```
import os
import requests
from typing import List, Dict



def check_bing_api_key():
    try:
        return os.getenv("BING_API_KEY")
    except Exception as error:
        print(f"Error {error}")
        raise None



def parse_and_merge_logs(logs: List[Dict[str, str]]) -> str:
    """
    Parses logs and merges them into a single string for input to an LLM.

    Parameters:
    logs (List[Dict[str, str]]): A list of dictionaries where each dictionary represents a log entry.

    Returns:
    str: A single string containing all log entries concatenated.
    """

    merged_logs = ""
    for log in logs:
```

```
        log_entries = [
            f"{key}: {value}" for key, value in log.items()
        ]
        log_string = "\n".join(log_entries)
        merged_logs += log_string + "\n\n"

    return merged_logs.strip()



def fetch_web_articles_bing_api(
    query: str = None,
) -> List[Dict[str, str]]:
    """
    Fetches four articles from Bing Web Search API based on the given query.

    Parameters:
    query (str): The search query to retrieve articles.
    subscription_key (str): The Bing Search API subscription key.

    Returns:
    List[Dict[str, str]]: A list of dictionaries containing article details.
    """
    subscription_key = check_bing_api_key()

    url = "https://api.bing.microsoft.com/v7.0/search"
    headers = {"Ocp-Apim-Subscription-Key": subscription_key}
```

```
params = {"q": query, "count": 4, "mkt": "en-US"}

response = requests.get(url, headers=headers, params=params)
response.raise_for_status()
search_results = response.json()

articles = []
for i, result in enumerate(
    search_results.get("webPages", {}).get("value", [])
):
    article_info = {
        "query": query,
        "url": result.get("url"),
        "title": result.get("name"),
        "publishedDate": result.get("dateLastCrawled"),
        "author": (
            result.get("provider")[0]["name"]
            if result.get("provider")
            else "Unknown"
        ),
        "id": str(i + 1),  # Generating a simple unique ID
    }
    articles.append(article_info)

articles = parse_and_merge_logs(articles)
return articles
```

```
# out = fetch_web_articles_bing_api("swarms ai github")

# print(out)
```