

[Strategy] Classic Hero: 슷폼 고전 콘텐츠 추천 및 수급 전략

본 문서는 Gutendex API를 통해 수집 가능한 데이터 중, 가공을 최소화하고도 10분 학습(Daily 10-Minute Classic)에 즉시 활용 가능한 최적의 콘텐츠 리스트와 필터링 전략을 제안합니다.

1. 유형별 슷폼 콘텐츠 추천 (수준별)

원문 자체가 짧아 "내용을 자르지 않아도" 학습용으로 적합한 카테고리입니다.

1.1 Beginner: Fables & Fairy Tales (우화 및 동화)

입문 단계에서는 문장이 단순하고 교훈이 명확한 콘텐츠가 유리합니다.

- **Aesop's Fables (이솝 우화):** 한 에피소드당 100~300단어 내외로 가장 완벽한 슷폼입니다.
- **Beatrix Potter (피터 래빗 시리즈):** 단어 선택이 고급스러우면서도 분량이 매우 짧습니다.
- **Grimm's Fairy Tales:** 일부 단편들은 5분 내외 독해가 가능합니다.

1.2 Intermediate: Short Stories & Letters (단편 소설 및 편지)

서사 구조를 갖추면서도 집중력을 유지할 수 있는 분량입니다.

- **O. Henry (오 헨리 단편선):** '마지막 잎새' 등 반전이 있는 단편들은 영미권 학습의 정석입니다.
- **Oscar Wilde (오스카 와일드 동화):** '행복한 왕자' 등은 문체가 유려하여 필사 학습에 좋습니다.
- **Benjamin Franklin's Letters:** 실용 영어의 시초로 불리며, 영웅 대화(Talk to the Hero)의 소스로 매우 훌륭합니다.

1.3 Advanced: Meditations & Speeches (명상록 및 연설문)

철학적 깊이가 있으나 호흡이 짧은 문장 중심의 콘텐츠입니다.

- **Marcus Aurelius - Meditations (명상록):** 각 '장(Book)'이 아닌 '구절(Paragraph)' 단위로 나뉘어 있어 조각 학습에 최적입니다.
- **Abraham Lincoln's Speeches:** 'Gettysburg Address'처럼 짧지만 강력한 문장력을 배울 수 있습니다.
- **Ralph Waldo Emerson - Nature/Essays:** 에세이 중 특정 단락(Paragraph)들이 독립적인 주제를 가집니다.

2. Gutendex API를 이용한 최적의 필터링 기법

Gutendex API 자체에는 '단어 수' 필터가 없으므로, PM으로서 다음 두 가지 우회 전략을 제안합니다.

2.1 검색 태그(Subject) 활용

API 호출 시 `topic` 이나 `search` 파라미터에 아래 키워드를 조합하여 슷폼 가능성이 높은 도서를 우선 순위로 가져옵니다.

- `topic=short stories`
- `topic=fables`

- `topic=essays`
- `topic=aphorisms` (격언/금언)

2.2 Metadata 기반 로컬 필터링 (Python 로직)

API 응답 결과에서 도서의 '파일 크기'나 '형식'을 통해 분량을 유추할 수 있습니다.

1. **Format Check:** `application/epub+zip` 이나 `text/plain` 파일의 용량이 **50KB~150KB** 사이인 도서를 타겟팅합니다. (수백 KB가 넘어가는 도서는 대개 장편 소설입니다.)
2. **Author-based Search:** 위에서 언급한 'O. Henry'나 'Aesop'과 같이 단편 위주 저자의 `author_year_start` 를 기준으로 필터링합니다.

3. 영웅 대화(Talk to the Hero)를 위한 콘텐츠 페어링

콘텐츠가 너무 길지 않아야 영웅과의 대화도 구체적일 수 있습니다.

- **추천 페어링:** - 콘텐츠: "The Way to Wealth" (벤자민 프랭클린의 조언)
 - 대화 영웅: **Benjamin Franklin**
 - 이점: 원문이 짧고 실용적인 충고 위주라 AI가 "당신은 방금 제 조언 중 '시간은 돈이다'라는 부분을 읽으셨군요"라고 대화를 이끌기 매우 쉽습니다.

4. PM의 결론 및 제언

학습 서비스에서 "너무 긴 내용"은 사용자의 이탈(Churn)을 초래합니다.

1. **Curated DB 구축:** API에서 실시간으로 가져오기보다, 위 추천 리스트를 기반으로 약 100~200개의 '검증된 솜품 원문'을 먼저 DB화(PostgreSQL) 하는 것을 권장합니다.
2. **AI Chunking Node:** 그럼에도 불구하고 긴 텍스트를 쓸 경우, 집사님께서 공부하시는 `**LangChain` 의 `RecursiveCharacterTextSplitter` **를 사용하여 의미 단위로 쪼개고, 각 조각에 '소제목'을 AI가 붙여주도록 처리하면 사용자 부담을 덜 수 있습니다.