

## 취준 에이전트



면접관

~면접질문~

~질문 답변~



사용자

~답변평가~



면접관

꼬리질문

모범답변

다음질문

메시지입력

보내기

1. 대화의 시작은 면접관이 질문을 하는것으로 시작한다.
2. 내가 응답을 하면 면접관은 해당 응답에 대한 평가를 제시하고, 하단에 [ 꼬리질문/모범답변/다음질문 ] 버튼을 생성한다.
3. 꼬리질문 클릭시
  - a. 질문, 답변을 기반으로 한 꼬리질문을 생성하여 질의한다.
4. 모범답변 클릭시
  - a. 질문, 답변을 기반으로 사용자의 답변을 평가한다.
5. 다음질문 클릭시
  - a. 새로운 질문을 생성한다.

## 취준 에이전트



면  
접  
관

~답변평가~

꼬리질문

모범답변

다음질문

꼬리질문



사  
용  
자



면  
접  
관

~꼬리질문~

~답변~



사  
용  
자

메시지입력

보내기

1. 사용자가 꼬리질문을 클릭하면, 꼬리질문 메시지를 입력하게 된다.
2. 꼬리질문이 입력되면 면접관은 지난 질문 + 사용자 답변을 기반으로 꼬리질문을 생성한다.
3. 사용자 답변이 입력되면 그에 대한 답변평가 [ 꼬리질문, 모범답변, 다음질문 ] 버튼을 동일하게 노출한다.

## 취준 에이전트



면  
접  
관

~답변평가~

꼬리질문

모범답변

다음질문

모범답변



사  
용  
자



면  
접  
관

~모범답변~

꼬리질문

모범답변

다음질문

메시지입력

보내기

1. 사용자가 모범답변을 클릭하면, 모범답변 메시지를 입력하게 된다.
2. 모범답변이 입력되면 면접관은 지난 질문 + 사용자 답변을 기반으로 모범답변을 생성하고, [ 꼬리질문, 모범답변, 다음질문 ] 버튼을 노출한다.

## 취준 에이전트



면접  
관

~답변평가~

꼬리질문

모범답변

다음질문

다음질문



사용  
자



면접  
관

~다음질문~

~답변~



사용  
자

메시지입력

보내기

1. 사용자가 다음질문을 클릭하면, 다음질문 메시지를 입력하게 된다.
2. 다음질문 이후엔 첫번째 **flow**와 같다.

```

Enums
- contentType : 'question' | 'answer' | 'modelAnser'

DTO
- ChatHistoryDTO
    - {
        id : string;
        children : List<ChatHistoryDTO> | null;
        type : contentType;
        speaker : string;
        content: string;
    }
- AnswerRequestDTO
    - {
        questionId: string;
        content : string;
    }
- QuestionDTO
    - {
        id: string;
        content : string;
    }
- AnswerDTO
    - {
        id : string;
        content : string;
    }

```

#### api 명세

1. (GET) /chatHistory
  - a. payload : null
  - b. response : List<ChatHistoryDTO>
2. (POST) /answer
  - a. payload : AnswerRequestDTO
  - b. response : null
3. (GET) /followUp
  - a. payload : {
    - questionId: string;
  - b. response : QuestionDTO
4. (GET) /question
  - a. payload : null
  - b. response : QuestionDTO
5. (GET) /modelAnswer
  - a. payload : {
    - questionId: string;
  - b. response : AnswerDTO