RAG · Reranker · Prompt API 명세서 (최종 검수판)

> **모든 엔드포인트에 대해**
>
> - **기본 정보** – Method · URL · 설명
> - **요청 파라미터** – 필수 여부·타입·설명 (Query / Body 구분)
> - **요청 예시** – 완전한 `curl` 명령
> - **응답 파라미터** – 필드·타입·설명
> - **성공 응답 예시**
> - **실패 응답 예시** (가능한 경우)

목차

> **모든 URL** 는 `http://localhost` 기준이며, 실제 배포 시 호스트/포트를 맞춰 수정하세요.

1. /rag/

기본 정보

항목	내용
Method	**GET**
URL	`/rag/`
설명	RAG FastCGI 서비스 헬스 체크

요청 파라미터

없음

요청 예시

```
curl -X GET http://localhost/rag/
```

응답 파라미터

필드	Type	설명
message	String	상태 메시지

성공 응답 예시

```
{ "message": "Hello, FastCGI is working!" }
```

---

2. /rag/insert

기본 정보

항목	내용
Method	**POST**
URL	`/rag/insert`
Content-Type	`application/json`
설명	문서를 Milvus 컬렉션에 삽입

요청 파라미터 (Body)

필드	필수	Type	설명
documents	Y	Array	삽입할 문서 배열
ignore	N	Boolean	중복 문서 무시 여부 (기본값: true)

각 문서 객체의 구조:

필드	필수	Type	설명
domain	Y	String	컬렉션 이름 (예: `news`)
title	Y	String	문서 제목
author	Y	String	작성자/기관
text	Y	String	본문
info	N	Object	`{ press_num, url }`
tags	Y	Object	`{ date(YYYYMMDD), user }`

요청 예시

```
curl -X POST http://localhost/rag/insert \
  -H "Content-Type: application/json" \
  -d '{
    "documents": [
      {
        "domain": "news",
        "title": "메타버스 뉴스",
        "author": "삼성전자",
        "text": "메타버스는 비대면 시대 뜨거운 화두로 떠올랐다...",
        "info": { "press_num": "비즈니스 워치", "url": "http://example.com/news/1"
},
        "tags": { "date": "20240315", "user": "admin" }
      }
    ],
    "ignore": true
  }'
```

응답 파라미터

필드	Type	설명
status	String	전체 처리 상태 ("success", "partial_success", "partial_error", "error")
message	String	처리 결과 메시지
status_counts	Object	상태별 처리 건수 (`success`, `skipped`, `error`)
results	Array	각 문서별 처리 결과

성공 응답 예시

```
{
  "status": "success",
  "message": "총 3개 문서 중 2개 성공, 1개 건너뜀",
```

```
  "status_counts": {
    "success": 2,
    "skipped": 1,
    "error": 0
  },
  "results": [
    {
      "status": "success",
      "message": "문서가 성공적으로 저장되었습니다.",
      "doc_id": "1234567890abcdef...",
      "raw_doc_id": "20240315-메타버스 뉴스-삼성전자",
      "domain": "news",
      "title": "메타버스 뉴스"
    },
    {
      "status": "skipped",
      "message": "이미 존재하는 문서로 건너뛰었습니다.",
      "doc_id": "abcdef1234567890...",
      "raw_doc_id": "20240315-AI 뉴스-LG전자",
      "domain": "news",
      "title": "AI 뉴스"
    }
  ]
}
```


실패 응답 예시


```
{
  "status": "error",
  "message": "요청 본문이 비어있습니다.",
  "error_code": "F000001"
}
```


---

3. /rag/search

기본 정보

항목	내용
Method	**GET**
URL	`/rag/search`
설명	Milvus에서 유사 문서 검색

요청 파라미터 (Query)

이름	필수	Type	기본	설명

이름	필수	Type	기본	설명
query_text	Y	String	–	검색어
top_k	N	Integer	5	검색 결과 수
domain	N	String[]	–	도메인 필터 (복수 지정 가능)
author	N	String	–	작성자 필터
start_date	N	String	–	시작일 `YYYYMMDD`
end_date	N	String	–	종료일 `YYYYMMDD`
title	N	String	–	제목 검색
info_filter	N	String	–	`info` JSON 문자열
tags_filter	N	String	–	`tags` JSON 문자열

요청 예시 (단일 도메인)

```
curl -G http://localhost/rag/search \
  --data-urlencode "query_text=메타버스" \
  --data-urlencode "top_k=5" \
  --data-urlencode "domain=news" \
  --data-urlencode "start_date=20240301" \
  --data-urlencode "end_date=20240315"
```

요청 예시 (복수 도메인)

```
curl -G http://localhost/rag/search \
  --data-urlencode "query_text=메타버스" \
  --data-urlencode "top_k=5" \
  --data-urlencode "domain=news" \
  --data-urlencode "domain=test" \
  --data-urlencode "start_date=20240301" \
  --data-urlencode "end_date=20240315"
```

응답 파라미터

필드	Type	설명
result_code	String	`F000000` 성공 등
message	String	결과 메시지
search_params	Object	적용된 검색 파라미터
total_results	Integer	전체 검색 결과 수

필드	Type	설명
returned_results	Integer	반환된 결과 수
search_result	Array	검색 결과 배열

각 검색 결과 객체의 구조:

필드	Type	설명
doc_id	String	문서 ID
passage_id	Integer	패시지 ID
domain	String	도메인
title	String	제목
author	String	작성자
text	String	본문 내용
info	Object	추가 정보
tags	Object	태그 정보
score	Float	유사도 점수

성공 응답 예시


```
{
  "result_code": "F000000",
  "message": "검색이 성공적으로 완료되었습니다.",
  "search_params": {
    "query_text": "메타버스",
    "top_k": 5,
    "domains": ["news", "blog"],
    "filters": {
      "date_range": {
        "start": "20240301",
        "end": "20240315"
      }
    }
  },
  "total_results": 10,
  "returned_results": 5,
  "search_result": [
    {
      "doc_id": "109f405744d2f1e0eccb880c70c6c6e9...",
      "passage_id": 1,
      "domain": "news",
      "title": "메타버스 뉴스",
      "author": "삼성전자",
      "text": "메타버스는 비대면 시대 뜨거운 화두로 떠올랐다...",
```

```
        "info": {
          "press_num": "비즈니스 워치",
          "url": "http://example.com/news/1"
        },
        "tags": {
          "date": "20240315",
          "user": "admin"
        },
        "score": 0.95
      }
    ]
}
```

실패 응답 예시

```
{
  "result_code": "F000001",
  "message": "검색어(query_text)는 필수 입력값입니다.",
  "search_result": null
}
```

---

4. /rag/delete

기본 정보

항목	내용
Method	**DELETE**
URL	`/rag/delete`
설명	문서 삭제

요청 파라미터 (Query)

이름	필수	Type	설명
doc_id	Y	String	문서 ID
domain	Y	String	도메인

요청 예시

```
curl -X DELETE "http://localhost/rag/delete?doc_id=20240315-메타버스-뉴스
&domain=news"
```

응답 파라미터

필드	Type	설명
status	String	"received"

성공 응답 예시

```
{ "status": "received" }
```

실패 응답 예시 (작성자 누락)

```
{
  "error": "author is required",
  "message": "작성자(author)는 필수 입력값입니다."
}
```

---

5. /rag/document

기본 정보

항목	내용
Method	**GET**
URL	/rag/document
설명	특정 문서 혹은 패시지 조회

요청 파라미터 (Query)

이름	필수	Type	설명
doc_id	Y	String	문서 ID
passage_id	N	String/Int	패시지 ID

요청 예시 (문서 전체)

```
curl -G http://localhost/rag/document --data-urlencode "doc_id=20240315-메타버스-뉴스"
```

응답 파라미터 (문서 전체)

필드	Type	설명
doc_id	String	문서 ID

필드	Type	설명
domain	String	도메인
title	String	제목
info	Object	추가 정보
tags	Object	태그
passages	Array	[{ passage_id, text, position }]

성공 응답 예시 (문서 전체)

```
{
  "doc_id": "20240315-메타버스-뉴스",
  "domain": "news",
  "title": "메타버스 뉴스",
  "info": { "press_num": "비즈니스 워치", "url": "http://example.com/news/1" },
  "tags": { "date": "20240315", "user": "admin" },
  "passages": [
    { "passage_id": 1, "text": "메타버스는...", "position": 1 }
  ]
}
```

실패 응답 예시 (문서 없음)

```
{
  "error": "Document not found",
  "doc_id": "20230101-존재-하지-않음",
  "message": "요청하신 문서를 찾을 수 없습니다."
}
```

---

6. /rag/data/show

기본 정보

항목	내용
Method	**GET**
URL	`/rag/data/show`

요청 파라미터 (Query)

이름	필수	Type	설명
collection_name	Y	String	컬렉션 이름

요청 예시

```
curl -G http://localhost/rag/data/show --data-urlencode "collection_name=news"
```

응답 파라미터 (정상)

필드	Type	설명
schema	Object	컬렉션 스키마
partition_names	Array	파티션 목록
partition_nums	Object	파티션 → 엔티티 수

성공 응답 예시

```
{
  "schema": { "fields": [ { "name": "doc_id", "type": "VARCHAR" } ] },
  "partition_names": [ "p1" ],
  "partition_nums": { "p1": 100 }
}
```

실패 응답 예시 (없는 컬렉션 – **HTTP 200**)

```
{
  "error": "유효한 Collection 이름을 입력해야 합니다.",
  "collection list": ["news", "description"]
}
```

---

7. /reranker/health

기본 정보

항목	내용
Method	**GET**
URL	`/reranker/health`
설명	Reranker 헬스 체크

요청 파라미터

없음

요청 예시

```
curl -X GET http://localhost/reranker/health
```

응답 파라미터

필드	Type	설명
status	String	"ok"
service	String	"reranker"

성공 응답 예시

```
{ "status": "ok", "service": "reranker" }
```

---

8. /reranker/enhanced-search

기본 정보

항목	내용
Method	**GET**
URL	`/reranker/enhanced-search`
설명	RAG 검색 결과를 가져와 Reranker로 재랭킹 후 반환

요청 파라미터 (Query)

이름	필수	Type	기본	설명
query_text	Y	String	–	검색어
top_k	N	Integer	5	최종 결과 수
raw_results	N	Integer	20	RAG에서 가져올 초기 결과 수
domain	N	String	–	도메인 필터
author	N	String	–	작성자 필터
start_date	N	String	–	시작일 `YYYYMMDD`
end_date	N	String	–	종료일 `YYYYMMDD`
title	N	String	–	제목 검색

요청 예시

```
curl -G http://localhost/reranker/enhanced-search \
  --data-urlencode "query_text=메타버스 최신 동향" \
  --data-urlencode "top_k=5" \
  --data-urlencode "raw_results=20" \
  --data-urlencode "domain=news"
```

응답 파라미터

필드	Type	설명
result_code	String	`F000000` 성공 등
message	String	결과 메시지
search_params	Object	실제 적용 파라미터
search_result	Array	재랭킹된 결과 목록

성공 응답 예시

```
{
  "result_code": "F000000",
  "message": "검색 및 재랭킹이 성공적으로 완료되었습니다.",
  "search_params": {
    "query_text": "메타버스 최신 동향",
    "top_k": 5,
    "raw_results": 20,
    "filters": { "domain": "news" }
  },
  "search_result": [
    {
      "doc_id": "20240315-메타버스-뉴스",
      "text": "메타버스는 비대면 시대 뜨거운 화두로 떠올랐다...",
      "score": 0.98,
      "title": "메타버스 뉴스",
      "author": "삼성전자"
    }
  ]
}
```

실패 응답 예시 (검색어 누락)

```
{
  "result_code": "F000001",
  "message": "검색어(query_text)는 필수 입력값입니다.",
  "search_result": null
}
```

9. /reranker/rerank

기본 정보

항목	내용
Method	**POST**
URL	`/reranker/rerank`
Content-Type	`application/json`
설명	단일 쿼리와 결과 목록을 재랭킹

요청 파라미터

Query: `top_k`(N)

Body (SearchResultModel)

필드	필수	Type	설명
query	Y	String	쿼리
results	Y	Array	문서 배열

요청 예시

```
curl -X POST "http://localhost/reranker/rerank?top_k=3" \
  -H "Content-Type: application/json" \
  -d '{
    "query": "메타버스 최신 동향",
    "results": [
      {
        "passage_id": 0,
        "doc_id": "20240315-메타버스-뉴스",
        "text": "메타버스는 비대면 시대 뜨거운 화두로 떠올랐다...",
        "score": 0.95,
        "metadata": {
          "title": "메타버스 뉴스",
          "author": "삼성전자",
          "tags": { "date": "20240315" }
        }
      }
    ]
  }'
```

응답 파라미터

SearchResultModel + `reranked: true`

성공 응답 예시

```
{
  "query": "메타버스 최신 동향",
  "results": [
    {
      "passage_id": 0,
      "doc_id": "20240315-메타버스-뉴스",
      "text": "메타버스는...",
      "score": 0.98
    }
  ],
  "total": 1,
  "reranked": true
}
```

실패 응답 예시 (Body 누락)

```
{ "error": "No JSON data provided" }
```

---

10. /reranker/batch_rerank

기본 정보

항목	내용
Method	**POST**
URL	`/reranker/batch_rerank`
Content-Type	`application/json`
설명	여러 쿼리를 한 번에 재랭킹

요청 파라미터

Query: `top_k`(N)

Body: `[ SearchResultModel, ... ]`

요청 예시

```
curl -X POST "http://localhost/reranker/batch_rerank?top_k=5" \
  -H "Content-Type: application/json" \
  -d '[
    {
      "query": "메타버스 최신 동향",
```

```
      "results": [ { "doc_id": "20240315-메타버스-뉴스", "text": "..." } ]
    },
    {
      "query": "가상현실 시장 전망",
      "results": [ { "doc_id": "20240312-VR-뉴스", "text": "..." } ]
    }
  ]'
```

응답 파라미터

배열 – 각 결과에 `total`, `reranked`

성공 응답 예시

```
[
  { "query": "메타버스 최신 동향", "total": 1, "reranked": true },
  { "query": "가상현실 시장 전망", "total": 1, "reranked": true }
]
```

실패 응답 예시

```
{ "error": "Batch reranking failed: ..." }
```

---

11. /prompt/health

기본 정보

항목	내용
Method	**GET**
URL	`/prompt/health`
설명	Prompt-Backend 헬스 체크

요청 파라미터

없음

요청 예시

```
curl -X GET http://localhost/prompt/health
```

성공 응답 예시

```
{
  "status": "ok",
  "timestamp": "2025-04-22T12:34:56Z",
  "service": "prompt-backend"
}
```

---

12. /prompt/summarize

기본 정보

항목	내용
Method	**POST**
URL	`/prompt/summarize`
Content-Type	`application/json`
설명	검색 → 재랭킹 → LLM 요약

요청 파라미터 (Body)

이름	필수	Type	설명
query	Y	String	요약할 쿼리
domain/author/start_date/end_date	N	String	필터

요청 예시

```
curl -X POST http://localhost/prompt/summarize \
  -H "Content-Type: application/json" \
  -d '{ "query": "메타버스 최신 동향", "domain": "news" }'
```

성공 응답 예시

```
{
  "query": "메타버스 최신 동향",
  "summary": "최근 메타버스 시장은 급성장 중...",
  "documents_count": 5,
  "prompt_length": 1024
}
```

실패 응답 예시 (쿼리 누락)

```
{ "error": "쿼리가 필요합니다" }
```

---

13. /prompt/chat

기본 정보

항목	내용
Method	**POST**
URL	`/prompt/chat`
Content-Type	`application/json`
설명	간단한 Q&A 챗봇

요청 파라미터 (Body)

이름	필수	Type	설명
query	Y	String	질문 내용
model	N	String	사용할 모델 (기본값: 서버 설정)

요청 예시

```
curl -X POST http://localhost/prompt/chat \
  -H "Content-Type: application/json" \
  -d '{ "query": "메타버스란 무엇인가요?", "model": "llama2" }'
```

성공 응답 예시

```
{
  "query": "메타버스란 무엇인가요?",
  "model": "llama2",
  "response": "메타버스는 가상과 현실이 융합된 디지털 공간입니다..."
}
```

실패 응답 예시

```
{ "error": "질문이 필요합니다" }
```

---

14. /prompt/models

기본 정보

항목	내용
Method	**GET**
URL	`/prompt/models`
설명	사용 가능한 LLM 모델 목록

요청 파라미터

없음

요청 예시

```
curl -X GET http://localhost/prompt/models
```

성공 응답 예시

```
{
  "models": ["llama2", "mistral", "gemma"],
  "default_model": "llama2",
  "total": 3
}
```

실패 응답 예시

```
{ "error": "모델 목록을 가져오는 중 오류가 발생했습니다" }
```

---

15. /vision/health

기본 정보

항목	내용
Method	**GET**
URL	`/vision/health`
설명	Vision 서비스 상태 확인

요청 파라미터

없음

요청 예시

```
curl -X GET http://localhost/vision/health
```

응답 파라미터

필드	Type	설명
status	String	서비스 상태
service	String	"vision"
default_model	String	기본 사용 모델

성공 응답 예시

```
{
  "status": "healthy",
  "service": "vision",
  "default_model": "llama:3.2-11b-vision"
}
```

---

16. /vision/analyze

기본 정보

항목	내용
Method	**POST**
URL	`/vision/analyze`
Content-Type	`application/json`
설명	이미지 분석 수행

요청 파라미터 (Body)

이름	필수	Type	설명
url	Y	String	분석할 이미지 URL
prompt	N	String	분석 프롬프트 (기본값: "이 이미지에 대해 설명해주세요")
model	N	String	사용할 모델 (기본값: llama:3.2-11b-vision)

요청 예시

```
curl -X POST http://localhost/vision/analyze \
  -H "Content-Type: application/json" \
  -d '{
    "url": "https://example.com/image.jpg",
    "prompt": "이 이미지에 대해 설명해주세요",
    "model": "llama:3.2-11b-vision"
  }'
```

응답 파라미터

필드	Type	설명
description	String	이미지 분석 결과
image_url	String	분석된 이미지 URL
model	String	사용된 모델
total_duration	Number	총 처리 시간 (ms)
load_duration	Number	모델 로딩 시간 (ms)
prompt_eval_count	Number	프롬프트 평가 횟수
eval_count	Number	총 평가 횟수
eval_duration	Number	평가 소요 시간 (ms)

성공 응답 예시

```
{
  "description": "이 이미지는 푸른 하늘을 배경으로 한 현대적인 도시 풍경을 보여줍니다...",
  "image_url": "https://example.com/image.jpg",
  "model": "llama:3.2-11b-vision",
  "total_duration": 2345,
  "load_duration": 123,
  "prompt_eval_count": 50,
  "eval_count": 100,
  "eval_duration": 2000
}
```

실패 응답 예시

```
{
  "error": "이미지 분석에 실패했습니다"
}
```

> **문서 업데이트 완료** – 필요 시 추가 요청 주세요.