

# FastAPI 구성 지시서

```
# FastAPI 구성 지시서 (v1.0)
# 목적: parser_service 폴더 내 회계 모듈을 FastAPI 기반으로 통합하여
#       외부 시스템·Django-HTMX 프론트엔드·다른 마이크로서비스에서 호출 가능하
#       도록 한다.
```

## 1. 프로젝트 디렉터리 구조

```
parser_service/
├── main.py          # FastAPI 인스턴스 및 라우트 집결점
├── core/            # 공통 유틸·미들웨어·예외처리
│   ├── auth.py
│   ├── errors.py
│   └── settings.py
├── models/          # Pydantic 스키마
│   └── schemas.py
├── services/        # 업무 로직 모듈 (이미 구현됨)
│   ├── je_builder.py
│   ├── gl_posting.py
│   ├── fs_aggregate.py
│   ├── validator.py
│   ├── auto_split.py
│   ├── closing_runner.py
│   └── dw_loader.py
└── __init__.py
```

## 2. **main.py** 최소 구성 예시

```
from fastapi import FastAPI, Depends, HTTPException, status
from fastapi.security import APIKeyHeader
from models.schemas import (
    InvoiceDict, JournalResult, GLPostResult,
```

```

    FSResult, ValidateResult, DWLoadResult
)
from services import (
    je_builder, gl_posting, fs_aggregate,
    validator, dw_loader
)

app = FastAPI(
    title="Accounting Parser Service",
    version="1.0.0",
    docs_url="/docs", redoc_url="/redoc"
)

# — 선택적 API Key 보호 —————
api_key_header = APIKeyHeader(name="X-API-Key", auto_error=False)

def verify_key(key: str = Depends(api_key_header)):
    if key != "YOUR_SECRET_KEY":      # ← ENV로 치환 권장
        raise HTTPException(status.HTTP_401_UNAUTHORIZED, detail="Invalid API key")

# — 엔드포인트 정의 —————
@app.post("/api/je/build", response_model=JournalResult,
          status_code=201, dependencies=[Depends(verify_key)])
def build_je_endpoint(invoice: InvoiceDict):
    return je_builder.build_journal_entries(invoice)

@app.post("/api/gl/post", response_model=GLPostResult, status_code=201)
def gl_post_endpoint(journal: JournalResult):
    return gl_posting.post_to_gl(journal)

@app.post("/api/fs/aggregate", response_model=FSResult)
async def fs_aggr_endpoint(_: str = Depends(verify_key)):
    return fs_aggregate.aggregate_statements()

@app.post("/api/validate", response_model=ValidateResult)
def validate_endpoint(data: InvoiceDict):

```

```
return validator.validate(data)
```

```
@app.post("/api/dw/load", response_model=DWLoadResult)
def dw_load_endpoint():
    return dw_loader.load_dw()
```

### 3. 엔드포인트 호출 예시

#### Python ( `requests` )

```
import requests
API = "http://localhost:8000"
headers = {"X-API-Key": "YOUR_SECRET_KEY"}

invoice = {...}          # InvoiceDict JSON
r = requests.post(f"{API}/api/je/build", json=invoice, headers=headers)
print(r.status_code, r.json())    # 201, JournalResult
```

#### `curl`

```
curl -X POST http://localhost:8000/api/je/build \
-H "Content-Type: application/json" \
-H "X-API-Key: YOUR_SECRET_KEY" \
-d @invoice_sample.json
```

### 4. 실행·테스트

```
# 로컬 개발
uvicorn parser_service.main:app --reload --port 8000

# 프로덕션 (예: Gunicorn + Uvicorn workers)
gunicorn parser_service.main:app -k uvicorn.workers.UvicornWorker -w 4
-b 0.0.0.0:8000
```

- `/docs` → Swagger UI

- `/redoc` → ReDoc UI

## 5. 예외 처리 표준화 ( `core/errors.py` 예시)

```
from fastapi import Request
from fastapi.responses import JSONResponse
from app import app

@app.exception_handler(ValueError)
async def value_error_handler(_, exc: ValueError):
    return JSONResponse(status_code=422, content={"detail": str(exc)})

@app.exception_handler(Exception)
async def generic_handler(_, exc: Exception):
    return JSONResponse(status_code=500, content={"detail": "Internal Error"})
```

## 6. 확장 고려사항

항목	간단 지침
JWT 인증	<code>fastapi.security.JWTBearer</code> 미들웨어 추가 후 <code>Depends</code> 주입
mTLS	Nginx/Envoy 앞단에서 클라이언트 인증서 검증 후 FastAPI로 프록시
Rate-Limit	<code>slowapi</code> 또는 API Gateway (Kong/Traefik) 활용
CORS	<code>from fastapi.middleware.cors import CORSMiddleware</code>
OpenTelemetry APM	<code>opentelemetry-instrumentation-fastapi</code>
Async DB 연결	<code>SQLModel</code> + <code>asyncpg</code> (연결 세션을 <code>Depends</code> 로 주입)

✅ 그대로 복사-붙여넣기 후 `YOUR_SECRET_KEY` 만 교체하면 바로 동작합니다.