Source: C:\Users\Admin\Desktop\CMC\ms1_emailIngestion\utils\config.py

```
"""
utils/config.py
Configuration th■ng nh■t cho toàn b■ microservice
"""
import os
from pathlib import Path
from dotenv import load_dotenv

# Load environment variables
load_dotenv()

# ============= Paths =============
BASE_DIR = Path(__file__).resolve().parent.parent
STORAGE_DIR = BASE_DIR / "storage"
ATTACH_DIR = STORAGE_DIR / "attachments"
LOG_DIR = STORAGE_DIR / "logs"

# T■o th■ m■c n■u ch■a có
for directory in [STORAGE_DIR, ATTACH_DIR, LOG_DIR]:
    directory.mkdir(parents=True, exist_ok=True)

# Convert to string for backward compatibility
ATTACH_DIR = str(ATTACH_DIR)
LOG_DIR = str(LOG_DIR)

# ============= Microsoft Graph API =============
CLIENT_ID = os.getenv("CLIENT_ID")
CLIENT_SECRET = os.getenv("CLIENT_SECRET")

SCOPES = [
    "Mail.ReadWrite",
    "Mail.Send",
    "User.Read"
]

if not CLIENT_ID or not CLIENT_SECRET:
    raise ValueError(
        "Missing CLIENT_ID or CLIENT_SECRET. "
        "Please set them in .env file"
    )

# ============= Downstream Services =============
MS2_CLASSIFIER_BASE_URL = os.getenv(
    "MS2_CLASSIFIER_BASE_URL",
    "http://localhost:8001/classify"
)

MS4_PERSISTENCE_BASE_URL = os.getenv(
    "MS4_PERSISTENCE_BASE_URL",
    "http://localhost:8002"
)

# ============= Service Ports =============
API_PORT = int(os.getenv("API_PORT", "8000"))
WEBHOOK_PORT = int(os.getenv("WEBHOOK_PORT", "8100"))

# ============= Polling Settings =============
DEFAULT_POLLING_INTERVAL = int(os.getenv("POLLING_INTERVAL", "300"))  # 5 phút
MAX_POLLING_ERRORS = int(os.getenv("MAX_POLLING_ERRORS", "3"))
MAX_POLL_PAGES = int(os.getenv("MAX_POLL_PAGES", "10"))

# ============= Webhook Settings =============
WEBHOOK_SUBSCRIPTION_EXPIRY_DAYS = int(os.getenv("WEBHOOK_EXPIRY_DAYS", "3"))
WEBHOOK_RENEWAL_THRESHOLD_HOURS = int(os.getenv("WEBHOOK_RENEWAL_HOURS", "1"))
MAX_WEBHOOK_ERRORS = int(os.getenv("MAX_WEBHOOK_ERRORS", "5"))


# ============= Spam Patterns =============
# Load from environment variable as a comma-separated string, then split into a list.
# This allows updating spam patterns without changing the code.
SPAM_PATTERNS_STR = os.getenv("SPAM_PATTERNS", (
    "accountprotection.microsoft.com,"
```

```python
    "security-noreply@,"
    "account-security-noreply@accountprotection.microsoft.com,"
    "noreply@email.microsoft.com"
))
SPAM_PATTERNS = [pattern.strip() for pattern in SPAM_PATTERNS_STR.split(',') if pattern.strip()]

# ============= Logging Settings =============
LOG_LEVEL = os.getenv("LOG_LEVEL", "INFO")
LOG_FORMAT = "[%(asctime)s] [%(name)s] [%(levelname)s] %(message)s"
LOG_DATE_FORMAT = "%Y-%m-%d %H:%M:%S"

# ============= Graph API Endpoints =============
GRAPH_BASE_URL = "https://graph.microsoft.com/v1.0"
GRAPH_MESSAGES_URL = f"{GRAPH_BASE_URL}/me/messages"
GRAPH_SUBSCRIPTIONS_URL = f"{GRAPH_BASE_URL}/subscriptions"

# ============= Feature Flags =============
ENABLE_ATTACHMENT_SAVE = os.getenv("ENABLE_ATTACHMENT_SAVE", "true").lower() == "true"
ENABLE_MS2_FORWARD = os.getenv("ENABLE_MS2_FORWARD", "true").lower() == "true"
ENABLE_MS4_FORWARD = os.getenv("ENABLE_MS4_FORWARD", "true").lower() == "true"
ENABLE_SPAM_FILTER = os.getenv("ENABLE_SPAM_FILTER", "true").lower() == "true"

# ============= Validation =============
def validate_config():
    """Validate configuration"""
    errors = []

    if not CLIENT_ID:
        errors.append("CLIENT_ID is required")

    if not CLIENT_SECRET:
        errors.append("CLIENT_SECRET is required")

    # Move port casting before validation
    # API_PORT = int(os.getenv("API_PORT", "8000"))
    # WEBHOOK_PORT = int(os.getenv("WEBHOOK_PORT", "8100"))
    if API_PORT == WEBHOOK_PORT:
        errors.append(f"API_PORT ({API_PORT}) and WEBHOOK_PORT ({WEBHOOK_PORT}) must be different")

    if errors:
        raise ValueError("Configuration errors:\n" + "\n".join(f"  - {e}" for e in errors))

# Validate on import
validate_config()

# ============= Print Configuration (Debug) =============
def print_config():
    """Print current configuration (for debugging)"""
    print("=" * 70)
    print("EMAIL INGESTION MICROSERVICE - CONFIGURATION")
    print("=" * 70)
    print(f"Storage Directory: {STORAGE_DIR}")
    print(f"Attachments: {ATTACH_DIR}")
    print(f"Logs: {LOG_DIR}")
    print("-" * 70)
    print(f"API Port: {API_PORT}")
    print(f"Webhook Port: {WEBHOOK_PORT}")
    print("-" * 70)
    print(f"MS2 Classifier: {MS2_CLASSIFIER_BASE_URL}")
    print(f"MS4 Persistence: {MS4_PERSISTENCE_BASE_URL}")
    print("-" * 70)
    print(f"Polling Interval: {DEFAULT_POLLING_INTERVAL}s")
    print(f"Webhook Expiry: {WEBHOOK_SUBSCRIPTION_EXPIRY_DAYS} days")
    print(f"Renewal Threshold: {WEBHOOK_RENEWAL_THRESHOLD_HOURS}h")
    print("-" * 70)
    print(f"Feature Flags:")
    print(f"  Attachment Save: {ENABLE_ATTACHMENT_SAVE}")
    print(f"  MS2 Forward: {ENABLE_MS2_FORWARD}")
    print(f"  MS4 Forward: {ENABLE_MS4_FORWARD}")
    print(f"  Spam Filter: {ENABLE_SPAM_FILTER}")
    print("=" * 70)

if __name__ == "__main__":
    print_config()
```