

```
import platform
```

```
import socket
```

```
import psutil
```

```
import uuid
```

```
from typing import Dict
```

```
import requests
```

```
from swarms.utils.loguru_logger import initialize_logger
```

```
logger = initialize_logger(log_folder="capture_sys_data")
```

```
def capture_system_data() -> Dict[str, str]:
```

```
    """
```

Captures extensive system data including platform information, user ID, IP address, CPU count, memory information, and other system details.

Returns:

Dict[str, str]: A dictionary containing system data.

```
    """
```

```
    try:
```

```
        system_data = {
```

```
            "platform": platform.system(),
```

```
            "platform_version": platform.version(),
```

```
            "platform_release": platform.release(),
```

```
            "hostname": socket.gethostname(),
```

```

"ip_address": socket.gethostbyname(socket.gethostname()),
"cpu_count": psutil.cpu_count(logical=True),
"memory_total": f"{psutil.virtual_memory().total / (1024 ** 3):.2f} GB",
"memory_available": f"{psutil.virtual_memory().available / (1024 ** 3):.2f} GB",
"user_id": str(uuid.uuid4()), # Unique user identifier
"machine_type": platform.machine(),
"processor": platform.processor(),
"architecture": platform.architecture()[0],
}

```

Get external IP address

try:

```

    system_data["external_ip"] = requests.get(
        "https://api.ipify.org"
    ).text

```

except Exception:

```

    system_data["external_ip"] = "N/A"

```

return system_data

except Exception as e:

```

    logger.error("Failed to capture system data: {}", e)
    return {}

```

def log_agent_data(data_dict: dict) -> dict | None:

"""

Logs agent data to the Swarms database with retry logic.

Args:

`data_dict` (dict): The dictionary containing the agent data to be logged.

`retry_attempts` (int, optional): The number of retry attempts in case of failure. Defaults to 3.

Returns:

`dict` | `None`: The JSON response from the server if successful, otherwise `None`.

Raises:

`ValueError`: If `data_dict` is empty or invalid

`requests.exceptions.RequestException`: If API request fails after all retries

"""

if not `data_dict`:

`logger.error("Empty data dictionary provided")`

`raise ValueError("data_dict cannot be empty")`

`url = "https://swarms.world/api/get-agents/log-agents"`

`headers = {`

`"Content-Type": "application/json",`

`"Authorization": "Bearer`

`sk-f24a13ed139f757d99cdd9cdcae710fccead92681606a97086d9711f69d44869",`

`}`

`requests.post(url, json=data_dict, headers=headers, timeout=10)`

`# response.raise_for_status()`

return None