

Unit 4 – Developing with SAP Extension Suite

Certification: [C_CPE_13](#)

Unit 4 – External Services

Agenda

- What have we learned so far
- SAP API Business Hub
- Add EDMX file to project
- Consume external service in UI application
- SAP BTP Connectivity
- SAP Cloud Connector
- Summary Points

Steps involved

1. Initialize full-stack project – Completed (Unit 1, 2)
2. Create the tables – Data Modeling – Completed (Unit 1, 2)
3. Generic handlers – Out-of-the-box CRUD functionality – Completed (Unit 1, 2)
4. Basic UI – Completed (Unit 3)
5. List Report layout – Completed (Unit 3)
6. Custom event handling – Business logic – Completed (Unit 3)
7. Support for external API
8. Connecting to Sandbox

Steps involved

9. Consume external service in UI

10. Manual deployment to CF using manifest.yml

11. Manual deployment to CF using mta.yml

12. Security – Restrictions and Roles

13. Security – Authorization and Trust Management

14. Creating an AppRouter

15. Adding AppRouter to mat

16. CI / CD Pipeline

Step 7 – Support for External API

Copy metadata document of external OData Service to srv folder
`cds import <Name of edmx file>`

`git checkout 6_external_api`

New Files:

`srv/external/<Name of edmx file>.csn`
`srv/external/<Name of edmx file>.edmx`

Modified Files:

`db/schema.cds`
`srv/risk-service.cds`
`package.json`

Step 7 – Support for External API


SAP API Business Hub

<https://api.sap.com>

SAP S/4HANA Cloud – Business Partner

https://api.sap.com/api/API_BUSINESS_PARTNER/overview

Documentation

 Business Documentation

API Resources

Attributes

API Specification

Configuration Details

Extensibility

Name	Value
JSON	↓
YAML	↓
EDMX	↓

```
// using an external service from S/4
using { API_BUSINESS_PARTNER as external } from '../srv/external/API_BUSINESS_PARTNER.csn';

entity BusinessPartners as projection on external.A_BusinessPartner {
    key BusinessPartner,
    LastName,
    FirstName
}
```

Point to .csn file...

No need to provide data type
Information read from .csn file
Name should match exactly

@readonly

```
entity BusinessPartners as projection on rm.BusinessPartners;
```

```
</EntityType>
▼<EntityType Name="BusinessPartners">
  ▼<Key>
    <PropertyRef Name="BusinessPartner"/>
  </Key>
  <Property Name="BusinessPartner" Type="Edm.String" MaxLength="10" Nullable="false"/>
  <Property Name="LastName" Type="Edm.String" MaxLength="40"/>
  <Property Name="FirstName" Type="Edm.String" MaxLength="40"/>
</EntityType>
```


Step 8 – Connecting to Sandbox API

Get the API key from SAP Business Hub

Add API key to .env file

```
git checkout 7_connecting_to_sandbox
```

Requirement:

- Send API key along with request headers
- Only get entries where there is FirstName and LastName

Modified Files:

.env

srv/risk-service.js

package.json

```

// connect to remote service
const BPsrv = await cds.connect.to("API_BUSINESS_PARTNER");

/**
 * Event-handler for read-events on the BusinessPartners entity.
 * Each request to the API Business Hub requires the apikey in the header.
 */
this.on("READ", BusinessPartners, async (req) => { Generic handler cannot be used for remote service
  // The API Sandbox returns alot of business partners with empty names.
  // We don't want them in our application
  req.query.where("LastName <> '' and FirstName <> '' ");

  return await BPsrv.transaction(req).send({
    query: req.query,
    headers: {
      apikey: process.env.apikey,
    },
  });
});

```

Core Data Services Query Language (CQL)
Only entries with both FirstName and LastName

Pass API key in header

```

// connect to remote service
const BPsrv = await cds.connect.to("API_BUSINESS_PARTNER");

/**
 * Event-handler for read-events on the BusinessPartners entity.
 * Each request to the API Business Hub requires the apikey in the header.
 */
this.on("READ", BusinessPartners, async (req) => { Generic handler cannot be used for remote service
  // The API Sandbox returns alot of business partners with empty names.
  // We don't want them in our application
  req.query.where("LastName <> '' and FirstName <> '' ");

  return await BPsrv.transaction(req).send({
    query: req.query,
    headers: {
      apikey: process.env.apikey,
    },
  });
});

```

Core Data Services Query Language (CQL)
Only entries with both FirstName and LastName

Pass API key in header

Standard ▾

Search 

Editing Status: All ▾

SelectionFields








Priority: 

HeaderInfo

Risks (3)

LineItems

 Title	Mitigation	Owner	Business Partner	New Line Item	Priority	Impact	
<input type="checkbox"/> CFR non-compliance	SLA violation: authorize account manager to offer service credits for recent delivery issues	Fred Fish	Abs		 3	 10,000	>
<input type="checkbox"/> SLA violation with possible termination cause	SLA violation: review third party contractors to ease service delivery challenges; trigger budget review	George Gung	Menon		 2	 90,000	>
<input type="checkbox"/> Shipment violating export control	Embargo violation: investigate source of shipment request, revoke authorization	Herbert Hunter	Jackie		 1	 200,000	>



Risk ▾

CFR non-compliance

Recent restructuring might violate CFR code 71

HeaderInfo - Title

HeaderInfo - Description

Facet

Main

Mitigation:

SLA violation: authorize account manager to offer service credits for recent delivery issues

Owner:

Fred Fish

FieldGroup with multiple Data Fields

Business Partner:

Abs

New Data Field

Priority:

⚠ 3

Impact:

⚠ 10,000

Step 9 – Consume external service in UI

```
git checkout 8_consume_external_service_in_UI (Use tab for branch name)
```

Modified Files

- `app/annotations.cds`
- `app/common.cds`
- `db/data/riskmanagement-Risks.csv`
- `db/schema.cds`
- `srv/risk-service.js`

Run application command

- `cds watch`

srv/risk-service.js

UI makes the following OData call...

GET

[http://localhost:4004/service/risk/Risks?\\$expand=bp,miti\(\\$select=ID,IsActiveEntity,descr\)](http://localhost:4004/service/risk/Risks?$expand=bp,miti($select=ID,IsActiveEntity,descr))

GET all Risks

- GET corresponding bp details for each Risk
- GET corresponding miti details for each Risk

Risks

?\$expand=**bp,miti**(\$select=ID,IsActiveEntity,descr)

Generic handler cannot expand bp → So we need to have custom handler to expand bp

srv/risk-service.js

Aim:

Read data from external data source along with Risks

Custom event handler: **On**, **READ**, **Risks**

Steps:

1. Remove expand **bp** clause from request query
2. Call the generic READ handler
3. Call the external data source for BP info

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

Destination Service – CF

Works for any publicly available OData Service

<https://services.odata.org/v2/northwind/northwind.svc/>

Destination Configuration



Name: *	northwind_api
Type:	HTTP
Description:	
URL: *	https://services.odata.org
Proxy Type:	Internet
Authentication:	NoAuthentication

Additional Properties

HTML5.Dyna...	true	
WebIDEEnab...	true	
WebIDESyst...	northwind_api	
WebIDEUsage	odata_gen	

☒ Use default JDK truststore

[Edit](#) [Clone](#) [Export](#) [Delete](#) [Check Connection](#)

Destination Service – CF

Scenarios:

1. Connecting to On-Premise OData Service
2. Retrieve connection details

Destination Configuration

Name: *	sales_api
Type:	HTTP
Description:	
URL: *	http://ics4:9001
Proxy Type:	OnPremise
Authentication:	BasicAuthentication
Location ID:	
User: *	i825339
Password:	*****

Additional Properties

HTML5.Dyna...	true	
sap-client	100	
WebIDEEnab...	true	
WebIDESyst...	sales_api	
WebIDEUsage	odata_gen	

Edit

Clone

Export

Delete

Check Connection

Connectivity Service – CF

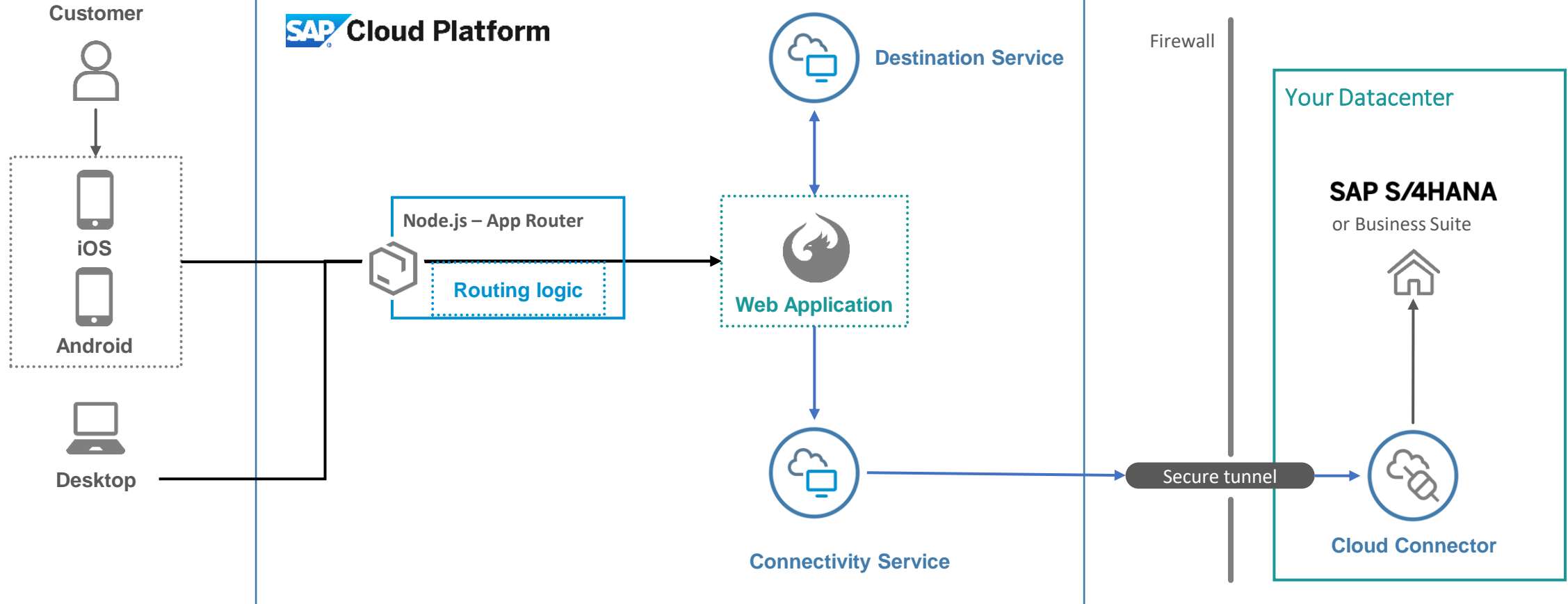
Set up On-Premise communication via HTTP or RFC for your cloud application



```
{
  "tenantmode": "c
  "clientid": "sb-clc
  "token_service_c
  "token_service_u
  "xsappname": "c
  "onpremise_prox
  "onpremise_sock
  "clientsecret": "X
  "onpremise_prox
  "url": "https://ic-c
  "onpremise_prox
  "uaadomain": "ai
  "onpremise_prox
  "verificationkey":
MIICljANBgkqhkiG9w
d8rKfYd6olGWigFd+3
lWTxe+FyNklvyZvoLrZ
MfeVf0P2th5C9MggY
  "identityzone": "i
  "tenantid": "5349
  "onpremise_prox
}
```

GKMnw7cvCwN
lhKIC7WLwCEJ
JwhTN1HvyXrs
) PUBLIC KEY--

INTERNET



SAP Cloud Connector

- Serves as a link between SAP BTP applications and on-premise systems.
- Runs as on-premise agent in a secured network.
- Provides fine-grained control over the connectivity.
- Enterprise Features
 - Recovers broken connections automatically.
 - Provides audit logging of inbound traffic and configuration changes.
 - Can be run in a high-availability setup.

Key Summary Points – Unit 4

Enterprise features of SAP Cloud Connector

- High Availability Setup – Main and Shadow instance
- Secure trace data – Secure sensitive network trace data
- Monitoring – State and activities of Cloud Connector
- Alerting – Send email alerts
- Audit Logging – View and manage audit log

Key Summary Points – Unit 4

- Cloud Foundry environment BTP Connectivity
 - 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

Key Summary Points – Unit 4

Enterprise features of SAP Cloud Connector

- High Availability Setup – Main and Shadow instance
- Secure trace data – Secure sensitive network trace data
- Monitoring – State and activities of Cloud Connector
- Alerting – Send email alerts
- Audit Logging – View and manage audit log

SAP Cloud Connector component is **required** to **connect SAP BTP to On-Premise system**

You access the SAP Cloud Connector at the **Subaccount** level

Key Summary Points – Unit 4

Question 7

Choose the correct answer(s).

What are some of the capabilities of the SAP Cloud Connector?

- ✓ ☒ Serves as a link between SAP BTP applications and on-premise systems.
- ✓ ☒ Runs as on-premise agent in a secured network.
- ☐ Manages dev spaces.
- ☐ Provides access to the SAP Cloud Identity Services.
- ✓ ☒ Provides fine-grained control over the connectivity.

Submit

Key Summary Points – Unit 4

Question 2

Choose the correct answer.

What does the .env file provide?

- ☒ Values into the runtime environment of a CAP service
- ☐ Values for your version-management-system

Submit

🔍 .env

×




1

```
apikey=JI
```

Don't check this file into GitHub
Contains sensitive info

Key Summary Points – Unit 4


 API Business Hub

ExploreResourcesDiscover IntegrationsPartner with Us

New SAP API Business Hub

Login

/ SAP S/4HANA Cloud

 Business Partner (A2X)

Create, read, update or delete the master data for business partner, supplier or customer using this synchronous inbound service

API key must be sent along with request headers

Show API Key

OverviewAPI ReferenceSchema ViewSAP Cloud SDKTry Out

Introduction

This service enables you to create, display, update, and delete data related to Business Partner, Supplier, and Customer with the data provided in a payload, in an API call. This service also supports create deep entity operation and batch processing.

[View the API Reference](#)[Check Schema View](#)[Try Out](#)

STATUS	TYPE
ACTIVE	ODATA
LAST MODIFIED	VERSION
30 Mar 2022	1

Key Summary Points – Unit 4

Question 6

Choose the correct answer.

What encryption type does the SAP Cloud Connector use?

- ☐ SSL
- ☐ SSH
- ☒ TLS
- ☐ XMPP

Submit