the loop condition is <code>True</code> , the loop will continue until <code>False</code> is returned or the maximum number of loops is reached.
Compress	<b>Compress</b> is used to compress and decompress the data. It supports the following compression and decompression formats: DEFLATE, GZIP, and ZLIB.

## Connectors

A connector is a comprehensive encapsulation of a system interaction protocol (common connectors such as Database) or a specific business system (application connectors such as Tencent Meeting), which contains authentication, protocol conversion, and feature APIs. You can use the Database connector to directly access a database or use the Tencent Meeting connector to directly call the APIs of Tencent Meeting with no need to understand how Tencent Meeting interacts with external systems or how API authentication works. A connector enables you to connect to your business systems at zero costs.

- Common connectors
- Application connectors
- Custom connectors
- Connections
- Actions
- Triggers
- Expressions

A common connector is an encapsulation of a protocol such as HTTP, SOAP, Kafka, or MQ. You can select a protocol for system connection based on the actual conditions of your business system.

# Strengths

Last updated : 2023-08-03 16:59:24

## Enterprise-grade cloud integration platform

iPaaS guarantees high availability and security as well as disaster recovery, service-level agreement (SLA), and all-around technical support.

## Comprehensive integration capabilities

iPaaS supports diverse integration scenarios, including application, data, B2B ecosystem, process, and IoT integration.

## Open platform

iPaaS sets up a developer and technology community and provides rich developer tools as well as diverse out-of-the-box connectors and application components.

## Flexible orchestration with integration apps

iPaaS allows you to develop complex integration business logic simply by dragging and dropping components and setting parameters on the configuration page in the console. It also supports script languages and provides a flexible development mode, enabling business and technical personnel to create complex integration scenarios and greatly improve their integration flow orchestration efficiency and capabilities.

## Preset SaaS connectors

iPaaS comes with out-of-the-box Tencent Meeting, Tencent Qidian, and Tencent Lexiang connectors for you to quickly connect to the relevant products, simplifying your integration design and development.

## Reduced integration TCO

Compared with other traditional system integration products, iPaaS offers a unified integration platform that incorporates data integration experience and best practices from throughout the industry, which greatly reduces costs and accelerates development by minimizing the need for custom development.

# Use Cases

Last updated : 2023-08-03 16:59:24

iPaaS supports various integration scenarios. It can connect the information silos between your applications on and off the cloud, data, IoT, devices, and partners for cross-cloud and cross-region interconnection and information sharing. You can quickly and easily connect to your systems in the console and integrate them by creating, deploying, and managing integration apps. In addition, the integration platform of iPaaS offers no-code or low-code self-delivery capabilities for integrations, effectively coping with complex business scenarios and making it easier for you to quickly integrate your systems.

## Data sync

Data sync refers to migrating a certain set of data from one system to another at a specified time point.

- **Challenges:** It takes the development and operations teams a lot of time to develop and manage custom applications to meet the data sync needs.
- **Solution:** The data sync mode allows you to create an integration service for automatic data migration to automatically transfer the data within the specified scope to downstream applications. Such a reusable service can save a lot of time for development and operations teams.

## Data broadcasting

Data broadcasting is to continuously move data from a single source system to multiple target systems in real time or quasi-real time.

- **Challenges:** The data source system needs a custom data sending service, which increases the Ops pressure on the non-core businesses of the original system.
- **Solution:** The data source system is connected to iPaaS, so as to leverage its data distribution capabilities to connect to the downstream data consumption systems flexibly.

## Data aggregation

Data aggregation is to receive data from multiple systems and move or replicate the data to another system.

- **Challenges:** You need to be familiar with each system and perform custom development and integration.
- **Solution:** You can connect to multiple systems through iPaaS configuration and aggregate data in iPaaS to meet the business needs.

## Two-way data sync

Two-way sync is to merge two or more datasets from two or more systems into one system where different datasets can be distinguished from each other.



- Challenges: As heterogeneous systems usually have different data formats, much custom development is required for the code to guarantee data consistency.
- Solution: You can use the visual data transform feature of iPaaS to reduce the integration development and late Ops costs.