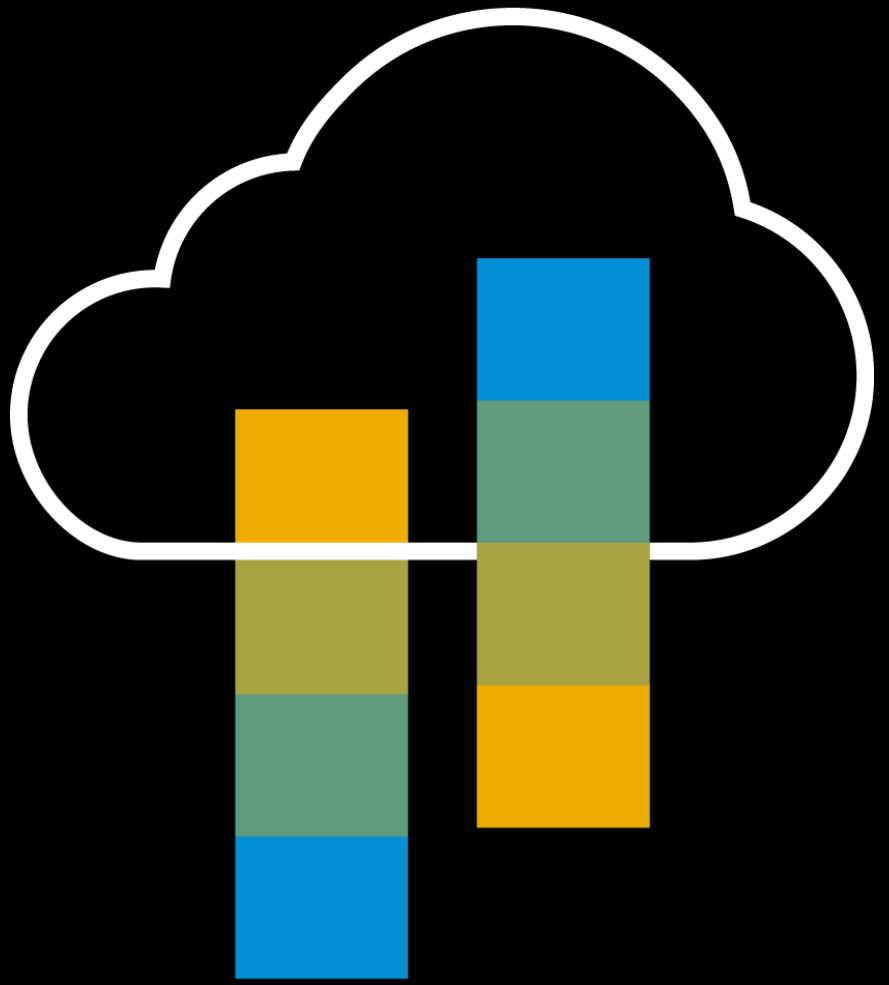
SAP Cloud SDK

JavaScript



PUBLIC

SAP Run Simple

SAP Cloud SDK for Javascript

Similar to the SDK for Java, the SAP Cloud SDK for JavaScript makes it easy and delightful to develop extensions to SAP S/4HANA as applications on SAP Cloud Platform.

The SAP Cloud SDK for JavaScript helps you to integrate SAP S/4HANA into your Cloud-native applications written in TypeScript or JavaScript, running in Node.js.

You can execute those applications locally or deploy them to SAP Cloud Platform, Cloud Foundry.

Capabilities of the SDK for JavaScript

Easy access to SAP S/4HANA Cloud APIs

Straightforward integration of SAP Cloud Platform

TypeScript or JavaScript?

Continuous delivery pipeline

Easy access to SAP S/4HANA Cloud APIs

No boilerplate

No Low-level code

Just business logic

```
import { BusinessPartner } from '@sap/cloud-sdk-vdm-business-partner-service';

BusinessPartner.requestBuilder()
    .getAll()
    .select(BusinessPartner.LAST_NAME)
    .filter(BusinessPartner.FIRST_NAME.equals("John"))
    .execute({destinationName: "S4HANACloud"})
    .then(businessPartners => {
        // process result of type BusinessPartner[]
    }).catch(reason => {
        // handle error
    });
```

Straightforward integration of SAP Cloud Platform

Out-of-the-box integration with the destination service on SAP Cloud Platform for managing the target system, or *destination*, and authentication credentials.

```
import { BusinessPartner } from '@sap/cloud-sdk-vdm-business-partner-service';

BusinessPartner.requestBuilder()
    .getAll()
    .select(BusinessPartner.LAST_NAME)
    .filter(BusinessPartner.FIRST_NAME.equals("John"))
    .execute({destinationName: "S4HANACloud"})
    .then(businessPartners => {
        // process result of type BusinessPartner[]
    }).catch(reason => {
        // handle error
    });
```

It’s easy to substitute the destination service with simple environment variables when running locally or during tests

TypeScript or JavaScript?

This example uses TypeScript, a typed superset of JavaScript.

The SAP Cloud SDK for JavaScript has been implemented in TypeScript, and can be used in JavaScript and TypeScript project alike

```
import { BusinessPartner } from '@sap/cloud-sdk-vdm-business-partner-service';

BusinessPartner.requestBuilder()
    .getAll()
    .select(BusinessPartner.LAST_NAME)
    .filter(BusinessPartner.FIRST_NAME.equals("John"))
    .execute({destinationName: "S4HANACloud"})
    .then(businessPartners => {
        // process result of type BusinessPartner[]
    }).catch(reason => {
        // handle error
    });
```

Continuous delivery pipeline

Include everything needed to run the continuous delivery pipeline on a Jenkins server

How to Access the JavaScript Libraries

The JavaScript libraries of the SAP Cloud SDK are freely accessible from SAP's npm registry

```
npm config set "@sap:registry" "https://npm.sap.com"
```

```
npm install @sap/cloud-sdk-vdm-business-partner-service
```

Benefits

- Scalability
  - Cloud native architecture ensures scalability and high availability
- Agility
  - Easier and faster customizations
  - Faster deployments (canary, A/B deployments)
  - Transparent application connectivity
  - Reusable Services and SDKs to improve developer productivity
- Upgradability
  - Avoid customizing enterprise applications
  - Decoupling by using well defined APIs and business events

Without Using SDK

```
const httpRequest = require('request-promise-native');

module.exports = { main: async function (event, context) {

    // get oauth token
    var oauthToken = await httpRequest({
        url: 'https://ec1808-sapcx-thkeynote.demo.hybris.com//sap/opu/oauth', method: 'POST', json: true, form: {
            'grant_type': 'client_credentials',
            'client_id': 'user',
            'client_secret': 'password'
        }
    });

     var url = 'https://ec1808-sapcx-thkeynote.demo.hybris.com/sap/opu/odata/sap/API_BUSINESS_PARTNER/A_BusinessPartner?';
     var query = `$select=BusinessPartner,FirstName&format=json&$top=5&$expand=to_BusinessPartnerAddress&$filter=BusinessPartnerCategory eq '2'`

     // get oauth token
     var resp = await httpRequest({'url', url  + query,
                                   json: true,
                                   auth: { 'bearer': oauthToken
                                  }});
     if (resp.d.results.length > 0)
     {
         var businessPartnerResults = [];
         for (var i in resp.d.results)
         {
             var address = businessPartnerResults[i].to_BusinessPartnerAddress[i];
             var bp = {
                 "businesspartner" : businessPartnerResults[i].BusinessPartner,
                 "firstname" : businessPartnerResults[i].FirstName,
                 "address" : address.AddressID + " " + address.CityName
             }
             businessPartnerResults.push(bp);
         }
}
}}
```

Authorization handling

Results contain OData metadata so conversion might be necessary


```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://elt-demo.apimanagement.eu2.hana.ondemand.com/
            /sap/API_BUSINESS_PARTNER/A_BusinessPartner('9001')",
          "uri": "https://elt-demo.apimanagement.eu2.hana.ondemand.com
            /sap/API_BUSINESS_PARTNER/A_BusinessPartner('9001')",
          "type": "API_BUSINESS_PARTNER.A_BusinessPartnerType"
        },
        "BusinessPartner": "9001",
        "FirstName": ""
      }
    ]
  }
}
```

SAP Cloud SDK, Cloud Foundry

```
const BusinessPartner = require('s4sdk-vdm/business-partner-service').BusinessPartner;

module.exports = { main: async function (event, context) {

    var resp = await BusinessPartner.requestBuilder().getAll().select(BusinessPartner.BUSINESS_PARTNER, BusinessPartner.FIRST_NAME)
                                                              .filter(BusinessParner.BUSINESS_PARTNER_CATEGORY.equals('2'))
                                                              .expand(BusinessPartner.TO_BUSINESS_PARTNER_ADDRESS)
                                                              .top(5).execute();
    return resp;
}}
```

- Generated Virtual Data Model (VDM) provides type safety

- SDK handles authentication using CF Destination Service
- URL and authorization managed in a separate configuration file (yaml)

Note: Cloud Platform Cockpit configuration coming soon in GA JS version (already supported in Java version)

```
---
applications:

- name: example-app
  memory: 256M
  random-route: true
  buildpacks:
    - nodejs_buildpack
  command: npm start
  env:
    destinations: >
      [
        {
          "name": "ErpQueryEndpoint",
          "url": "https://mys4hana.com",
          "username": "<USERNAME>",
          "password": "<PASSWORD>"
        }
```

Questions?

Thank you.