

v.0.0.4

최병권

AI와 대화하기

목차

A TABLE OF CONTENTS

- 1 기획의도
- 2 구성요소
- 3 Dialog-flow
- 4 Back-End



The image features a central composition of overlapping, semi-transparent spheres in shades of orange, teal, and blue. A large white number '1' is positioned in the center of these spheres. Surrounding the central spheres are three triangles: an orange triangle at the top, a purple triangle on the left, and a blue triangle on the right. Each triangle is outlined in white and partially overlaps the central spheres. The background is a smooth gradient transitioning from light blue on the left to light orange on the right.

1

기획의도

기획 의도

요즘 가장 유행하고 있는 Chat-GPT 영감을 받아서 Google Cloud에서 머신 러닝을 진행하였고, 일반 사람들과 다르지 않게 대화를 주고 받을 수 있게 작성해보았다. Google Cloud Platfrom(GCP)에서 Intents와 Entities를 추가하여 진행하였다. (intent, Entity는 대화를 주고 받기 위해 로봇 두뇌에 대답 리스트를 집어넣는 과정)





2

구성요소



- java 17
- Spring Boot
- Postgres

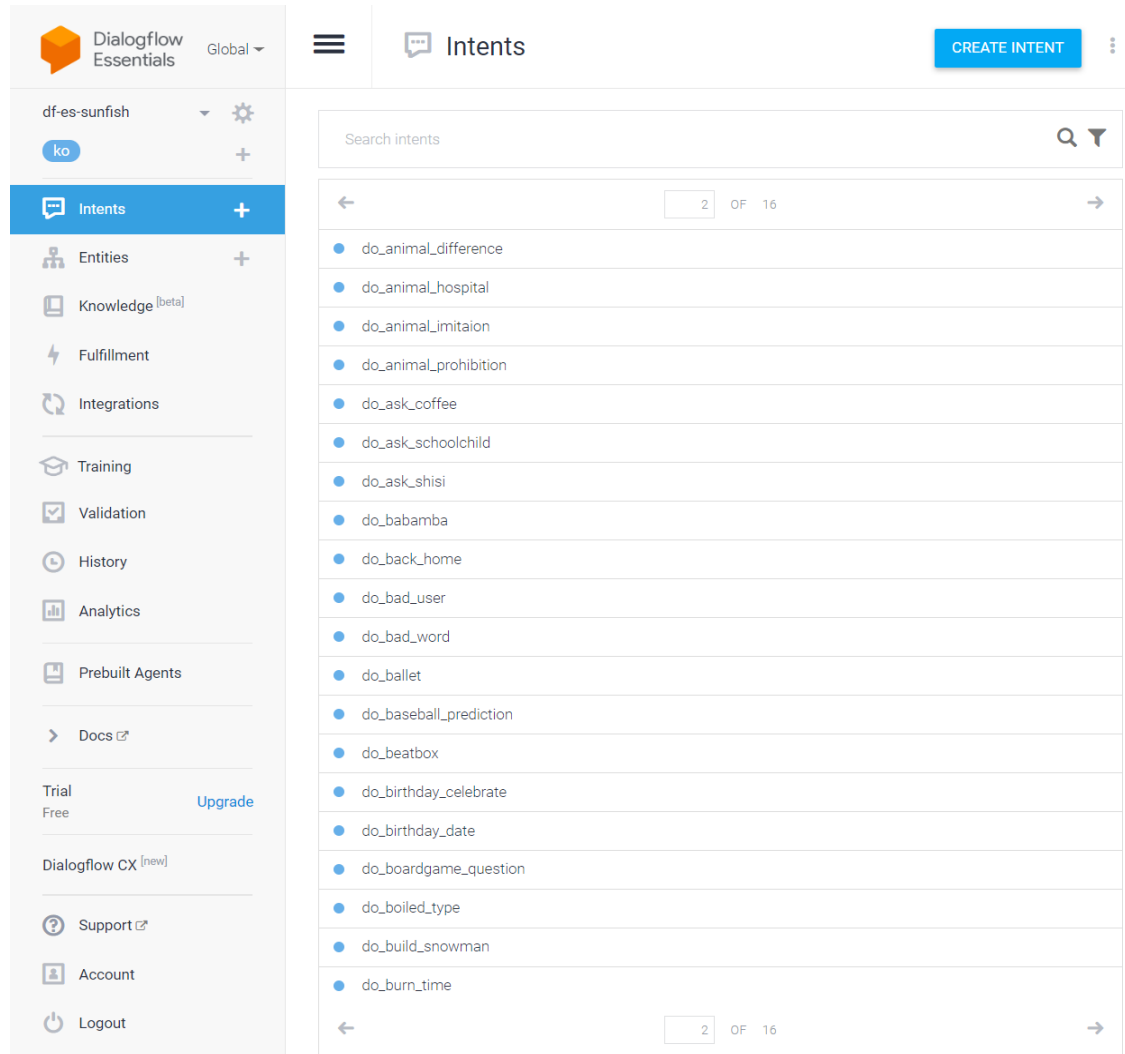




3

Dialog-flow

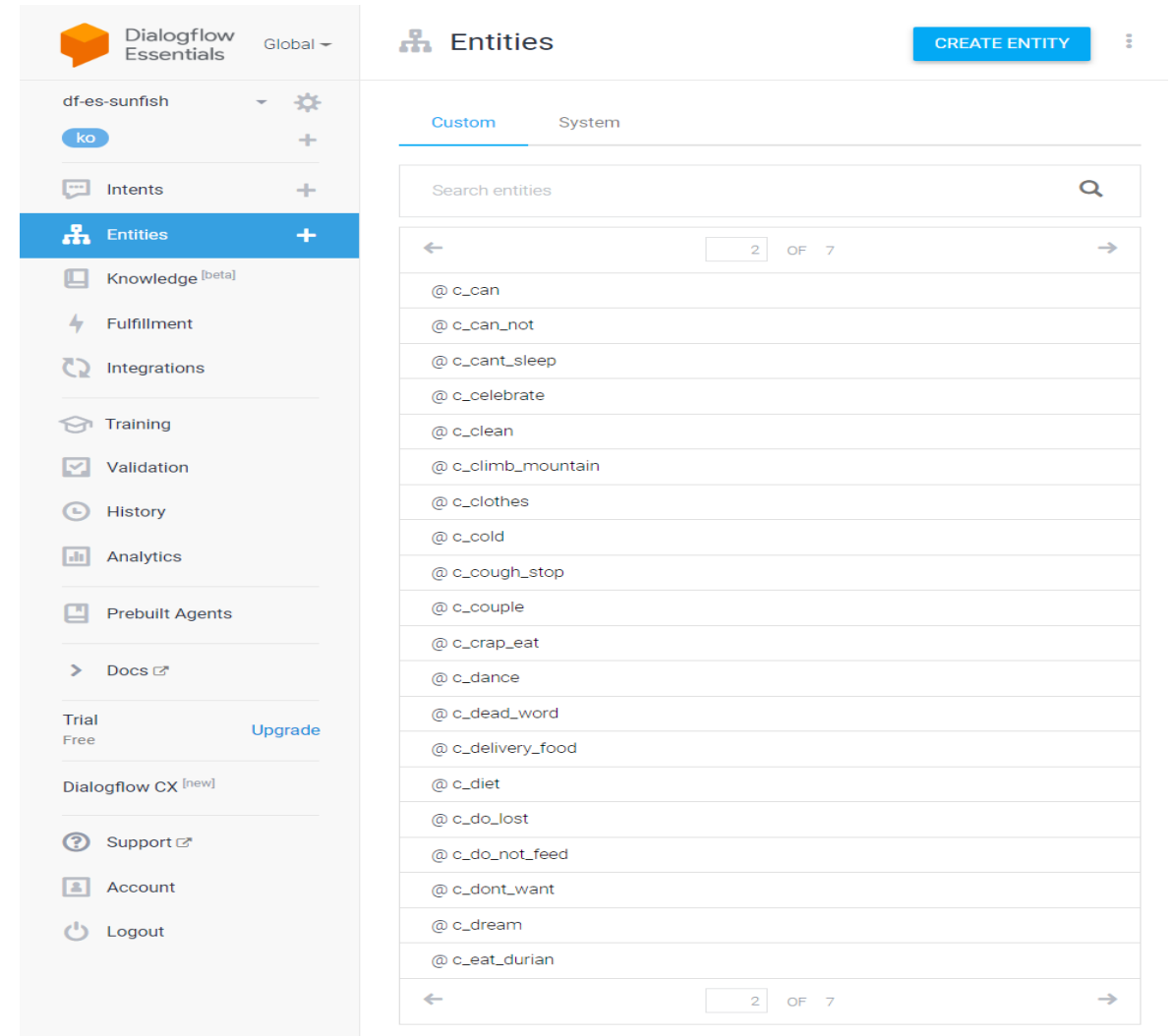
Part 3



The screenshot shows the Dialogflow Intents page for the 'df-es-sunfish' agent. The left sidebar contains navigation links for Intents, Entities, Knowledge, Fulfillment, Integrations, Training, Validation, History, Analytics, and Prebuilt Agents. The main area displays a list of 16 intents, with the first two being 'do_animal_difference' and 'do_animal_hospital'. The page number '2 OF 16' is visible at the bottom of the list.

Intent
do_animal_difference
do_animal_hospital
do_animal_imitation
do_animal_prohibition
do_ask_coffee
do_ask_schoolchild
do_ask_shisi
do_babamba
do_back_home
do_bad_user
do_bad_word
do_ballet
do_baseball_prediction
do_beatbox
do_birthday_celebrate
do_birthday_date
do_boardgame_question
do_boiled_type
do_build_snowman
do_burn_time

(총 16 페이지)



The screenshot shows the Dialogflow Entities page for the 'df-es-sunfish' agent. The left sidebar contains navigation links for Intents, Entities, Knowledge, Fulfillment, Integrations, Training, Validation, History, Analytics, and Prebuilt Agents. The main area displays a list of 7 entities, with the first two being '@ c_can' and '@ c_can_not'. The page number '2 OF 7' is visible at the bottom of the list.

Entity
@ c_can
@ c_can_not
@ c_cant_sleep
@ c_celebrate
@ c_clean
@ c_climb_mountain
@ c_clothes
@ c_cold
@ c_cough_stop
@ c_couple
@ c_crap_eat
@ c_dance
@ c_dead_word
@ c_delivery_food
@ c_diet
@ c_do_lost
@ c_do_not_feed
@ c_dont_want
@ c_dream
@ c_eat_durian

(총 7 페이지)



4

Back-End

```

@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class DfeCondition {

    no usages
    @ApiModelProperty(notes = "시퀀스")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    1 usage
    @ApiModelProperty(notes = "df-entity 이름")
    @Column(nullable = false, length = 50)
    private String dfeName;

    1 usage
    @ApiModelProperty(notes = "df-entity 타입")
    @Column(nullable = false, length = 10)
    @Enumerated(EnumType.STRING)
    private DfeKind dfeKind;

    1 usage
    @ApiModelProperty(notes = "영향 여부")
    @Column(nullable = false)
    private Boolean isImpact;

```

```

@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class DieCondition {

    no usages
    @ApiModelProperty(notes = "시퀀스")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    1 usage
    @ApiModelProperty(notes = "intent 명")
    @Column(nullable = false, length = 150)
    private String intentName;

    1 usage
    private DieCondition(Builder builder) { this.intentName = builder.intentName; }

    1 usage
    public static class Builder implements CommonModelBuilder<DieCondition> {

        2 usages
        private final String intentName;

        no usages
        public Builder(DieConditionCreateRequest conditionCreateRequest) {
            this.intentName = conditionCreateRequest.getIntentName();
        }

        2 usages
        @Override
        public DieCondition build() { return new DieCondition( builder: this); }

    }
}

```

DfeKind

```
package shop.dev.choibk.api.game.logic.enums;

import lombok.AllArgsConstructor;
import lombok.Getter;

5 usages
@Getter
@AllArgsConstructor
public enum DfeKind {
    no usages
    POSITIVE( name: "긍정", weighted: 10),
    no usages
    NEGATIVE( name: "부정", weighted: -10),
    no usages
    NEUTRAL( name: "중립", weighted: 0);

    no usages
    private final String name;
    no usages
    private final Integer weighted; // 곱셈 연산
}
```


DialogFlowResponse

```
@Getter
@Setter
public class DialogFlowResponse {

    no usages
    @ApiModelProperty(notes = "사람이 개복치에게 하는 말")
    private String queryText;

    no usages
    @ApiModelProperty(notes = "개복치가 사람에게 하는 말")
    private String fulfillmentText;

    no usages
    @ApiModelProperty(notes = "df- 의도 이름")
    private String intentName;

    no usages
    @ApiModelProperty(notes = "df- 사용된 Entity")
    private List<String> entitys;
}
```

```

@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class DfeCondition {

    no usages
    @ApiModelProperty(notes = "시퀀스")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    1 usage
    @ApiModelProperty(notes = "df-entity 이름")
    @Column(nullable = false, length = 50)
    private String dfeName;

    1 usage
    @ApiModelProperty(notes = "df-entity 타입")
    @Column(nullable = false, length = 10)
    @Enumerated(EnumType.STRING)
    private DfeKind dfeKind;

    1 usage
    @ApiModelProperty(notes = "영향 여부")
    @Column(nullable = false)
    private Boolean isImpact;

```

```

@Entity
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class DieCondition {

    no usages
    @ApiModelProperty(notes = "시퀀스")
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    1 usage
    @ApiModelProperty(notes = "intent 명")
    @Column(nullable = false, length = 150)
    private String intentName;

    1 usage
    private DieCondition(Builder builder) { this.intentName = builder.intentName; }

    1 usage
    public static class Builder implements CommonModelBuilder<DieCondition> {

        2 usages
        private final String intentName;

        no usages
        public Builder(DieConditionCreateRequest conditionCreateRequest) {
            this.intentName = conditionCreateRequest.getIntentName();
        }

        2 usages
        @Override
        public DieCondition build() { return new DieCondition( builder: this); }

    }
}

```

GameRequest

```
@Getter
@Setter
public class GameRequest {
    no usages
    @ApiModelProperty(notes = "개복치에게 요청할 말")
    @NotNull
    @Length(min = 5, max = 100)
    private String message;
}
```

GameResponse

```
@Getter
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class GameResponse {
    1 usage
    @ApiModelProperty(notes = "사람 입력값")
    private String humanQueryText;

    1 usage
    @ApiModelProperty(notes = "ai 응답값")
    private String aiFulfillmentText;

    1 usage
    @ApiModelProperty(notes = "사망 여부")
    private Boolean isDead;

    1 usage
    @ApiModelProperty(notes = "변화되는 생명력 수치")
    private Integer changeVitalityFigures;
}
```


Repository

DfeConditionRepository

```
package shop.dev.choibk.api.game.logic.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import shop.dev.choibk.api.game.logic.entity.DfeCondition;

import java.util.Optional;

1 usage
public interface DfeConditionRepository extends JpaRepository<DfeCondition, Long> {
    no usages
    Optional<DfeCondition> findByDfeName(String dfeName);
}
```

DieConditionRepository

```
package shop.dev.choibk.api.game.logic.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import shop.dev.choibk.api.game.logic.entity.DieCondition;

1 usage
public interface DieConditionRepository extends JpaRepository<DieCondition, Long> {
    no usages
    long countByIntentName(String intentName);
}
```

DialogFlowService

```
@Service
public class DialogFlowService {
    no usages
    public DialogFlowResponse detectIntentTexts(String callMsg) throws ApiException {
        String projectId = "vivid-memento-379101"; // ID 확인 필수

        String randomSessionId = UUID.randomUUID().toString();

        String langCode = "ko-KR";

        // Instantiates a client
        try (SessionsClient sessionsClient = SessionsClient.create()) {
            // Set the session name using the sessionId (UUID) and projectID (my-project-id)
            SessionName session = SessionName.of(projectId, randomSessionId);

            TextInput.Builder textInput = TextInput.newBuilder().setText(callMsg).setLanguageCode(langCode);

            // Build the query with the TextInput
            QueryInput queryInput = QueryInput.newBuilder().setText(textInput).build();

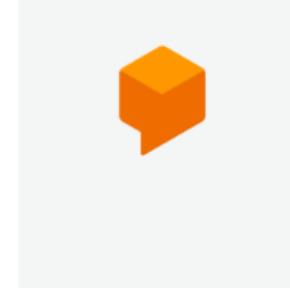
            // Performs the detect intent request
            DetectIntentResponse response = sessionsClient.detectIntent(session, queryInput);

            QueryResult result = response.getQueryResult();

            DialogFlowResponse resultResponse = new DialogFlowResponse();
            resultResponse.setQueryText(result.getQueryText());
            resultResponse.setFulfillmentText(result.getFulfillmentText());
            resultResponse.setIntentName(result.getIntent().getDisplayName());

            List<String> dfEntitys = new LinkedList<>();
            Struct temp1 = result.getParameters();
            Map<String, Value> temp2 = temp1.getFieldsMap();
            temp2.forEach((key, value) -> {
                dfEntitys.add(key);
            });
            resultResponse.setEntitys(dfEntitys);

            return resultResponse;
        }
    }
}
```



DESCRIPTION

Describe your agent (will be used in Web Demo integration)

DEFAULT TIME ZONE

(GMT+9:00) Asia/Tokyo

Date and time requests are resolved using this timezone if not provided in the API requests.

AGENT AVATAR URI

Define URI to agent avatar that will be used in [Web Demo](#) and [Google Chat](#) integrations.

GOOGLE PROJECT

Project ID

vivid-memento-379101

GameService (1)

```

@Service
@RequiredArgsConstructor
public class GameService {

    1 usage
    private final DieConditionRepository dieConditionRepository;

    1 usage
    private final DfeConditionRepository dfeConditionRepository;

    1 usage
    private final DialogFlowService dialogFlowService;

    no usages
    public GameResponse doCalculation(GameRequest gameRequest) {
        DialogFlowResponse dialogFlowResponse = dialogFlowService.detectIntentTexts(gameRequest.getMessage());

        // ----- 즉사여부 확인 -----
        // DieCondition 에서 위에서 가져온 displayName 에 해당하는 값이 있는지 확인함 -> count 로 확인
        long intentDuplicateCount = dieConditionRepository.countByIntentName(dialogFlowResponse.getIntentName());

        // 만약 데이터가 있으면 즉사처리
        if (intentDuplicateCount > 0) {
            return new GameResponse.Builder(dialogFlowResponse.getQueryText(),
                dialogFlowResponse.getFulfillmentText(),
                true, 0).build();
        } else { // 데이터가 없으면 점수 계산하는 2단계로 넘어감
            return new GameResponse.Builder(dialogFlowResponse.getQueryText(),
                dialogFlowResponse.getFulfillmentText(),
                false, calculateChangeVitalityFigures(dialogFlowResponse.getEntitys())).build();
        }

        // ---- 즉사여부 확인 끝 ----
    }
}

```

GameService (2)

```

1 usage
private int calculateChangeVitalityFigures(List<String> paramKeys) {
    List<DfeCondition> dfeConditions = new LinkedList<>();

    // 내가 가진 키들의 상세 데이터를 dfeConditions 에 모
    for (String key : paramKeys) {
        Optional<DfeCondition> tempResult = dfeConditionRepository.findByDfeName(key);
        tempResult.ifPresent(dfeConditions::add);
    }

    int weightValue = 0; // 최종 가중치
    int weightCount = 0; // 수치 영향 여부 총 갯수
    for (DfeCondition dfeCondition : dfeConditions) {
        weightValue += dfeCondition.getDfeKind().getWeighted();
        if (dfeCondition.getIsImpact()) {
            weightCount += 1;
        }
    }

    return weightValue * weightCount;
}
}

```


GameController

```
@Api(tags = "대화")
@RestController
@RequiredArgsConstructor
@RequestMapping("/v1/game")
public class GameController {
    1 usage
    private final GameService gameService;

    no usages
    @ApiOperation(value = "개복치와 대화하기")
    @PostMapping("/send")
    public SingleResult<GameResponse> sendMessage(@RequestBody @Valid GameRequest gameRequest) {
        return ResponseService.getSingleResult(gameService.doCalculation(gameRequest));
    }
}
```

 **swagger**

Select a spec V1

AI와 대화하기 API V1 ^{V1}

[Base URL: localhost:8080/]
<http://localhost:8080/v2/api-docs?group=V1>

Swagger API Docs

[Terms of service](#)

[mmr - Website](#)

[Send email to mmr](#)

[Licenses](#)

Authorize 

대화 Game Controller 

POST

/v1/game/send 개복치와 대화하기



대화 Game Controller

POST /v1/game/send 개복치와 대화하기

Parameters

Name

Description

gameRequest * required

gameRequest

(body)

Example Value | Model

```
{
  "message": "밥 먹는다!"
}
```

사람이 입력할 대화

Request URL

`http://localhost:8080/v1/game/send`

Server response

Code

Details

200

Response body

```
{
  "isSuccess": true,
  "code": 0,
  "msg": "성공하였습니다.",
  "data": {
    "humanQueryText": "밥 먹는다!",
    "aiFulfillmentText": "맛있게 먹어~",
    "isDead": false,
    "changeVitalityFigures": 0
  }
}
```

AI가 출력할 대답

대화 Game Controller

POST /v1/game/send 개복치와 대화하기

Parameters

Name	Description
gameRequest * required (body)	gameRequest

Example Value | Model

```
{  "message": "넌 멍청이야"}
```

↓
사람이 입력할 대화

Request URL

http://localhost:8080/v1/game/send

Server response

Code	Details
200	<div>Response body<pre>{ "isSuccess": true, "code": 0, "msg": "성공하였습니다.", "data": { "humanQueryText": "넌 멍청이야", "aiFulfillmentText": "멍청이라니.. 너무해!!!!", "isDead": false, "changeVitalityFigures": 0 }}</pre><p>↓ AI가 출력할 대답</p></div>

대화 Game Controller

POST /v1/game/send 개복치와 대화하기

Parameters

Name	Description
gameRequest * required (body)	gameRequest

Example Value | Model

```
{  "message": "야채 좋아하니?"}
```

↓

사람이 입력할 대화

Request URL

`http://localhost:8080/v1/game/send`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "isSuccess": true, "code": 0, "msg": "성공하였습니다.", "data": { "humanQueryText": "야채 좋아하니?", "aiFulfillmentText": "당연히 좋아하지 ^^", "isDead": false, "changeVitalityFigures": 0 }}</pre> <p>↓</p> <p>AI가 출력할 대답</p>

대화 Game Controller

POST /v1/game/send 개복치와 대화하기

Parameters

Name	Description
gameRequest * required (body)	gameRequest

Example Value	Model
<pre>{ "message": "투수 오타니는 어떤사람 같니?"}</pre>	

↓

사람이 입력할 대화

Request URL

`http://localhost:8080/v1/game/send`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "isSuccess": true, "code": 0, "msg": "성공하였습니다.", "data": { "humanQueryText": "투수 오타니는 어떤사람 같니?", "aiFulfillmentText": "멋있는 사람 같아", "isDead": false, "changeVitalityFigures": 0 }}</pre> <p>↓</p> <p>AI가 출력할 대답</p>



The End