SAP Cloud SDK

GitHub Repositories

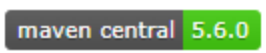
- SAP Cloud SDK Learning Journey GitHub repo
  - https://github.com/SAP-samples/cloud-sdk-learning-journey.git
- Business Partner API (Mock Server) GitHub repo
  - https://github.com/SAP/cloud-s4-sdk-book.git
- Sample SAP Cloud SDK application GitHub repo
  - https://github.com/miltonchandradas/cloudsdk.git

SAP Cloud SDK

The one-stop shop for developing and extending SAP applications in the cloud.

Getting Started

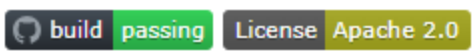
SAP Cloud SDK for Java



The SAP Cloud SDK for Java allows you to develop, extend, and communicate with SAP solutions SAP S/4HANA Cloud, SAP SuccessFactors, and many others.

Get started with the SDK for Java.

SAP Cloud SDK for JavaScript



The SAP Cloud SDK for JavaScript helps you build cloud-based apps and extensions to SAP solutions using the power and flexibility of Node.js and its ecosystem.

Get started with the SDK for JavaScript

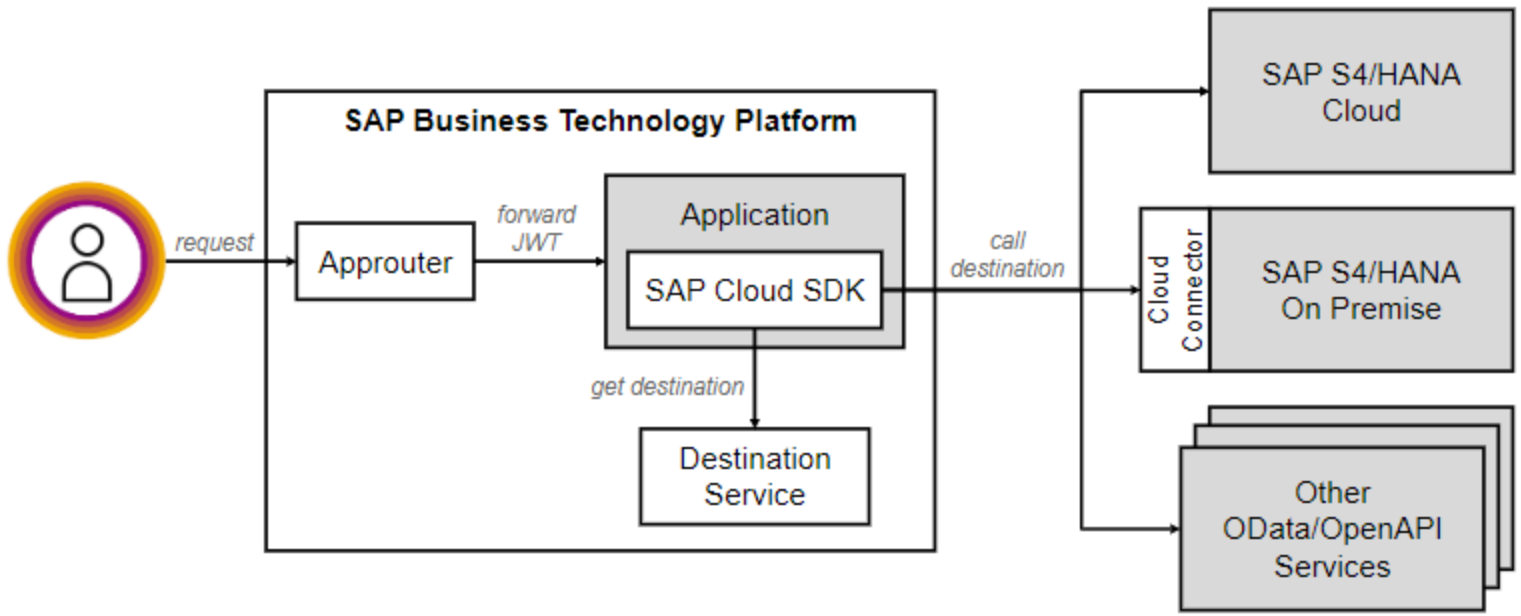
About the SAP Cloud SDK

- Set of libraries and tools to build cloud-native applications on SAP BTP
- Abstracts complexities of SAP BTP – Focus on business logic

- Capabilities
  - Connectivity abstractions for SAP BTP – handling destinations, proxies, authentication
  - Common resiliency patterns such as retry, circuit-breaker etc.
  - Multi-tenancy support
  - Typed clients for OData and OpenAPI services

Products using the SAP Cloud SDK

- SAP Cloud Application Lifecycle Management
- SAP Cloud Application Programming Model
  - Recommended for building APIs (OData, REST, Messaging)
  - Consuming and emitting enterprise messages
  - Depending on use case – you can use SAP Cloud SDK and CAP individually or in combination
- SAP Cloud for Real Estate
- SAP Portfolio and Project Management
- SAP Business Technology Platform Workflow


SAP Business Technology Platform
request
Approuter
forward JWT
Application
SAP Cloud SDK
get destination
Destination Service
call destination
SAP S4/HANA Cloud
Cloud Connector
SAP S4/HANA On Premise
Other OData/OpenAPI Services

SAP BTP Connectivity

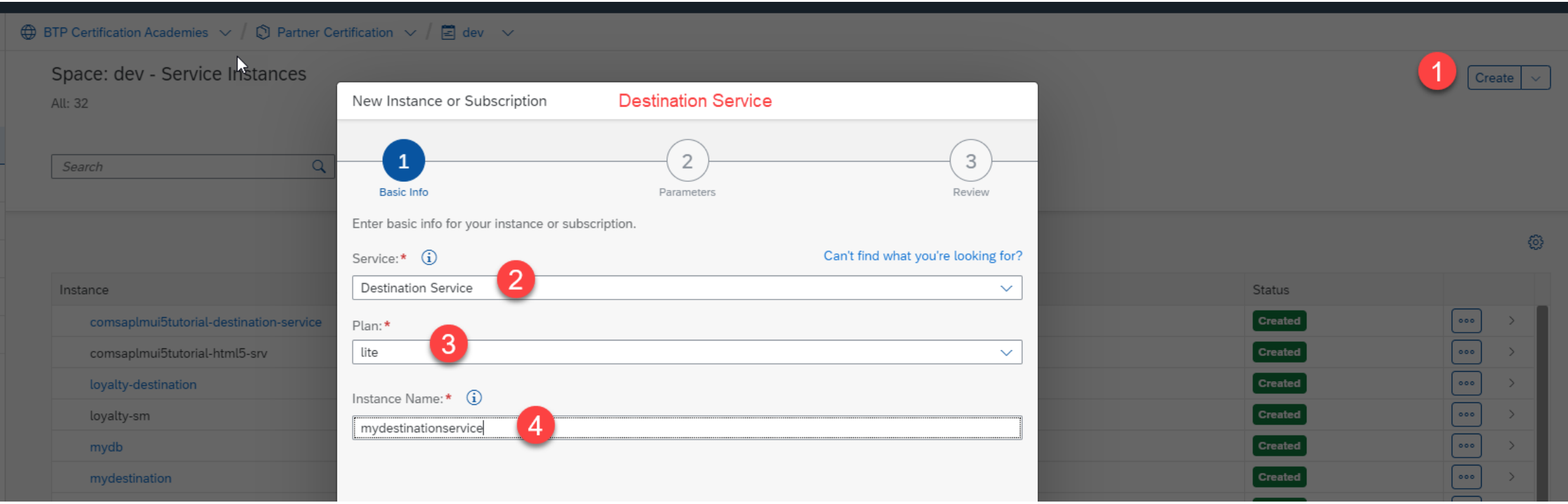
- Cloud Foundry environment
  - Connectivity Service – Connectivity proxy to access on-premise resource
  - Destination Service – Retrieve and store technical info about target resource

I want to…	Services required
Connect to publicly available Services	Destination Service
Connect to On-Premise Services	Destination Service, Connectivity Service

Destinations

- In SAP BTP, destinations are managed separately from application code
- Destinations can be retrieved through Destination Service at runtime

- Benefits of separating destinations and application code
  - Securely store authentication information
  - Update target resource technical info without modifying code
  - Different environments may want to configure different systems
  - Multiple applications might want to access the same system
  - Execute arbitrary requests, OData requests, send emails etc.

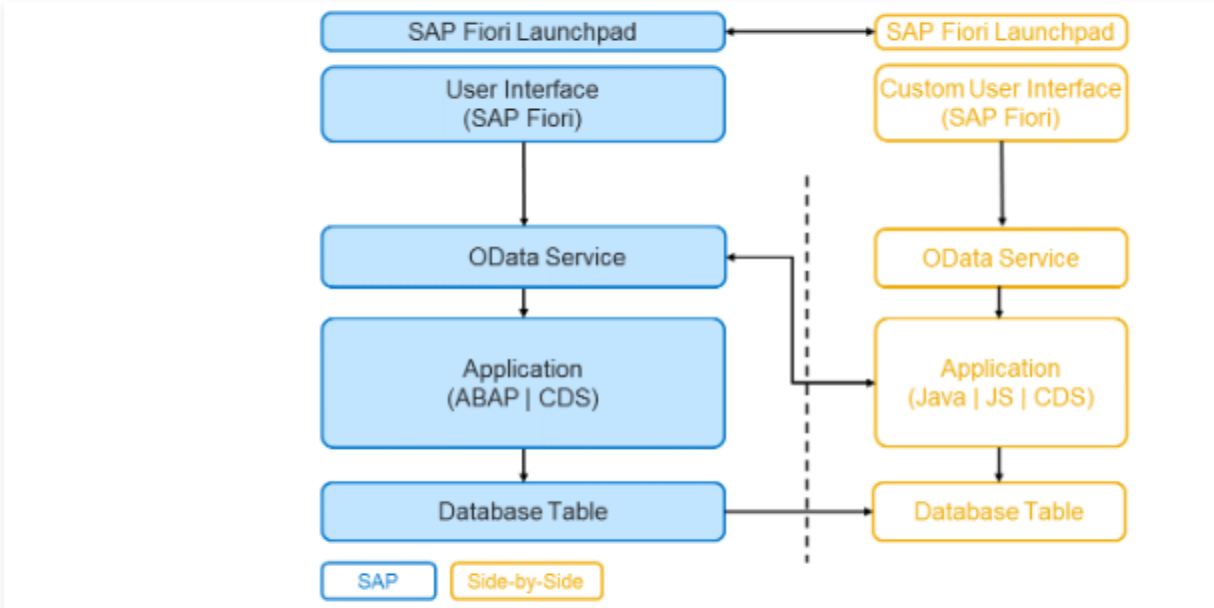

BTP Certification Academies
Partner Certification
dev
Space: dev - Service Instances
All: 32
Search
1
Create
New Instance or Subscription
Destination Service
1
Basic Info
2
Parameters
3
Review
Enter basic info for your instance or subscription.
Service: *
Can't find what you're looking for?
Destination Service
2
Plan: *
lite
3
Instance Name: *
mydestinationservice
4
Instance
Status
comsaplmui5tutorial-destination-service
comsaplmui5tutorial-html5-srv
loyalty-destination
loyalty-sm
mydb
mydestination
Created


Name: *
northwind
Type:
HTTP
Description:
URL: *
https://services.odata.org/V2/(S(wd5zwkiarvfvgybus0l5a3x...
Proxy Type:
Internet
Authentication:
NoAuthentication
Additional Properties
HTML5.Dynam...
true
WebIDEEnabled
true
Use default JDK truststore
Destination

The software architecture of a side-by-side extension

A side-by-side extension is a custom application that runs alongside SAP S/4HANA Cloud.

It can be integrated with SAP S/4HANA Cloud at different levels, such as the user interface, application logic, or data level.

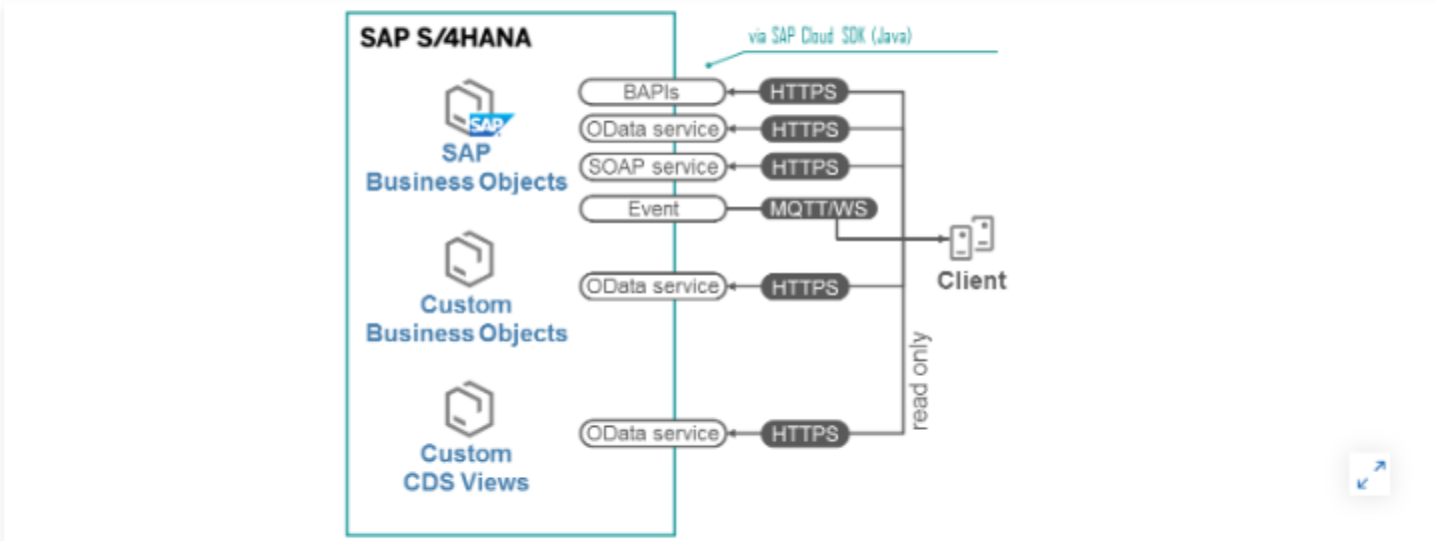


Side-by-side extensions have the following key features:

- They are decoupled from the core application, and integration happens through APIs, events, and replication services.
- Their lifecycle is independent from the SAP application lifecycle.
- They run on separate application stacks.

Introducing SAP APIs

SAP provides a set of APIs that allow side-by-side extensions to interact with SAP S/4HANA Cloud. These APIs allow developers to read SAP data and perform actions that users can perform through the UI, such as getting a list of business partners or creating a new business partner address.



SAP APIs are lifecycle-stable, meaning they will continue to work the same way even when SAP S/4HANA Cloud is upgraded.

The APIs are also designed based on open protocols (OData, REST, and SOAP), to support the widest possible range of client libraries.

The virtual data model

Due to the way the data is structured in SAP S/4HANA, it is difficult to query it manually. Consider a scenario where you need modeled data output related to the *Sales Order* transaction. Such cases would involve joins and queries to multiple database structures to get the desired results. Therefore, we need a tool that can ease the way we query the data and provide the desired results in a structured format. This is where the SAP S/4HANA virtual data model (VDM) comes into the picture.

SAP S/4HANA introduced a VDM that aims to remove this complexity and provide data in a semantically meaningful and easy to consume way. SAP S/4HANA shields the existing primary ERP structure/tables with an understandable, comprehensive logical data model.

Some of the features of the VDM include the following:

- User-known business terminology
- Documenting the relationships between entities
- Enriching entities with business semantics
- Metadata-driven creation of smart UIs

The VDM is built using the technology of SAP ABAP Core Data Services.

When to use SAP Cloud SDK ?

Scenario 1

For the most part, when building applications using the SAP CAP Model, you will not even be aware that it is using the SAP Cloud SDK behind the covers. In fact, the recommendation in SAP CAP Model when connecting to remote services is not to directly leverage the SAP Cloud SDK. But there are times when you might want to directly leverage the SAP Cloud SDK from within the SAP CAP Model application. For example, if you have a requirement within your CAP application to retrieve the JWT token from the request object, then you can do so with the following code.

Code snippet

Copy code

```
1
2   const { retrieveJwt } = require("@sap-cloud-sdk/core");
3   const jwt = retrieveJwt(req);
```

SAP CAP Model is fully compatible with SAP Cloud SDK. So even though, CAP Model hides the complexity to the developer, you could still directly leverage the SAP Cloud SDK to perform custom logic.

Scenario 2

If you are planning to build your application using MongoDB, then you will not be able to use the CAP Model (since it only supports HANA for production). In this case, you will have to build your application without using the CAP Model. In such scenarios, more than likely you will need to use the SAP Cloud SDK for connecting to remote systems, consuming APIs in a type-safe manner etc. You can also use the SAP Cloud SDK to secure your application, provide multi-tenant support etc.

Of course, this scenario is not restricted to supporting MongoDB - but any scenario where you decide not to use the CAP Model for any reason. In all these scenarios, you will use the SAP Cloud SDK directly to build your extension applications.

Scenario 3

If you already have an existing Java application that was built without using the SAP CAP Model and you want to add more features to it. Here again, you can make use of the SAP Cloud SDK tool set.

Scenario 4

If you need advanced features in your app (currently, not provided by CAP Model) - for example, Resilience. SAP Cloud SDK also allows you to generate the Virtual Data Model (VDM) of custom OData Services as well as standard SAP OData Services. This VDM can then be readily consumed by your application.

What is batch processing ?

- Allows grouping of multiple operations into a single HTTP request payload
- Submitted as a single HTTP POST request to the $batch endpoint of a service

Format of a Batch Request – Request Body

- Ordered series of GET operations (AND / OR)
- ChangeSets

What is a ChangeSet ?

- Unordered group of CUD operations
- Atomic unit of work
- Cannot contain GET requests
- Cannot be nested (ChangeSet cannot contain a ChangeSet)
- Represented as a distinct MIME part
- Separated by boundary marker defined in Content-Type header

Example of Batch Request Body

- GET request
- Change Set A
  - Insert
  - Update
- Change Set B
  - Insert
  - Update
- GET request

Circuit Breakers

Circuit Breaker is a design pattern where the application automatically stops making remote service calls if the remote service call has failed too many times. The CircuitBreaker is implemented via a finite state machine with 3 normal states.

- CLOSED
- OPEN
- HALF_OPEN

When the number of consecutive remote service call failures exceed a configured threshold, the circuit breaker switches to the OPEN state. No more remote service calls are made by the application for the duration of the timeout period. After the timeout expires, the circuit breaker switches to the HALF_OPEN state. Limited remote service calls are made by the application. If these remote service calls succeed, the circuit breaker switches to the CLOSED state and normal operations are resumed.

Bulkheads

Bulkhead pattern is used to limit the number of concurrent requests to a remote service. If the number of concurrent incoming requests exceed the configured threshold, the bulkhead is said to be saturated. In this case, further requests are automatically stopped until existing requests are completed.

Timeouts

If the response time of a remote service call exceeds the configured timeout duration, the remote service call is considered to have failed. Resilience4j allows setting custom timeout durations for every remote service call.