수집된 데이터 및 데이터 전처리 문서

1. 데이터 설명 및 구성

1.1. 데이터 및 필드 설명

구분	필드명	데이터 타입	설명	예시
DATA-001 ~ DATA-004	id	string	고유 데이터 식별자_숫자	card_012
	title	string	FAQ 질문	Q: 재발급 신청은 본인만 할 수 있나요?
	content	string	FAQ 답변	A: 나라사랑카드 재발급을 포함한 모든 금융업무는 본인만 가능합니다.(부모님, 친구 등 대리인 불가) 현역병사는 병영생활에 불편이 없도록 분실, 파손 등에 주의하여 나라사랑카드를 사용하기 바랍니다.
	text	string	FAQ 질문 + FAQ 답변	Q: 재발급 신청은 본인만 할 수 있나요? | A: 나라사랑카드 재발급을 포함한 모든 금융업무는 본인만 가능합니다.(부모님, 친구 등 대리인 불가) 현역병사는 병영생활에 불편이 없도록 분실, 파손 등에 주의하여 나라사랑카드를 사용하기 바랍니다.
	card_name	string	카드 이름	나라사랑카드
	category	string	카테고리	발급/신청
DATA-005 ~ DATA-006	id	string	고유 데이터 식별자_숫자	card_100
	title	string	제목	놀이공원 할인
	content	string	본문	에버랜드, 롯데월드 현장예매 50% 환급할인 건당 이용금액 3만원 이상 시 건당 최대 이용금액 5만원까지 할인 적용(최대 할인액 25,000원)
	text	string	제목 + 본문	놀이공원 할인 에버랜드, 롯데월드 현장예매 50% 환급할인 건당 이용금액 3만원 이상 시 건당 최대 이용금액 5만원까지 할인 적용(최대 할인액 25,000원)
	card_name	string	카드이름	나라사랑체크카드

구분	필드명	데이터 타입	설명	예시
	category	string	카테고리	혜택/할인
DATA-007 ~ DATA-010	id	string	고유 데이터 식별자_숫자	guide_14
	title	string	제목	상환방법 안내
	content	string	본문	1. 원리금 균등: 대출기간 동안 매월 같은 금액(원금+이자)을 납부하는 방식입니다. 매월 납부금액이 같기 때문에 지출 계획을 세우기 좋습니다.\n2. 원금 균등: 대출기간 동안 원금을 매달 같은 금액으로 납부하는 방식입니다. 만기 시점과 가까워질수록 상환해야 할 금액이 줄어듭니다.\n3. 거치 후 원리금 균등: 일정기간 동안 이자만 갚고 싶다면 거치기간 동안 이자만 납입하고 그 이후로는 매월 같은 금액(원금+이자)을 납부하는 방식입니다.\n4. 만기일시: 이자만 내고 원금은 나중에 갚고 싶다면 매달 이자만 상환하고, 만기일에 원금을 상환합니다. 만기 시점에 일시 상환 또는 대출기간을 연장할 수 있습니다.
	text	string	제목 + 본문	상환방법 안내 1. 원리금 균등: 대출기간 동안 매월 같은 금액(원금+이자)을 납부하는 방식입니다. 매월 납부금액이 같기 때문에 지출 계획을 세우기 좋습니다.\n2. 원금 균등: 대출기간 동안 원금을 매달 같은 금액으로 납부하는 방식입니다. 만기 시점과 가까워질수록 상환해야 할 금액이 줄어듭니다.\n3. 거치 후 원리금 균등: 일정기간 동안 이자만 갚고 싶다면 거치기간 동안 이자만 납입하고 그 이후로는 매월 같은 금액(원금+이자)을 납부하는 방식입니다.\n4. 만기일시: 이자만 내고 원금은 나중에 갚고 싶다면 매달 이자만 상환하고, 만기일에 원금을 상환합니다. 만기 시점에 일시 상환 또는 대출기간을 연장할 수 있습니다.
	category1	string	대분류	금융안내
	category2	string	중분류	카드대금 납부
DATA-011 ~ DATA-012	id	string	고유 데이터 식별자_숫자	공지사항_15
	tag	string	태그(이벤트/긴급/시스템/피해)	[이벤트]

구분	필드명	데이터 타입	설명	예시
	title	string	제목	아시아나항공 마일리지 전환 종료 안내
	content	string	본문	아시아나항공 마일리지 전환 서비스가 2026.6.30(화)까지만 운영될 예정입니다.\n\n신청 마감일 \n\n2026.6.30(화)\n\n* 자세한 전환방법 및 대상카드는 삼성카드 홈페이지 또는 모니모 앱 '마이삼성 → 카드 → 포인트 조회 → 포인트 사용 → 아멕스 제휴사 마일리지 전환'에서 확인\n\n삼성카드 대표 전화 1588-8700
	date	string	날짜	2025.12.30
DATA-013	id	string	고유 식별자	hana_consultation_{source_id}
	consultation_id	string	Frontend API 호환 ID	CS-HANA-{source_id}
	document_type	string	문서 타입	consultation_transcript
	title	string	상담 제목	카테고리 기반 자동 생성 예: 도난/분실 신청/해제 상담
	content	string	상담 대화 내용	상담사: 상담원 [상담원명#1]입니다.\n손님: 네, 저 [카드사명#1] ...
	source_id	string	원본 CSV ID	20593 CSV의 source_id 컬럼
	category	string	상담 카테고리	도난/분실 신청/해제
	keywords	list	키워드 리스트	카테고리 기반 + 빈출 명사 추출, VectorDB 검색에 활용 ["도난/분실 신청/해제", "카드", "결제", "발급", "이용"]
	slot_types	list	개인정보 태그 타입 목록	Entity Tracking 정보, 중복 제거된 고유 타입만 포함 ["상담원명", "카드사명", "고객명", "생년월일", "전화번호", "은행명", "카드번호_구성요소", "날짜"]
	scenario_tags	list	시나리오 태그 목록	규칙 기반 추출 "scenario_tags": [

구분	필드명	데이터 타입	설명	예시
				"카드교체발급", "카드유효기간만료"]
	summary	string	AI 요약	null
	created_at	string	생성 시점	ISO 8601 형식 예: `2026-01-07T10:20:15.099495`

- 데이터 수집 기간: 2026.01.02-2026.01.05
- 전체 수집 데이터 건수
  - 카드 정보 Index 구축용 : 512건
  - 카드사 이용 안내 Index 구축용 : 281건
  - 상담 사례 Index 구축용 : 6533건
  - 공지사항 RDB 구축용: 52건

---

2. 데이터 수집 및 활용의 적법성 검토

구분	상업 이용 가능	학습 사용 허용	크롤링 이용약관 준수	개인정보 보호	비고
DATA-001	X	O	O	O	
DATA-002	X	O	O	O	
DATA-003	O	O	O	O	출처 표시 후 사용 가능
DATA-004	X	O	O	O	
DATA-005	X	O	O	O	국민카드 - 카드사의 서비스 정보를 이용하여 얻은 정보를 카드사의 사전 승낙 없이 복제 또는 유통시키거나 상업적으로 이용하는 행위만 불가능
DATA-006	X	O	O	O	
DATA-007	O	O	O	O	
DATA-008	X	O	O	O	
DATA-009	X	O	O	O	

- **저작권 준수**

  해당 데이터의 이용약관을 검토하여 상업적 이용 및 2차 가공 가능 여부 확인
- **크롤링 이용약관 준수**

Robots.txt 규정에 따라 서버에 부하를 주지 않는 방식으로 크롤링 수행

- **개인정보 보호**

  수집 과정에서 개인 식별 정보(이름, 연락처 등)는 비식별 처리

---

3. 수집 자동화

- 수집 항목 및 품질 기준: 수집 데이터 필드와 정상 상태(기준) 작성

  수집 데이터 필드

  - `id`: String
    - 카테고리명 + 숫자 조합으로 생성
    - 전체 데이터셋 내에서 유일해야 함
  - `text`: String
    - 질문/답변 또는 섹션 내용 전체를 포함
    - 공백 및 특수문자 정리된 텍스트
  - `title`: String
    - 섹션 제목 또는 FAQ 질문 등 대표 타이틀
    - 빈 문자열 불가
  - `category`: String
    - 미리 정의된 분류값(예: 카드명, 혜택 섹션 등) 중 하나

  정상 상태(유효 데이터 기준)

  - `id` 는 중복되지 않아야 함
  - `text`, `title` 은 빈 문자열이 아니어야 함
  - `category` 는 사전에 정의된 허용 목록 내 값이어야 함
  - 전체 결과는 JSON 배열 형태를 유지해야 함

- 수집 방식 및 도구:

  데이터는 대상 페이지의 특성에 따라 동적 페이지와 정적 페이지를 구분하여 수집한다.

  동적 페이지의 경우 Selenium WebDriver(Chromedriver)를 사용해 카테고리 탭, 목록, 상세 페이지를 순차적으로 클릭하여 콘텐츠를 로딩한 뒤, 로딩된 HTML에서 FAQ, 안내 문구, 혜택 설명 등 페이지 내 주요 텍스트 콘텐츠를 추출한다.

  정적 페이지는 requests와 BeautifulSoup을 이용해 직접 HTTP 요청 후 HTML을 파싱하여 데이터를 수집한다.

- 데이터 수집 자동화 코드:

```
# Selenium common_module
def get_driver(chrome_path="chromedriver.exe"):
```

```
    service = Service(chrome_path)
    options = webdriver.ChromeOptions()
    driver = webdriver.Chrome(service=service, options=options)
    return driver
```

```
# Selenium
from selenium import webdriver
from selenium.webdriver.common.by import By
import json
import time
from common_module import get_driver, clean_text

driver = get_driver()
categories = ['회원정보', '적립', '지급', '카드', '이용방법']
data = []

try:
    url = "https://korea-pass.kr/notice/faqList.do"
    driver.get(url)
    time.sleep(1)

    for i in range(2, 7):
        print(f'===== {i}번째 카테고리 크롤링 =====')
        category = categories[i-2]

        tab_xpath = f'//*[@id="tab0{i}"]'
        driver.find_element(By.XPATH, tab_xpath).click()
        time.sleep(1)

        lis = driver.find_elements(By.CSS_SELECTOR, '#faqDiv > li')
        count = len(lis)
        print(f"리스트 개수 : {count}개")

        for j in range(1, count + 1):
            try:
                driver.find_element(By.XPATH, tab_xpath).click()
                time.sleep(1)

                question_xpath = f'//*[@id="faqDiv"]/li[{j}]/a'
                btn = driver.find_element(By.XPATH, question_xpath)

                # 질문 추출
                q_text = btn.find_element(By.TAG_NAME, 'h4').text
                q_text = clean_text(q_text)
                print(q_text)
```

```
            # 상세 페이지 진입
            driver.execute_script("arguments[0].click();", btn)
            time.sleep(1)

            # 답변 추출
            answer_element = driver.find_element(By.CSS_SELECTOR, 'section p')
            a_text = answer_element.text
            a_text = clean_text(a_text)
            print(a_text)

            # 데이터 저장
            data_entry = {
                "category_index": category,
                "id": j,
                "question": q_text,
                "answer": a_text
            }
            data.append(data_entry)

            driver.back()
            time.sleep(2)

        except Exception as e:
            print(f"{j}번 처리 중 오류 발생 : {e}")
            continue

    print("===== 데이터 저장 중 =====")
    with open('./../../data/special_card/kpass_faq.json', 'w', encoding='utf-8') as f:
        json.dump(data, f, ensure_ascii=False, indent=4)

except Exception as e:
    print(f"오류 발생 : {e}")

finally:
    driver.quit()
```

- 데이터 유효성을 검증하는 방법: 검증 과정에 대한 설명(오류 발생 시 예외 처리 전략)

1. FAQ 항목 로딩 검증

- 카테고리 탭 클릭
  - 각 카테고리 탭 클릭 후 FAQ 리스트 요소(`#faqDiv > li`)가 정상적으로 로드되는지 확인
  - 리스트 개수(`len(lis)`)가 0인 경우 해당 카테고리는 수집 대상에서 제외
  - `print(f'===== {i}번째 카테고리 크롤링 =====')` 프린트 문에 현재 카테고리를 출력하여 진행상황 확인

2. 리스트 로딩 검증

- FAQ 리스트 클릭
  - 각 리스트 클릭 후 FAQ 리스트 상세 내용이 정상적으로 로드되는지 확인
- 예외 처리 전략
  - 리스트 탐색 실패 시 해당 FAQ 항목만 skip
  - 리스트 탐색 실패 시 `try-except`로 예외 처리
  - 실패한 리스트 번호와 오류 로그 출력 후 다음 항목 처리 (`continue`)
    `print(f"{j}번 처리 중 오류 발생 : {e}")`
  - 크롤링 전체 중단 없이 부분 실패 허용

3. 질문 데이터 유효성 검증

- 질문 텍스트 추출 검증
  - `<h4>` 태그에서 질문 텍스트 추출
  - 정제 후 질문 문자열이 비어 있는 경우 해당 항목 제외
  - `print(q_text)` 프린트문에 질문을 출력하여 확인

3. 답변 데이터 유효성 검증

- 답변 텍스트 추출 검증
  - `section p` 선택자를 통해 답변 텍스트 추출
  - `print(a_text)` 프린트문에 답변을 출력하여 확인
  - `driver.back()`으로 페이지 복귀 후 다음 질문으로 진행

4. 전체 크롤링 안정성 검증

- 페이지 이동 검증
  - FAQ 상세 페이지 → 목록 페이지로 정상 복귀 여부 확인
  - 매 항목마다 카테고리 탭을 다시 클릭하여 DOM 상태 초기화
- 예외 처리 전략
  - 페이지 이동 중 오류 발생 시 해당 항목만 제외하고 크롤링 지속

    - 모든 예외는 로그로 기록하며, 최종적으로 수집 가능한 데이터만 JSON 파일로 저장
      `print(f"오류 발생 : {e}")`

5. 저장 단계 검증

- 모든 수집 데이터는 JSON 형식으로 저장
- `ensure_ascii=False` 옵션으로 한글 데이터 손실 방지
- 크롤링 종료 시 `driver.quit()` 호출로 리소스 정리

- 수집 데이터 건수:
    - 카드 정보 Index 구축용 : 512건
    - 카드사 이용 안내 Index 구축용 : 281건
    - 상담 사례 Index 구축용 : 6533건
    - 공지사항 RDB 구축용: 52건
- 자동화 여부 및 주기:
    - 크롤링 스크립트를 통해 자동 수집 진행
    - 초기 구축 단계에서 1회성 수집

---

4. 전처리 프로세스 개요

- 전체 흐름도:
  ① 수집→ ② 데이터 전처리 → ③ 정규화 → ④ 데이터 청킹 → ⑤ 데이터 형식 통일
- 전처리 파이프라인 요약:

단계	목적	수행 작업	사용 도구/라이브러리
수집	다양한 카드·공공 FAQ 및 문서 데이터 확보	-Selenium 기반 웹 FAQ 크롤링 - requests 기반 정적 HTML 수집 - PDF 문서 수집	Selenium, WebDriver, requests, BeautifulSoup
데이터 전처리	비정형 데이터를 구조화된 텍스트로 변환	- HTML에서 FAQ 질의/응답 추출 - PDF 다단 레이아웃 분리(1~4단) - 페이지·섹션 단위 텍스트 추출	BeautifulSoup, pdfplumber
정규화	노이즈 제거 및 텍스트 품질 개선	- 특수문자, 불필요 문구 제거 - 개행/탭/중복 문장 정리 - OCR 오타·URL 공백·반복 블록 제거	re
데이터 청킹	RAG에 적합한 입력 단위 생성	- 문장/의미 단위 청킹 - 길이 제한 기반 분할 - 중복 청크 제거	custom chunking logic (Python)

데이터 형식 통일	벡터화·RAG 파이프라인 통합	- 서로 다른 소스(JSON/FAQ/PDF) 통합 - id / text / metadata 스키마로 변환 - 최종 벡터 적재용 JSON 생성	custom Python script

- 도식화: