뉴스 크롤러 서비스 API 인터페이스 설계서

개요

뉴스 크롤러 서비스는 뉴스 데이터 수집, FTP 업로드, 파일서버 데이터베이스 저장 기능을 제공하는 REST API 서비스입니다.

기본 정보

- **서비스명**: 뉴스 크롤러 서비스 (Crawler Service)
- **포트**: 8083
- **기본 URL**: `http://localhost:8083/api`
- **인코딩**: UTF-8
- **Content-Type**: application/json

---

1. 크롤링 관리 API

1.1 크롤링 시작

엔드포인트: `POST /api/crawler/start`

설명: 배포 환경에 최적화된 뉴스 크롤링을 비동기로 시작합니다.

요청:

```
{
  // 요청 본문 없음 (빈 객체)
}
```

응답:

```
{
  "status": "success",
  "message": "배포 환경 최적화 크롤링이 시작되었습니다.",
  "timestamp": "2025-01-20T10:30:00"
}
```

에러 응답:

```
{
  "status": "error",
  "message": "크롤링 시작 실패: [에러 메시지]",
  "timestamp": "2025-01-20T10:30:00"
}
```

HTTP 상태 코드:

- `200 OK`: 크롤링 시작 성공
- `500 Internal Server Error`: 서버 오류

1.2 파일서버 데이터 DB 저장

엔드포인트: `POST /api/crawler/save-fileserver`

설명: 파일서버에 있는 뉴스 데이터를 데이터베이스에 저장합니다.

요청:

```
{
  // 요청 본문 없음 (빈 객체)
}
```

응답:

```
{
  "status": "success",
  "message": "파일서버 데이터 DB 저장이 시작되었습니다.",
  "timestamp": "2025-01-20T10:30:00"
}
```

에러 응답:

```
{
  "status": "error",
  "message": "파일서버 데이터 DB 저장 시작 실패: [에러 메시지]",
  "timestamp": "2025-01-20T10:30:00"
}
```

HTTP 상태 코드:

- `200 OK`: 저장 작업 시작 성공
- `500 Internal Server Error`: 서버 오류

1.3 크롤링 상태 확인

엔드포인트: `GET /api/crawler/status`

설명: 크롤러 서비스의 현재 상태를 확인합니다.

요청: 없음

응답:

```
{
  "status": "running",
  "message": "크롤러 서비스가 실행 중입니다.",
  "timestamp": "2025-01-20T10:30:00",
  "service": "crawler-service",
  "port": "8083",
  "deployment-optimized": true
}
```

HTTP 상태 코드:

- `200 OK`: 상태 조회 성공

1.4 크롤러 설정 조회

엔드포인트: `GET /api/crawler/config`

설명: 크롤러의 현재 설정 정보를 조회합니다.

요청: 없음

응답:

```
{
  "targetCount": 100,
  "batchSize": 10,
  "maxConcurrentRequests": 5,
  "retryAttempts": 3,
  "retryDelay": 3000,
  "requestDelay": 1000,
  "categories": [
    "POLITICS",
    "ECONOMY",
    "SOCIETY",
    "LIFE",
    "INTERNATIONAL",
    "IT_SCIENCE",
    "VEHICLE",
    "TRAVEL_FOOD",
    "ART"
  ],
  "deployment-optimized": true,
  "fileserver-based-deduplication": true,
  "scheduling-enabled": true,
  "schedule": "09:00, 19:00 (Asia/Seoul)"
}
```

HTTP 상태 코드:

- `200 OK`: 설정 조회 성공

1.5 헬스 체크

엔드포인트: `GET /api/crawler/health`

설명: 서비스의 헬스 상태를 확인합니다.

요청: 없음

응답:

```
{
  "status": "UP",
  "service": "crawler-service",
  "deployment-optimized": true,
  "timestamp": "2025-01-20T10:30:00"
}
```

HTTP 상태 코드:

- `200 OK`: 서비스 정상

---

2. FTP 업로드 API

2.1 CSV 파일 업로드 (JSON)

엔드포인트: `POST /api/ftp/upload`

설명: CSV 파일 내용을 JSON으로 전송하여 FTP 서버에 업로드합니다.

요청:

```
{
  "path": "pm/2025-01-20_pm/list",
  "filename": "politics_list_2025-01-20-15-26.csv",
  "content": "제목,내용,카테고리,날짜\n뉴스제목1,뉴스내용1,POLITICS,2025-01-20\n..."
}
```

요청 필드:

- `path` (String, 필수): FTP 상대 경로 (예: "pm/2025-01-20_pm/list")
- `filename` (String, 필수): 업로드할 파일명
- `content` (String, 필수): CSV 파일 내용

응답:

```
"업로드 성공"
```

에러 응답:

```
"업로드 실패"
```

또는

```
"업로드 오류: [에러 메시지]"
```

HTTP 상태 코드:

- `200 OK`: 업로드 성공
- `500 Internal Server Error`: 업로드 실패

2.2 파일 업로드 (MultipartFile)

엔드포인트: `POST /api/ftp/upload-file`

설명: MultipartFile을 사용하여 파일을 FTP 서버에 업로드합니다.

요청:

```
Content-Type: multipart/form-data

file: [파일 데이터]
path: "pm/2025-01-20_pm/list"
```

요청 파라미터:

- `file` (MultipartFile, 필수): 업로드할 파일
- `path` (String, 필수): FTP 상대 경로

응답:

```
"파일 업로드 성공"
```

에러 응답:

```
"파일 업로드 실패"
```

또는

```
"파일 업로드 오류: [에러 메시지]"
```

HTTP 상태 코드:

- `200 OK`: 업로드 성공
- `500 Internal Server Error`: 업로드 실패

---

3. 공통 응답 형식

성공 응답 패턴

```
{
  "status": "success",
  "message": "작업이 성공적으로 완료되었습니다.",
  "timestamp": "2025-01-20T10:30:00"
}
```

에러 응답 패턴

```
{
  "status": "error",
  "message": "에러 메시지",
  "timestamp": "2025-01-20T10:30:00"
}
```

---

4. FTP 서버 정보

연결 정보

- **서버**: dev.macacolabs.site
- **포트**: 21
- **사용자**: newsone
- **비밀번호**: newsone
- **루트 경로**: /1/

폴더 구조

```
/1/
├── am/
│   └── yyyy-MM-dd_am/
│       └── list/
```

```
└── pm/
    └── yyyy-MM-dd_pm/
        └── list/
```

---

5. 크롤링 카테고리

지원 카테고리

- `POLITICS`: 정치
- `ECONOMY`: 경제
- `SOCIETY`: 사회
- `LIFE`: 생활
- `INTERNATIONAL`: 국제
- `IT_SCIENCE`: IT/과학
- `VEHICLE`: 자동차
- `TRAVEL_FOOD`: 여행/음식
- `ART`: 예술

---

6. 배포 최적화 기능

주요 특징

- **비동기 처리**: 모든 크롤링 작업이 비동기로 실행
- **파일서버 기반 중복 제거**: 파일서버 데이터를 활용한 중복 제거
- **스케줄링 지원**: 자동 스케줄링 (09:00, 19:00)
- **배포 환경 최적화**: 클라우드 환경에 최적화된 설정
- **에러 처리**: 강화된 에러 처리 및 로깅

성능 설정

- **대상 수**: 100개 뉴스
- **배치 크기**: 10개씩 처리
- **최대 동시 요청**: 5개
- **재시도 횟수**: 3회
- **재시도 지연**: 3초
- **요청 지연**: 1초