BTP CAP Monolithic Application Hands-on

Exercises / Solutions

Speakers:   
한정우 (SAP SE)

Table of ContentS

[1. 실습 개요 3](#_Toc131075023)

[2. Prerequisite 3](#_Toc131075024)

[2.1 Monolithic Project 생성 (Basic) 4](#_Toc131075025)

[2.2 Unsubscribe && Undeploy application 10](#_Toc131075026)

[2.3 Wrap-up & Assignment 10](#_Toc131075027)

[3. References 10](#_Toc131075028)

# 실습 개요

SAP BTP는 PaaS환경을 제공하며, 애플리케이션을 개발하여 사용자들에게 서비스를 제공할 수 있습니다. SAP BTP는 어플리케이션 개발을 위해 CAP(Cloud Application Programming Model) framework를 제공하며, 디자인타임에 데이타 모델과 서비스 정의하고, 이를 기반으로 런타임시에 다양한 형태의 data, service, UI layer가 생성됩니다.

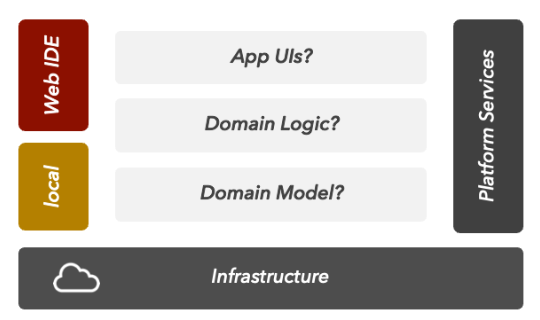
아울러, CAP는 BTP에 있는 서비스들과 seamless integration이 가능해, 어플리케이션 기능들을 다양하고 빠르게 구성할 수 있습니다.

# Prerequisite

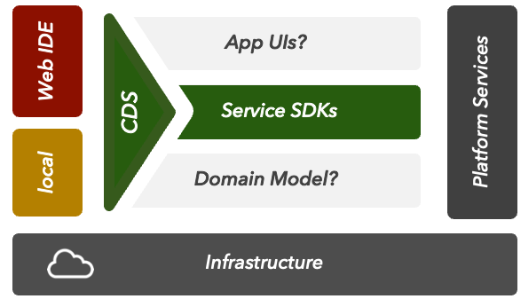
아래 링크들에 대한 이해가 필요합니다.

[Developing with the SAP Cloud Application Programming Model](https://help.sap.com/docs/btp/sap-business-technology-platform/developing-with-sap-cloud-application-programming-model)

<Design-time Model>



<Run-time Model>



.

## Monolithic Project 생성 (Basic)

예제 프로그램을 좀더 쉽게 생성하기 위해 BAS에 추가적인 component를 인스톨 합니다.

| Explanation | Screenshot |
| --- | --- |
| 1. Extension 컴포넌트 설치 |  |
| 1. SAP Partner Engineering BAS Installer 설치 |  |
| 1. Click on BAS install SAP HANA Academy yeoman generators   View >> Command Palette  >> BAS: Install SAP HANA Academy yeoman generators. |  |
| 1. SAP HANA Academy CAP 선택   View / Find Command  Input : wizard |  |
| 1. Project Name : capapp## |  |
| 1. Runtime 선택 |  |
| 1. HANA Cloud 선택 |  |
| 1. external API 선택 🡺 “NO” |  |
| 1. 인증 및 권한 부분 선택 |  |
|  |  |
|  |  |
|  |  |
| 1. Cds watch로 mockup data 확인 | $>cds watch |
| 1. Test.http 수행해보기 |  |
| 1. Data modeling 확인. | context capapp01.db {  entity Sales {  key ID : Integer;  region : String(100);  country : String(100);  org : String(4);  amount : Integer;  comments : String(100);  criticality : Integer;  };  }  @cds.persistence.exists  @cds.persistence.calcview  entity CV\_SALES {  key REGION : String(100);  AMOUNT : Integer;  }  @cds.persistence.exists  @cds.persistence.calcview  entity CV\_SESSION\_INFO {  key ITEM : String(5000);  VALUE : String(5000);  } |
| 1. Service modeling 확인 | using {capapp01.db as db} from '../db/data-model';  using {CV\_SALES, CV\_SESSION\_INFO} from '../db/data-model';  service CatalogService @(path : '/catalog')  @(requires: 'authenticated-user')  {  entity Sales  as select \* from db.Sales  actions {  action boost() returns Sales;  }  ;  @readonly  entity VSales  as select \* from CV\_SALES  ;  @readonly  entity SessionInfo  as select \* from CV\_SESSION\_INFO  ;  function topSales  (amount: Integer)  returns many Sales; |
| 1. Event handling 부분 확인 | this.after('READ', Sales, (each) => {  if (each.amount > 500) {  each.criticality = 3;  if (each.comments === null)  each.comments = '';  else  each.comments += ' ';  each.comments += 'Exceptional!';  debug(each.comments, {"country": each.country, "amount": each.amount});  log.info(each.comments, {"country": each.country, "amount": each.amount});  } else if (each.amount < 150) {  each.criticality = 1;  } else {  each.criticality = 2;  }  });  this.on('boost', Sales, async req => {  try {  const ID = req.params[0];  const tx = cds.tx(req);  await tx.update(Sales)  .with({ amount: { '+=': 250 }, comments: 'Boosted!' })  .where({ ID: { '=': ID } })  ;  debug('Boosted ID:', ID);  const cs = await cds.connect.to('CatalogService');  let results = await cs.read(SELECT.from(Sales, ID));  return results;  } catch (err) {  req.reject(err);  }  });  this.on('topSales', async (req) => {  try {  const tx = cds.tx(req);  const results = await tx.run(`CALL "CAPAPP01\_DB\_SP\_TopSales"(?,?)`, [req.data.amount]);  return results.RESULT;  } catch (err) {  req.reject(err);  }  }); |
| 1. UI Source(Fiori Element 확인) | using CatalogService as catalog from '../../../srv/catalog-service';  annotate catalog.Sales with @(  UI: {  Identification: [ {Value: country} ],  LineItem: [  {Value: ID},  {Value: region},  {Value: country},  {Value: amount, Criticality: criticality},  {$Type: 'UI.DataFieldForAction', Label: '{i18n>boost}', Action: 'CatalogService.boost', Inline: true},  {Value: comments}  ],  HeaderInfo: {  TypeName: '{i18n>country}',  TypeNamePlural: '{i18n>countries}',  Title: {Value: country},  Description: {Value: ID}  }  }  );  annotate catalog.Sales with {  ID @title:'{i18n>ID}' @UI.HiddenFilter;  region @title:'{i18n>region}';  country @title:'{i18n>country}';  amount @title:'{i18n>amount}';  comments @title:'{i18n>comments}';  }; |
| 1. Mta.yaml파일 확인 & Build | $> mbt build  [2023-03-27 10:21:34] INFO generating the metadata...  [2023-03-27 10:21:34] INFO generating the "/home/user/projects/capapp01/.capapp01\_mta\_build\_tmp/META-INF/mtad.yaml" file...  [2023-03-27 10:21:34] INFO generating the MTA archive...  [2023-03-27 10:21:34] INFO the MTA archive generated at: /home/user/projects/capapp01/mta\_archives/capapp01\_0.0.1.mtar  [2023-03-27 10:21:34] INFO cleaning temporary files... |
| 1. CF Login   View >> Command Palette >> CF Login(Key-in) |  |
| 1. Deploy   Mta\_archive 폴더 >> capapp...mtar파일 우클릭 >> Deploy MTA Archive 선택 | $> cf deploy "/home/user/projects/capapp01/mta\_archives/capapp01\_0.0.1.mtar"  …  Started async upload of application "capapp01-app"  Uploading application "capapp01-srv"...  Started async upload of application "capapp01-srv"  Staging application "capapp01-app"...  Staging application "capapp01-srv"...  Uploading application "capapp01-db-deployer"...  Started async upload of application "capapp01-db-deployer"  Staging application "capapp01-db-deployer"...  Application "capapp01-app" staged  Starting application "capapp01-app"...  Application "capapp01-srv" staged  Application "capapp01-app" started and available at "btpkrmta-producer-dev-capapp01-app.cfapps.ap12.hana.ondemand.com"  Starting application "capapp01-srv"...  Application "capapp01-srv" started and available at "btpkrmta-producer-dev-capapp01-srv.cfapps.ap12.hana.ondemand.com"  Application "capapp01-db-deployer" staged  Executing task "deploy" on application "capapp01-db-deployer"...  Skipping deletion of services, because the command line option "--delete-services" is not specified. |
| 1. Fiori Element 기반의 Guided Development 맛보기  * Run “Fiori: Open Guided Development” |  |
|  |  |

## Unsubscribe && Undeploy application

반드시, Consumer tenant에서 Unsubscribe작업 수행 후, Provider application을 삭제 해야 함.

| Explanation | Screenshot |
| --- | --- |
| 1. Unsubscribe | Unsubscribe at the consumer tenant |
| 1. Undeploy application | Undeploy Provider application  $>cf undeploy capapp01 --delete-services --delete-service-keys -f |

## Wrap-up & Assignment

## References

[Developing with the SAP Cloud Application Programming Model](https://help.sap.com/docs/btp/sap-business-technology-platform/developing-with-sap-cloud-application-programming-model)

[Developing Multitenant Applications in the Cloud Foundry Environment](https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/5e8a2b74e4f2442b8257c850ed912f48.html)

[Cloud-Native Lab #4 – Multi-tenant Web Apps in SAP BTP](https://blogs.sap.com/2021/04/22/cloud-native-lab-4-multi-tenant-apps-in-sap-btp/)

[Multitenancy Architecture on SAP Cloud Platform, Cloud Foundry environment](https://blogs.sap.com/2018/09/26/multitenancy-architecture-on-sap-cloud-platform-cloud-foundry-environment/)

[Developing Multitenant Applications on SAP Cloud Platform, Cloud Foundry environment](https://blogs.sap.com/2018/09/17/developing-multitenant-applications-on-sap-cloud-platform-cloud-foundry-environment/)

[Using SaaS Provisioning Service to develop Multitenant application on SAP Cloud Platform, Cloud Foundry Environment](https://blogs.sap.com/2018/10/25/using-saas-registry-to-develop-multitenant-application-on-sap-cloud-platform-cloud-foundry-environment/)