

Notification Script

```
import requests
import time

# Replace these values with your Jellyfin server details
JELLYFIN_SERVER_URL = "http://172.22.20.1:8096"
JELLYFIN_API_KEY = "5f1013b18bba4cada3c6741be6c49596"

# Function to get session IDs
def get_session_ids():
    headers = {"Authorization": f"MediaBrowser Token={JELLYFIN_API_KEY}"}
    params = {"activeWithinSeconds": 60} # Adjust the time as needed

    response = requests.get(f"{JELLYFIN_SERVER_URL}/Sessions", headers=headers, params=params)

    if response.status_code == 200:
        sessions = response.json()
        session_ids = [session["Id"] for session in sessions]
        return session_ids
    else:
        print(f"Failed to get session IDs. Status code: {response.status_code}")
        return []

# Function to send a message to a specific session
def send_message(session_id, header, text, timeout_ms=10000):
    headers = {"Authorization": f"MediaBrowser Token={JELLYFIN_API_KEY}"}
    url = f"{JELLYFIN_SERVER_URL}/Sessions/{session_id}/Message"

    message_data = {
        "Header": header,
        "Text": text,
        "TimeoutMs": timeout_ms
    }

    response = requests.post(url, headers=headers, json=message_data)
```

```

if response.status_code == 204:
    print(f"Message sent to session {session_id}")
else:
    print(f"Failed to send message to session {session_id}. Status code:
{response.status_code}")

# Function to send maintenance countdown messages
def send_maintenance_countdown(initial_countdown=60):
    header = "Maintenance Alert"

    # Announce initial countdown
    text = f"Server maintenance will start in {initial_countdown} minutes"
    send_message_to_all(header, text)
    time.sleep(600) # Wait for 10 minutes

    # Announce remaining countdown
    for minutes in range(initial_countdown - 10, 0, -10):
        text = f"Server maintenance will start in {minutes} minute{'s' if minutes > 1 else
''}"

        send_message_to_all(header, text)
        if minutes > 5:
            time.sleep(600) # Wait for 10 minutes
        else:
            time.sleep(60 * minutes) # Wait for remaining minutes

# Function to send a message to all online users
def send_message_to_all(header, text):
    session_ids = get_session_ids()

    if session_ids:
        for session_id in session_ids:
            send_message(session_id, header, text)

# Main script
def main():
    # Example: Send immediate messages
    header_immediate = "Immediate Alert"
    text_immediate = "This is an immediate alert!"
    send_message_to_all(header_immediate, text_immediate)

```

```
# Example: Send maintenance countdown messages with initial countdown of 60 minutes
```

```
#send_maintenance_countdown(initial_countdown=60)
```

```
if __name__ == "__main__":
```

```
    main()
```

Revision #1

Created 2025-08-08 06:48:17 CEST by pknw1

Updated 2025-08-08 06:48:40 CEST by pknw1