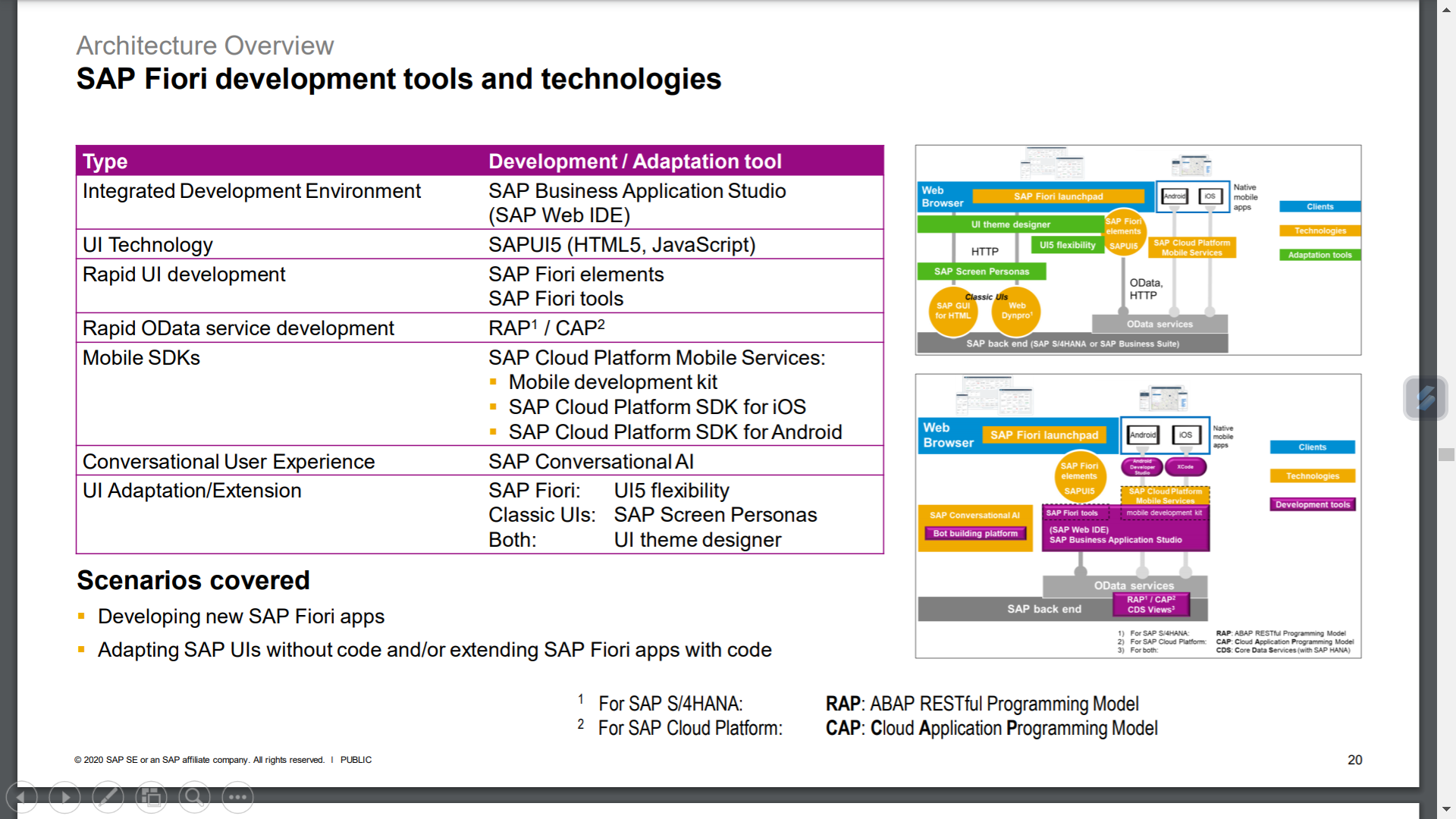
# Cloud Application Programming Using JavaScript



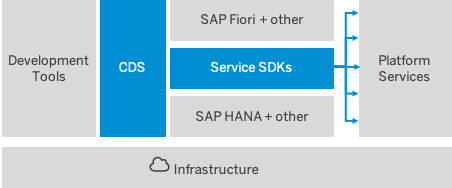
BTP / S4 Hana -2020 > Above.

New ways to build.

What is CAP?

CAP is a framework of languages, libraries, and tools for building enterprise-grade services and applications.

The CAP framework features a mix of proven and broadly adopted open-source and SAP technologies



CAP mainly adds:

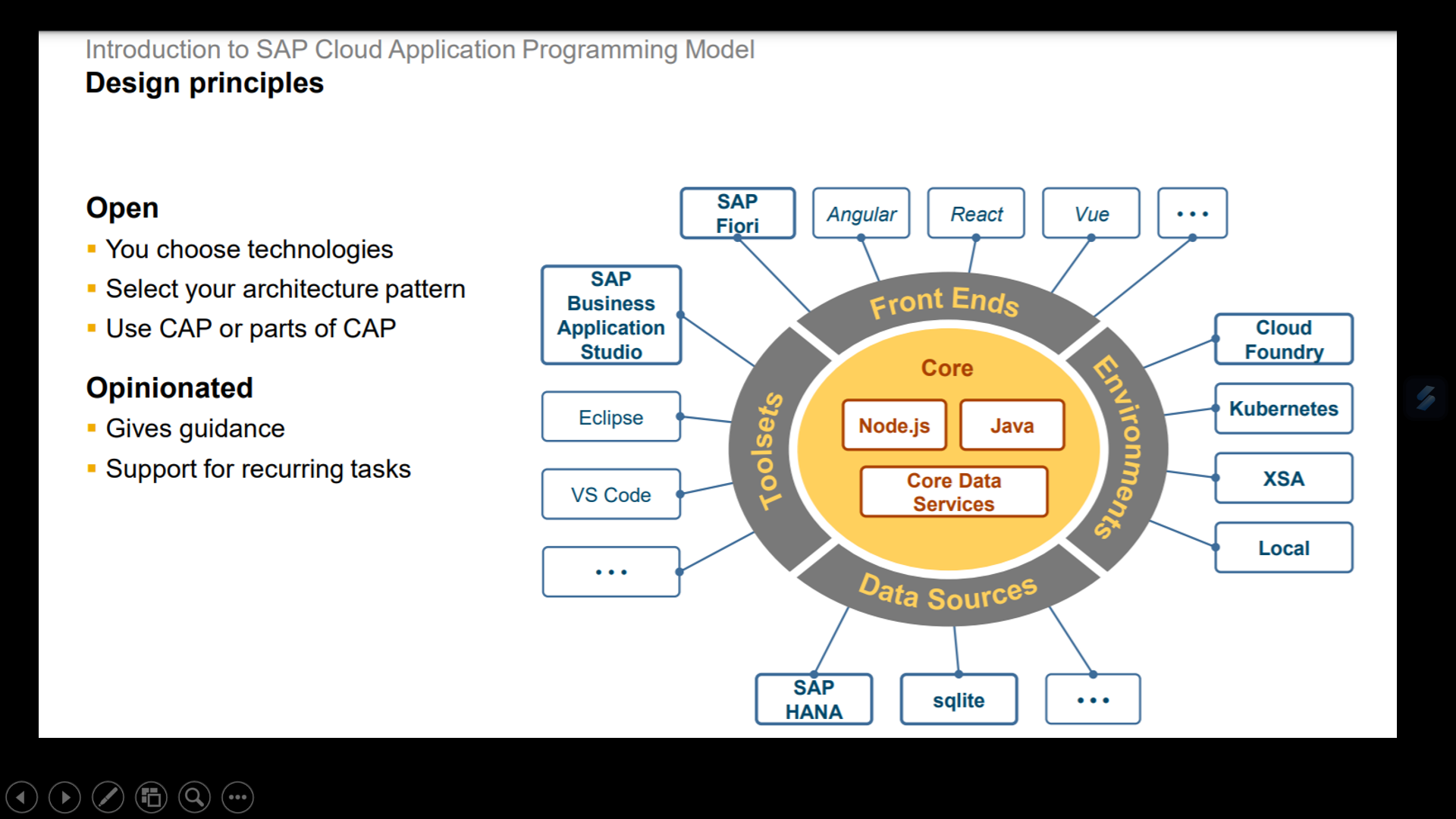
* Core Data Services (CDS) as our universal modeling language for both domain models and service definitions.
* Service SDK and runtime for Node.js and Java, offering libraries to implement and consume services as well as generic provider implementations serving many requests automatically.

ABAP – CDS generation / SQL lite/ Java

Consumers – Open UI5,

Producer – Odata – SAP Gateway (segw/cds views)

SAP CAP – (cds views – node.js/java)



Getting Started –

Beginner - <https://developers.sap.com/tutorials/btp-app-introduction.html>

Beginner - <https://developers.sap.com/mission.cp-starter-extensions-cap.html>

Intermediate- <https://learning.sap.com/learning-journey/build-side-by-side-extensions-on-sap-btp>

Advanced - <https://learning.sap.com/learning-journey/develop-full-stack-applications-using-productivity-tools-in-sap-business-application-studio>

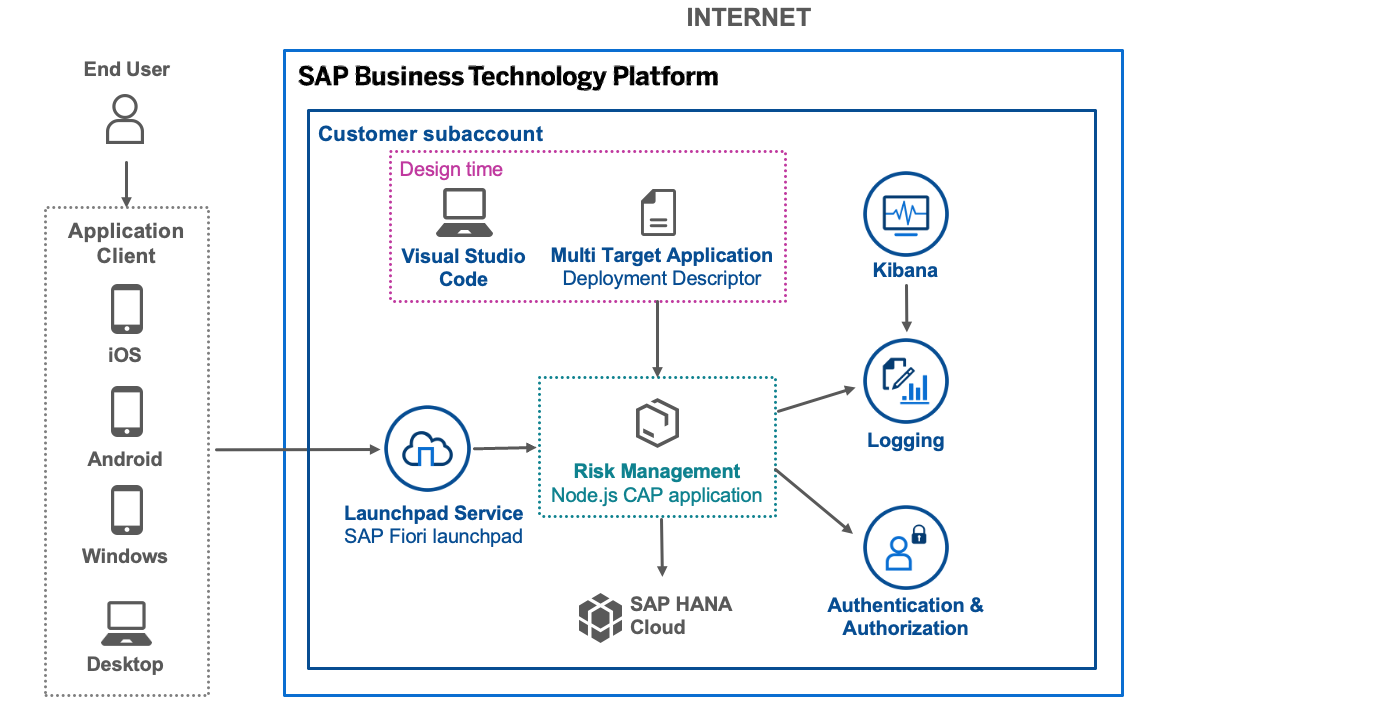
**Sample CAP Odata service**- git clone https://github.com/SAP-samples/cloud-cap-risk-management

cd cloud-cap-risk-management.

Tools required –

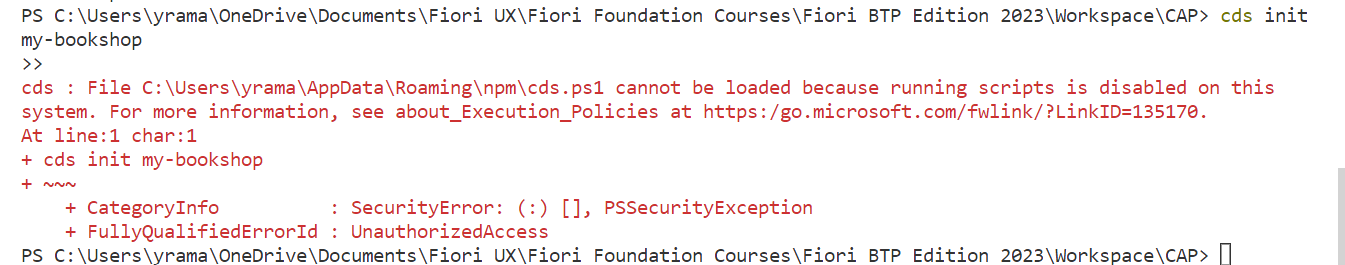
1. Node.js
2. Visual Studio Code
3. SQL Lite -https://sqlite.org/download.html
4. Rest Client
5. BTP Trial account – US east region
6. **MBT tool – (optional for build activities) [use** npm install -g mbt]
7. MultiApps CF CLI Plugin [ [download](https://github.com/cloudfoundry/multiapps-cli-plugin/blob/master/README.md) ] [use download path and install using cf install-plugin <path-to-the-plugin> -f] hint: Right click and use copy path option from windows

Basic Architecture –



Trouble shooting –

Error 1 :



Solution: Goto powershell as administrator and execute below command .

**Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Unrestricted**

https://stackoverflow.com/questions/41117421/ps1-cannot-be-loaded-because-running-scripts-is-disabled-on-this-system

Error 2 – Unable top run cds watch –> reopen the folder to reflect changes.

Getting Started with CAP project

**Keep it handy -**

[https://developers.sap.com/tutorials/cp-apm-nodejs-create-service.html#d1bd1dc9-182f-4afd-b41f-766a621af482](https://developers.sap.com/tutorials/cp-apm-nodejs-create-service.html" \l "d1bd1dc9-182f-4afd-b41f-766a621af482)

1 . cds init my-bookshop ( create project structure)

2. cds watch ( metadata) - refer error 2 . If does not load empty metadata.  
3. Service – odata service - entity

4. Db – all tables

Below are key steps -

* Use db folder for defining tables and test payload.
* Use service folder for defining odata entities .

Association test query - <http://localhost:4004/catalog/Authors?$expand=books>

1. Install sql lite locally (for BAS install – extension SQLite Viewer)

npm i sqlite3 -D

To deploy database schema to sql lite

cds deploy --to sqlite:db/my-bookshop.db

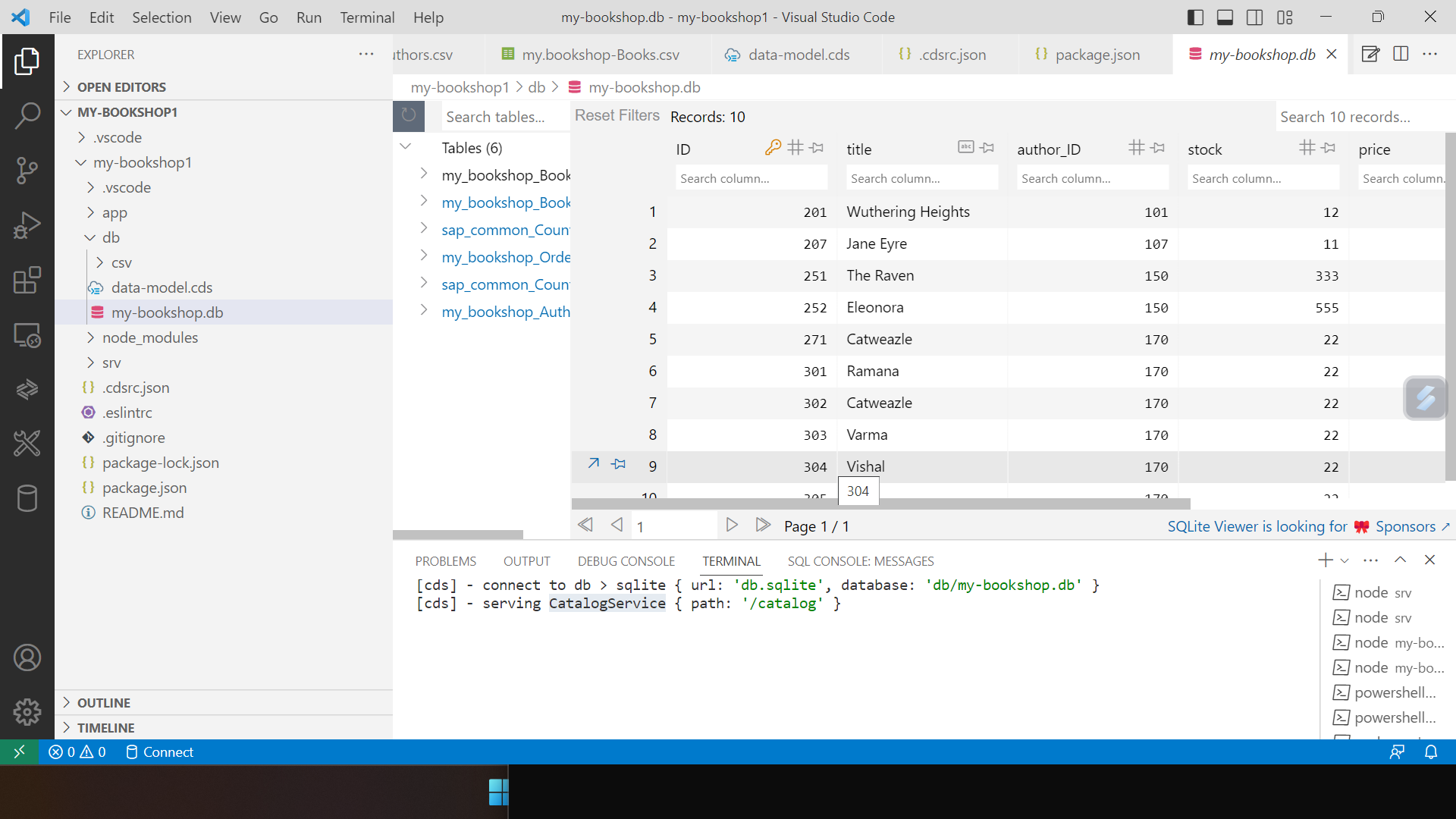
For additional deployment – cds deploy.

Cds watch – ctrl+ C to stop local service odata.

Sqlite command a) deploy - cds deploy --to sqlite:db/my-bookshop.db

b) to run from sqllite – (not working) - sqlite3 db/my-bookshop.db -cmd .dump

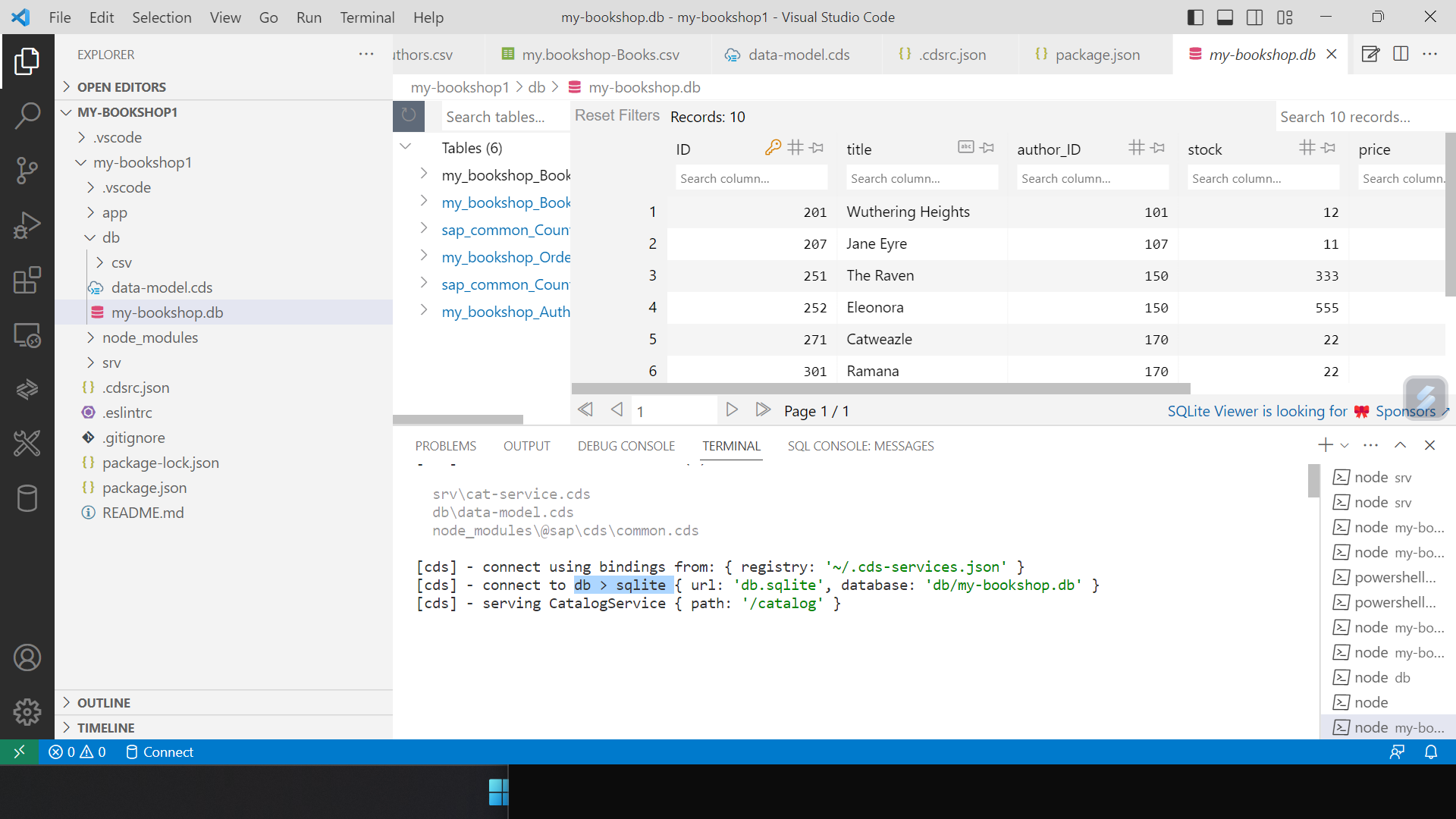
or double click the .db file to view data.



This ensures data schema is deployed and data is updated in SQL Lite.

Cds watch

It should fetch odata directly from db. Validates data is fetched .



For Writing custom logic in db layer using javascript-

<https://cap.cloud.sap/docs/node.js/events#cds-request>

to, before, after – lifecyle methods . ( sqlite specific syntax along js logic)

Install Rest Client -> From Extensions. This is used for testing our custom logic.

Create a new file – test.http in the root folder.

Double click to test the queries.

Reuse the template for modify the payload.

###

#

# Get Author wit ID 101

#

GET http://localhost:4004/catalog/Authors(201)

###

#

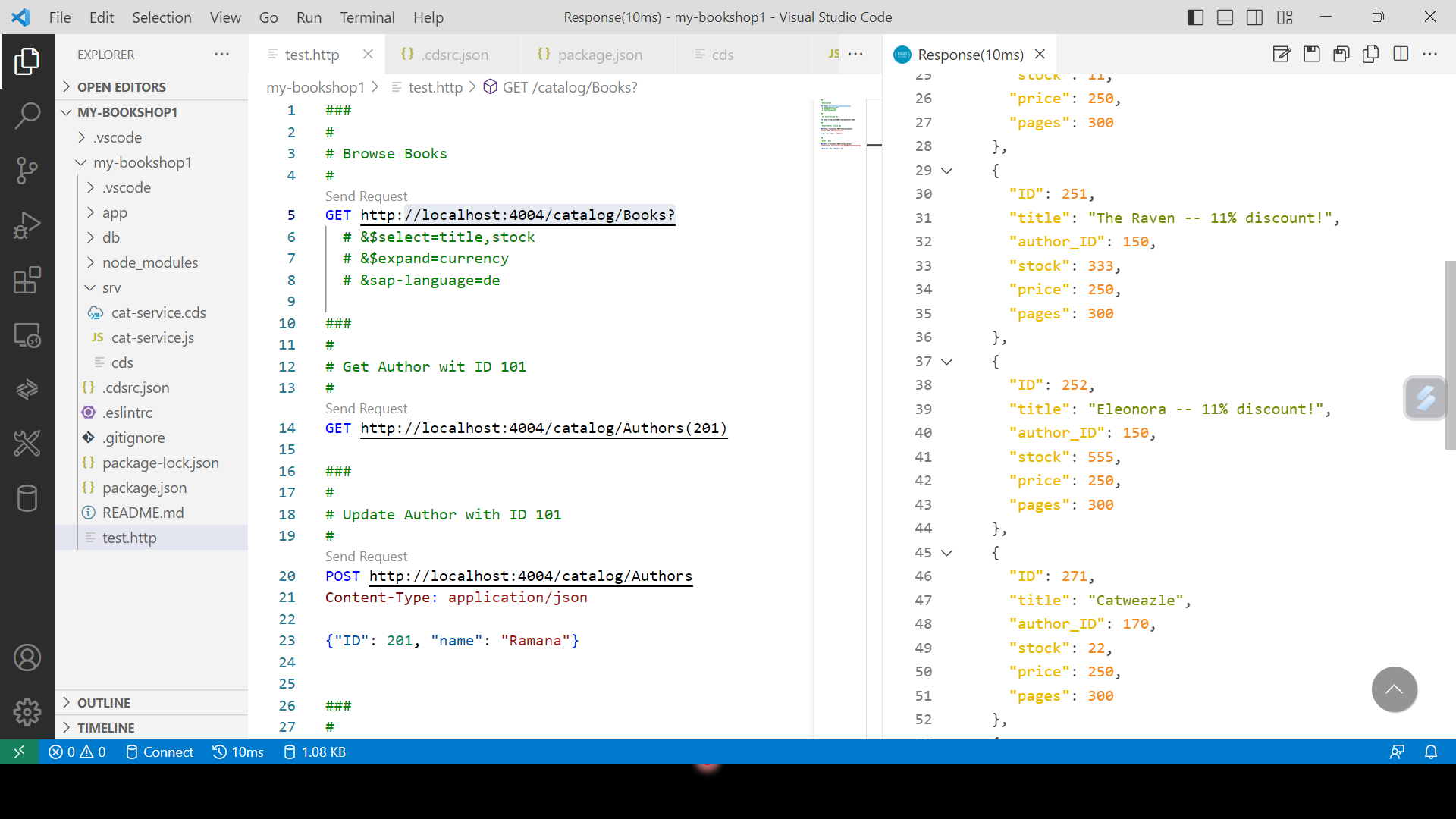
# Update Author with ID 101

#

POST http://localhost:4004/catalog/Authors

Content-Type: application/json

{"ID": 201, "name": "Ramana"}



2nd usecase –

git clone <https://github.com/SAP-samples/cloud-cap-risk-management>

CAP Odata service.

* Next Action –
* UI using CAP generated Service.

STEP 1 - Creating a CDS based oData service.

Common Tasks for all CAP based : <https://developers.sap.com/tutorials/cp-apm-nodejs-create-service.html>

* Cds init <project name>
* Cds watch (running the service) – Choose project and open integrated terminal to run the metadata.
* To stop CDS Watch – Ctrl+C . (Before deploying to SQL , stop the cds watch service)
* Double click .db file to view data from database. On CDS Watch same data can view from odata service.

For CRUD – node.js query are used

* Test.http file to manage all crud operations. Use Generic Handlers as template to perform CRUD operations.

Git copy - <https://github.com/ramana24/CAP-project>

Step 2: Deploying to BTP – Cloudfoundry.

Use BTP trail – with US East.

1. Create additional config files using below command

- cds add hana,mta,xsuaa,approuter --for <cf space>

cds add hana,mta,xsuaa,approuter --for dev

2. Step run in terminal –

cds compile srv/ --to xsuaa > xs-security.json

3. step to run in terminal

cds add approuter --for dev

4. step to run in terminal –

npm update --package-lock-only

5. Final step of Build command – it should create .mtar file in gen folder.

mbt build -t gen --mtar mta.tar

6. Deploy to BTP .

cf deploy gen/mta.tar

-----------------------------------------------------------------------

* Login to BTP

**Cf login (use single BTP trial instance to avoid conflicts )**

**Update xsuaa – file for role access with API endpoint**

[https://api.cf.us10-001.hana.ondemand.com](https://api.cf.us10-001.hana.ondemand.com/)

**MBT tool – (optional for build activities) use** npm install -g mbt for

**mbt build -t gen --mtar mta.tar**

todo : install mbt from npm and intall multicf app plugin

* <https://github.com/cloudfoundry/multiapps-cli-plugin/blob/master/README.md>
* Download Above plugin and copy the path - [right and use copy path option and run below command]
* "C:\SQL\multiapps-plugin.win64.exe"

cf install-plugin "C:\SQL\multiapps-plugin.win64.exe"-f

Missing - .tar build file generation. – skip

Step 3 – Create a Hana Database on BTP

* Go to cloud foundry , dev space and create hana cloud db instance
* UserID: DBADMIN
* Pwd : Fioriuxdb1

TO DO : Import a github project and try to generate a CAP odata service – RISK.

<https://learning.sap.com/learning-journey/build-side-by-side-extensions-on-sap-btp/creating-a-cap-based-service_cc9e93f1-9dda-4f67-9d6b-c6bfefcc0b99>

More SQL – queries – do practice queries for CRUD at db level.

<https://www.w3schools.com/sql/sql_join.asp>

# CAP – Overall Approach

CAP project - BAS

-

SQL lite – Deployment

Node.js Syntax ->

http.test - > insert logic in SQL

CAP CDS -entity Books

Create Call

Fiori App

-

Hana Cloud Instance - BTP

CF -deployment

MBT - Build

* **Odata can be designed with hana standard db tables without ABAP layer.**

UseCase – At the start , we have to ensure.

Fiori App –

Fiori Elements – annotations

Fiori freestyle –

Fiori with Extension Developer – CAP/ RAP

Developer – CAP platform must know things ( JS Developer Fullstack)

Extension Application Developer -

SAP ( OData API) and Non- SAP ( FedEX Api) - > API

* SAMPLES <https://github.com/SAP-samples/cloud-cap-samples>
* Side by Side extension:

<https://learning.sap.com/learning-journey/build-side-by-side-extensions-on-sap-btp>

* In-App Extension : SAP ERP/ SAP S/4 Hana (Fiori Standard / ABAP / API extension)
* Integration suite – sap.Api.hub (segw/SOAP /CDS/ v2/v4) – 3000 API & Non-SAP API - > CPI - Middleware