**Plant UML 사용 가이드**

- 목차 -

1. PlantUML 설정
   1. IntelliJ 설치
   2. PlnatUML 다운로드
   3. 프로젝트 Clone 방법
2. Plant UML 사용법
   1. 공통 문법
   2. Class Diagram
   3. Sequence Diagram
3. 사용 예시
   1. Class Diagram
   2. Sequence Diagram

# Plant UML 설정

## IntelliJ 설치

## Plant UML 다운로드

- Graphviz 설치 : Graphviz 다운로드

- palntuml.jar 다운로드 및 설정 : palntuml.jar를 C:\UserPrograms에 다운로드 후 환경변수 설정

1. 프로젝트 clone 방법

# Plant UML 사용법

1. 공통문법

|  |  |  |
| --- | --- | --- |
| **기능** | **문법** | **화면** |
| UML 시작과 끝 문법 | @startuml…..@enduml | 텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진  자동 생성된 설명 |

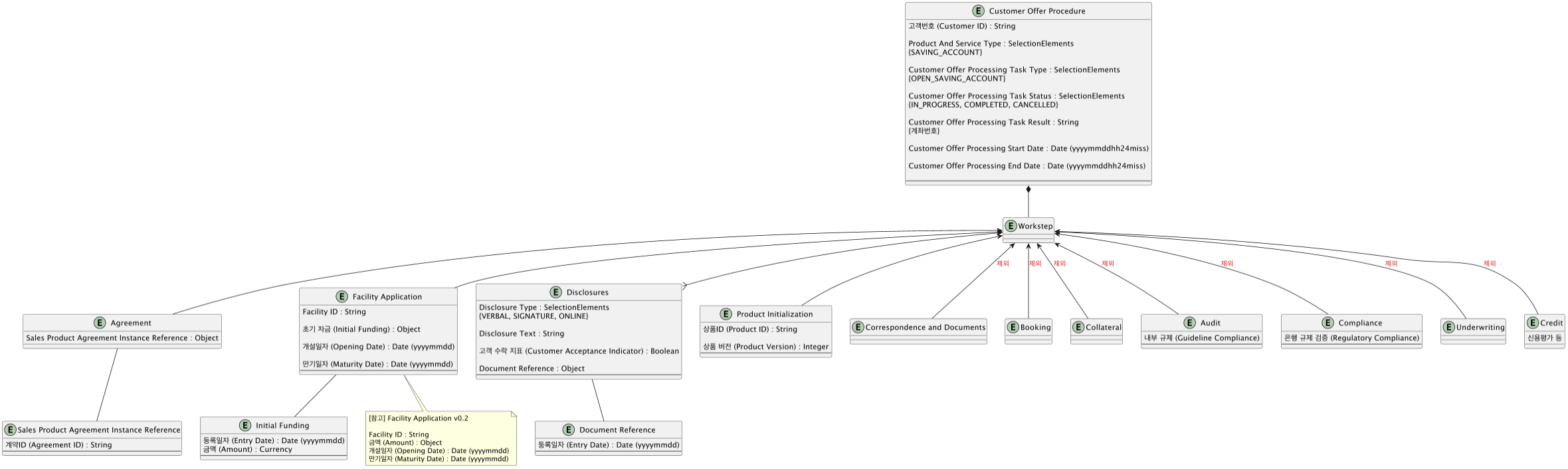
1. Class Diagram

|  |  |  |
| --- | --- | --- |
| **기능** | **문법** | **화면** |
| Entity 선언 | @startumlentity "Product Initialization"{ 상품ID (Product ID) : String  상품 버전 (Product Version) : Integer -- }@enduml | 텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진  자동 생성된 설명 |
| 화살표 방향, 내용이 없는 class는 선언하지 않아도 자동 생성(default : class Entity) | @startuml  foo -left-> dummyLeft  foo -right-> dummyRight  foo -up-> dummyUp  foo -down-> dummyDown  @enduml | 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진  자동 생성된 설명 |
| 화살표 종류 | @startuml  Class01 <|-- Class02  Class03 \*-- Class04  Class05 o-- Class06  Class07 .. Class08  Class09 -- Class10  @enduml | 스크린샷, 텍스트, 폰트, 라인이(가) 표시된 사진  자동 생성된 설명 |
|  | @startuml  Class11 <|.. Class12  Class13 --> Class14  Class15 ..> Class16  Class17 ..|> Class18  Class19 <--\* Class20  @enduml | 스크린샷, 텍스트, 폰트, 도표이(가) 표시된 사진  자동 생성된 설명 |
|  | @startuml  Class21 #-- Class22  Class23 x-- Class24  Class25 }-- Class26  Class27 +-- Class28  Class29 ^-- Class30  @enduml | 스크린샷, 텍스트, 폰트, 라인이(가) 표시된 사진  자동 생성된 설명 |
| 그룹핑 : skinparam groupInheritance를 선언하고 숫자를 기입하면 기입한 숫자 이상의 화살표는 그룹화가 됨 | @startuml  skinparam groupInheritance 2  A1 <|-- B1  A2 <|-- B2  A2 <|-- C2  A3 <|-- B3  A3 <|-- C3  A3 <|-- D3  @enduml | 도표, 라인, 디자인이(가) 표시된 사진  자동 생성된 설명 |
| 화살표/텍스트 색상 | @startuml  foo --> bar : normal  foo --> bar1 #line:red;line.bold;text:red : red bold  foo --> bar2 #green;line.dashed;text:green : green dashed  foo --> bar3 #blue;line.dotted;text:blue : blue dotted  @enduml | 텍스트, 도표, 라인, 폰트이(가) 표시된 사진  자동 생성된 설명 |
| 참고 사항 기입 | @startuml  note bottom of "Facility Application" [참고] Facility Application v0.2  Facility ID : String 금액 (Amount) : Object 개설일자 (Opening Date) : Date (yyyymmdd) 만기일자 (Maturity Date) : Date (yyyymmdd) end note  @enduml | 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명 |

1. Sequence Diagram

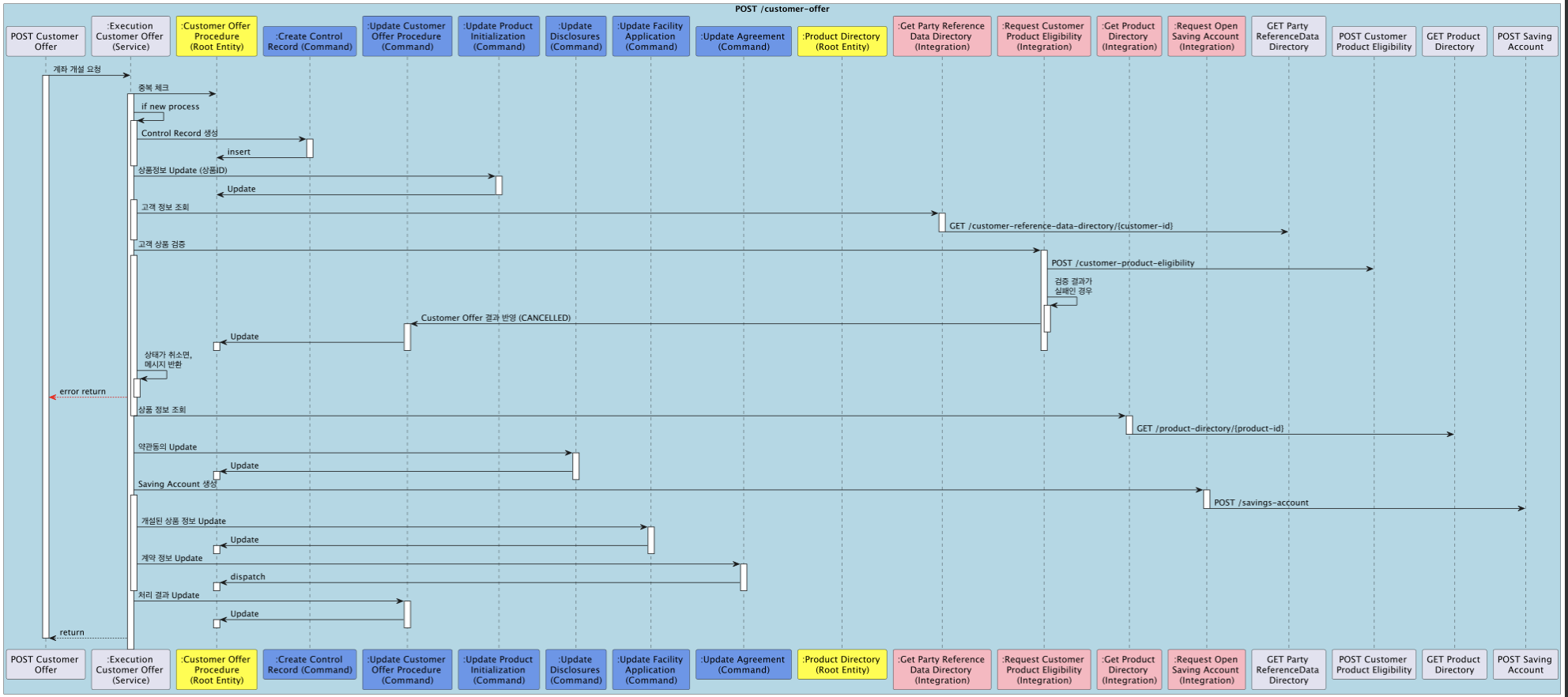
|  |  |  |
| --- | --- | --- |
| **기능** | **문법** | **화면** |
| Participant 선언 : participant, actor, boundary, control, entity, database, collections, queue를 각각의 모양으로 선언할 수 있으며 ‘as’ 로 변수명을 간단히 설정할 수 있음 | @startumlparticipant Participant as Fooactor Actor as Foo1boundary Boundary as Foo2control Control as Foo3entity Entity as Foo4database Database as Foo5collections Collections as Foo6queue Queue as Foo7Foo -> Foo1 : To actorFoo -> Foo2 : To boundaryFoo -> Foo3 : To controlFoo -> Foo4 : To entityFoo -> Foo5 : To databaseFoo -> Foo6 : To collectionsFoo -> Foo7: To queue@enduml | 텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진  자동 생성된 설명 |
| Participant 색상 / 화살표 색상 | @startuml participant Alice participant Bob #99FF99  Alice->Bob: Authentication Request Bob-[#red]>Alice: Authentication Response @enduml |  |
| 화살표 방향 : 자기 자신에게 | @startuml Alice -> Alice : 확인 @enduml | 텍스트, 폰트, 스크린샷, 번호이(가) 표시된 사진  자동 생성된 설명 |
| 분기 추가 : group 또는 loop를 선언하고 [ ]에 머릿글을 추가할 수 있음 | @startuml  Alice -> Bob: Authentication Request  Bob -> Alice: Authentication Failure  group My own label [My own label 2]  Alice -> Log : Log attack start  loop 1000 times  Alice -> Bob: DNS Attack  end  Alice -> Log : Log attack end  end  @enduml |  |
| 생명선 : activate 와 deactivate 는 participant의 활성화여부를 표현함 | @startuml participant User User -> A: DoWork activate A A -> B: createRequest activate B B --> A: RequestCreated deactivate B A -> User: Done deactivate A @enduml | 텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진  자동 생성된 설명 |

1. 사용 예시
2. Class Diagram



|  |
| --- |
| @startuml  entity "Customer Offer Procedure" {  고객번호 (Customer ID) : String  Product And Service Type : SelectionElements  {SAVING\_ACCOUNT}  Customer Offer Processing Task Type : SelectionElements  {OPEN\_SAVING\_ACCOUNT}  Customer Offer Processing Task Status : SelectionElements  {IN\_PROGRESS, COMPLETED, CANCELLED}  Customer Offer Processing Task Result : String  {계좌번호}  Customer Offer Processing Start Date : Date (yyyymmddhh24miss)  Customer Offer Processing End Date : Date (yyyymmddhh24miss)  --  }  entity Workstep{  }  entity "Product Initialization"{  상품ID (Product ID) : String  상품 버전 (Product Version) : Integer  --  }  entity Agreement{  Sales Product Agreement Instance Reference : Object  }  entity "Correspondence and Documents"  entity "Booking"  entity "Collateral"  entity "Audit"{  내부 규제 (Guideline Compliance)  --  }  entity "Compliance"{  은행 규제 검증 (Regulatory Compliance)  --  }  entity "Underwriting"  entity "Credit"{  신용평가 등  }  entity "Disclosures"{  Disclosure Type : SelectionElements  {VERBAL, SIGNATURE, ONLINE}  Disclosure Text : String  고객 수락 지표 (Customer Acceptance Indicator) : Boolean  Document Reference : Object  --  }  entity "Facility Application"{  Facility ID : String  초기 자금 (Initial Funding) : Object  개설일자 (Opening Date) : Date (yyyymmdd)  만기일자 (Maturity Date) : Date (yyyymmdd)  --  }  entity "Document Reference"{  둥록일자 (Entry Date) : Date (yyyymmdd)  --  }  entity "Initial Funding"{  둥록일자 (Entry Date) : Date (yyyymmdd)  금액 (Amount) : Currency  --  }  entity "Sales Product Agreement Instance Reference"{  계약ID (Agreement ID) : String  --  }  note bottom of "Facility Application"  [참고] Facility Application v0.2  Facility ID : String  금액 (Amount) : Object  개설일자 (Opening Date) : Date (yyyymmdd)  만기일자 (Maturity Date) : Date (yyyymmdd)  end note  Workstep -up-\* "Customer Offer Procedure"  "Product Initialization" -up-> Workstep  Agreement -up-> Workstep  "Correspondence and Documents" -up-> Workstep #text:red : 제외  "Booking" -up-> Workstep #text:red : 제외  "Collateral" -up-> Workstep #text:red : 제외  "Audit" -up-> Workstep #text:red : 제외  "Compliance" -up-> Workstep #text:red : 제외  "Underwriting" -up-> Workstep #text:red : 제외  "Credit" -up-> Workstep #text:red : 제외  "Disclosures" }-up-> Workstep  "Facility Application" -up-> Workstep  "Document Reference" -up- "Disclosures"  "Initial Funding" -up- "Facility Application"  "Sales Product Agreement Instance Reference" -up- Agreement  @enduml |

1. Sequence Diagram



|  |
| --- |
| @startuml  !pragma teoz true  box "POST /customer-offer" #LightBlue  participant "POST Customer \nOffer"  participant ":Execution \nCustomer Offer \n(Service)"  participant ":Customer Offer \nProcedure\n(Root Entity)" #Yellow  participant ":Create Control \nRecord (Command)" #6495ED  participant ":Update Customer \nOffer Procedure \n(Command)" #6495ED  participant ":Update Product \nInitialization \n(Command)" #6495ED  participant ":Update \nDisclosures \n(Command)" #6495ED  participant ":Update Facility \nApplication \n(Command)" #6495ED  participant ":Update Agreement \n(Command)" #6495ED  participant ":Product Directory \n(Root Entity)" #Yellow  participant ":Get Party Reference \nData Directory \n(Integration)" #Lightpink  participant ":Request Customer \nProduct Eligibility\n(Integration)" #Lightpink  participant ":Get Product \nDirectory \n(Integration)" #Lightpink  participant ":Request Open \nSaving Account \n(Integration)" #Lightpink  participant "GET Party \nReferenceData \nDirectory"  participant "POST Customer \nProduct Eligibility"  participant "GET Product \nDirectory"  participant "POST Saving \nAccount"  end box  "POST Customer \nOffer" -> ":Execution \nCustomer Offer \n(Service)" : 계좌 개설 요청  activate "POST Customer \nOffer"  ":Execution \nCustomer Offer \n(Service)" -> ":Customer Offer \nProcedure\n(Root Entity)" : 중복 체크  activate ":Execution \nCustomer Offer \n(Service)"  ":Execution \nCustomer Offer \n(Service)" -> ":Execution \nCustomer Offer \n(Service)" ++ : if new process  ":Execution \nCustomer Offer \n(Service)" -> ":Create Control \nRecord (Command)" : Control Record 생성  activate ":Create Control \nRecord (Command)"  ":Create Control \nRecord (Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : insert  deactivate ":Create Control \nRecord (Command)"  deactivate ":Execution \nCustomer Offer \n(Service)"  ":Execution \nCustomer Offer \n(Service)" -> ":Update Product \nInitialization \n(Command)" : 상품정보 Update (상품ID)  activate ":Update Product \nInitialization \n(Command)"  ":Update Product \nInitialization \n(Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : Update  deactivate ":Update Product \nInitialization \n(Command)"  activate ":Execution \nCustomer Offer \n(Service)"  ":Execution \nCustomer Offer \n(Service)" -> ":Get Party Reference \nData Directory \n(Integration)" : 고객 정보 조회  activate ":Get Party Reference \nData Directory \n(Integration)"  ":Get Party Reference \nData Directory \n(Integration)" -> "GET Party \nReferenceData \nDirectory" : GET /customer-reference-data-directory/{customer-id}  deactivate ":Get Party Reference \nData Directory \n(Integration)"  deactivate ":Execution \nCustomer Offer \n(Service)"  ":Execution \nCustomer Offer \n(Service)" -> ":Request Customer \nProduct Eligibility\n(Integration)" : 고객 상품 검증  activate ":Request Customer \nProduct Eligibility\n(Integration)"  activate ":Execution \nCustomer Offer \n(Service)"  ":Request Customer \nProduct Eligibility\n(Integration)" -> "POST Customer \nProduct Eligibility" : POST /customer-product-eligibility  ":Request Customer \nProduct Eligibility\n(Integration)" -> ":Request Customer \nProduct Eligibility\n(Integration)" ++ : 검증 결과가\n실패인 경우  ":Request Customer \nProduct Eligibility\n(Integration)" -> ":Update Customer \nOffer Procedure \n(Command)" : Customer Offer 결과 반영 (CANCELLED)  activate ":Update Customer \nOffer Procedure \n(Command)"  deactivate ":Request Customer \nProduct Eligibility\n(Integration)"  ":Update Customer \nOffer Procedure \n(Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : Update  activate ":Customer Offer \nProcedure\n(Root Entity)"  deactivate ":Update Customer \nOffer Procedure \n(Command)"  deactivate ":Request Customer \nProduct Eligibility\n(Integration)"  deactivate ":Customer Offer \nProcedure\n(Root Entity)"  ":Execution \nCustomer Offer \n(Service)" -> ":Execution \nCustomer Offer \n(Service)" ++ : 상태가 취소면,\n메시지 반환  ":Execution \nCustomer Offer \n(Service)" --[#red]> "POST Customer \nOffer" : error return  deactivate ":Execution \nCustomer Offer \n(Service)"  deactivate ":Execution \nCustomer Offer \n(Service)"  ":Execution \nCustomer Offer \n(Service)" -> ":Get Product \nDirectory \n(Integration)" : 상품 정보 조회  activate ":Get Product \nDirectory \n(Integration)"  ":Get Product \nDirectory \n(Integration)" -> "GET Product \nDirectory" : GET /product-directory/{product-id}  deactivate ":Get Product \nDirectory \n(Integration)"  ":Execution \nCustomer Offer \n(Service)" -> ":Update \nDisclosures \n(Command)" : 약관동의 Update  activate ":Update \nDisclosures \n(Command)"  ":Update \nDisclosures \n(Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : Update  activate ":Customer Offer \nProcedure\n(Root Entity)"  deactivate ":Update \nDisclosures \n(Command)"  deactivate ":Customer Offer \nProcedure\n(Root Entity)"  ":Execution \nCustomer Offer \n(Service)" -> ":Request Open \nSaving Account \n(Integration)" : Saving Account 생성  activate ":Request Open \nSaving Account \n(Integration)"  activate ":Execution \nCustomer Offer \n(Service)"  ":Request Open \nSaving Account \n(Integration)" -> "POST Saving \nAccount" : POST /savings-account  deactivate ":Request Open \nSaving Account \n(Integration)"  ":Execution \nCustomer Offer \n(Service)" -> ":Update Facility \nApplication \n(Command)" : 개설된 상품 정보 Update  activate ":Update Facility \nApplication \n(Command)"  ":Update Facility \nApplication \n(Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : Update  activate ":Customer Offer \nProcedure\n(Root Entity)"  deactivate ":Update Facility \nApplication \n(Command)"  deactivate ":Customer Offer \nProcedure\n(Root Entity)"  ":Execution \nCustomer Offer \n(Service)" -> ":Update Agreement \n(Command)" : 계약 정보 Update  activate ":Update Agreement \n(Command)"  ":Update Agreement \n(Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : dispatch  activate ":Customer Offer \nProcedure\n(Root Entity)"  deactivate ":Update Agreement \n(Command)"  deactivate ":Customer Offer \nProcedure\n(Root Entity)"  deactivate ":Execution \nCustomer Offer \n(Service)"  ":Execution \nCustomer Offer \n(Service)" -> ":Update Customer \nOffer Procedure \n(Command)" : 처리 결과 Update  activate ":Update Customer \nOffer Procedure \n(Command)"  ":Update Customer \nOffer Procedure \n(Command)" -> ":Customer Offer \nProcedure\n(Root Entity)" : Update  activate ":Customer Offer \nProcedure\n(Root Entity)"  deactivate ":Update Customer \nOffer Procedure \n(Command)"  deactivate ":Customer Offer \nProcedure\n(Root Entity)"  ":Execution \nCustomer Offer \n(Service)" --> "POST Customer \nOffer" : return  deactivate "POST Customer \nOffer"  @enduml |

가이드 참조 문서 : https://pdf.plantuml.net/PlantUML\_Language\_Reference\_Guide\_ko.pdf