USER MANUAL

Credit Management Extension for OpenWebUI

Contents

1 Introduction

This manual describes how to use and manage the Credit Management Extension integrated into the OpenWebUI platform. The purpose of the application is to introduce a credit-based access system for managing and monitoring the usage of premium AI models by employees and students of the University of West Bohemia (ZČU). This document is intended for both end users and system administrators.

2 Installation guide

To use this extension, follow the steps below.

2.1 Admin Panel Setup

1. Download credit_admin.zip and copy the contents of the functions folder, which contains three functions, into OpenWebUI from this link: https://drive.google.com/drive/u/0/folders/0AD2ydOn22J9EUk9PVA

2. Extract it in the same directory where the OpenWebUI folder is located.

3. Build and run the admin panel using Docker. Open a terminal in the ***credit_admin*** directory and run:
   docker compose up --build (*This example is for Debian-based systems*.)

   Figure 1: Compilation of admin panel

Once the container is running, the Admin Panel will be available at:

http://0.0.0.0:8000

4. If there is a port conflict, please change the port in docker-compose.yml and Dockerfile, or stop the service that is blocking the port.

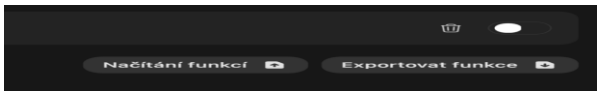
5. If the data is not being written correctly, the issue might be the name of the OpenWebUI folder, which is by default called my-openwebui. Either rename the OpenWebUI folder accordingly, or update the path in the program. If you have chosen a different location than the one mentioned above, please point the extension to the correct path of the OpenWebUI database.

2.2 User Interface Setup

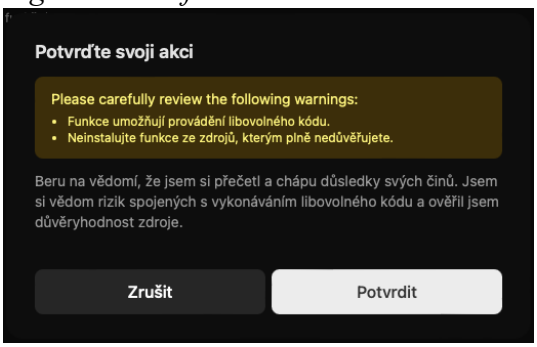
In OpenWebUI, go to the ***admin/functions*** section.

1. **Repeat the following steps for each of the three downloaded files from the functions folder** (function-credit_management_charging_credits-export-1747362819654.json, function-credit_management_enough_credits-export-1747362817459.json, and function-credit_management_models-export-1747362814961.json):

    1.1. Click the button and upload one of the files.
    Figure 2: Upload Function button

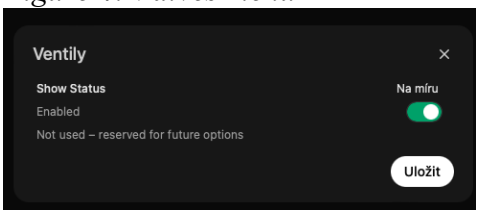


    1.2. Confirm that you trust the function.
    Figure 3: Confirm the action



    1.3. Activate the function.
    1.4. In the function settings, set it to global.
    1.5. In the Valves menu, set 'Show status' to 'Enable'.

    Figure 4: Valves menu



Once all functions are added and the Admin Panel is running, the extension is ready for use

3 Admin Panel guide

The Admin section of the Credit Management Extension allows authorized personnel to allocate credits to users, view and manage credit transactions, and monitor overall AI model usage. Access to these features is restricted to users with administrative privileges.

3.1 Allocating Credits to Users

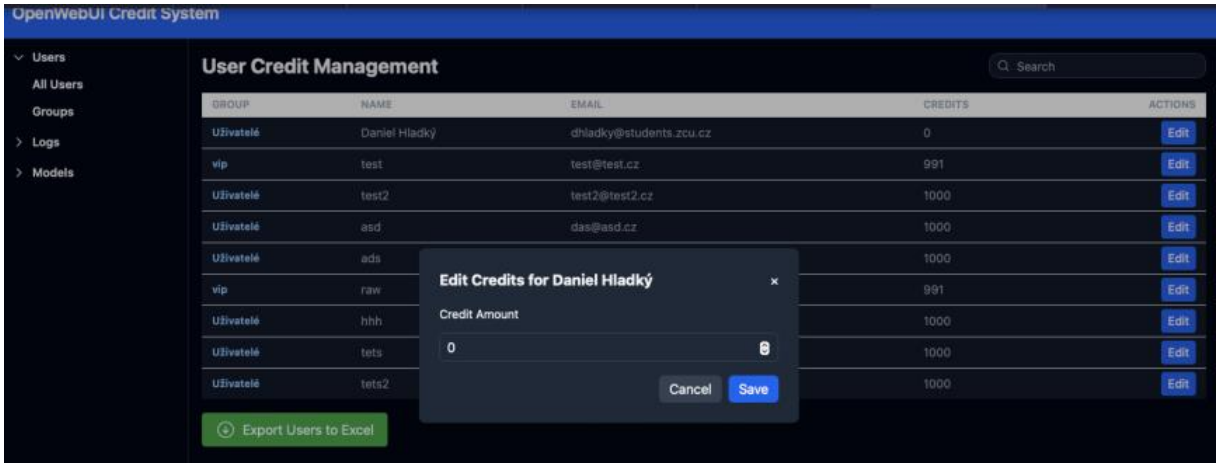
Administrators can directly view and manage credit balances for individual users. This feature is useful for manually assigning monthly credits, issuing custom bonuses, or performing test adjustments.

Steps to allocate or edit user credits:

1. In the **Credit Management** section, navigate to the **Users** page from the left-hand menu.
2. A table will list all users along with their group, email, and current **Credits**.
3. Find the user whose credit balance you want to change.
4. Click the **Edit** button in the **Actions** column.
5. In the popup window, enter the desired **Credit Amount**.
6. Click **Save** to apply the change, or **Cancel** to discard it.

⚠️ **Note:** All manual credit changes are recorded in the system logs and take effect immediately.

Figure 5: User Management Table



3.2 Allocating Credits to Groups

Administrators can configure a default credit balance for each user group. When a new user is added to a group, they automatically receive the group's default credit amount. This simplifies the credit allocation process for managing multiple users at once.

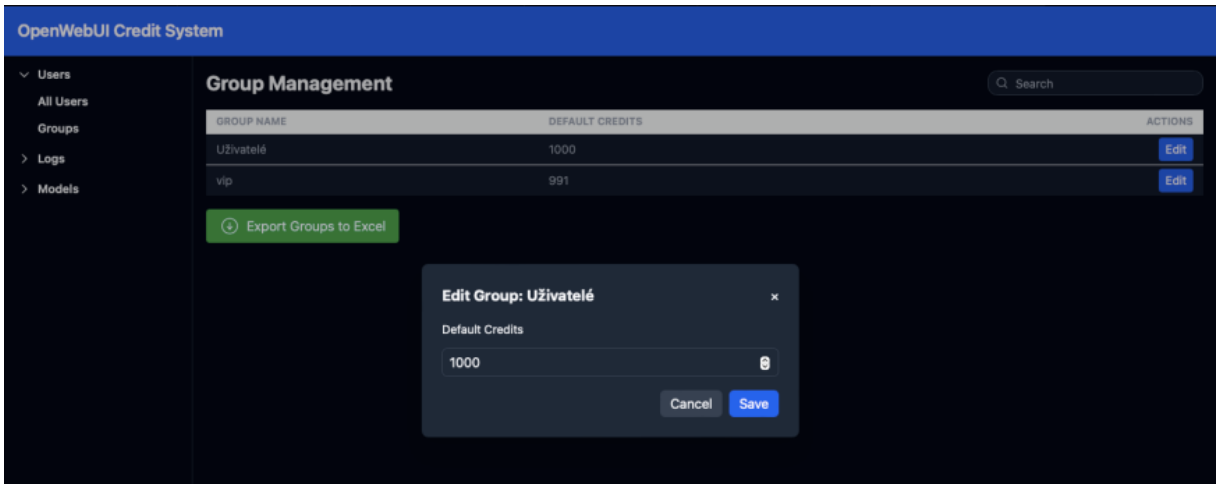
Steps to modify default group credits:

1. In the **Credit Management** section, go to the **Groups** page via the left-hand menu.

2. A table will display all existing groups and their associated **Default Credits**.
3. Identify the group you want to edit.
4. Click the **Edit** button in the **Actions** column.
5. Enter the new **Default Credits** value in the popup dialog.
6. Click **Save** to apply the change.

⚠️ **Note:** The updated default credit value only affects **new users** added to the group. Existing users will retain their current credit balance unless updated manually.

Figure 6: Group Credit Management Table



3.3 Logs

The **Logs** section in the Admin Panel provides insight into key system events and credit-related operations. It is divided into two subsections:

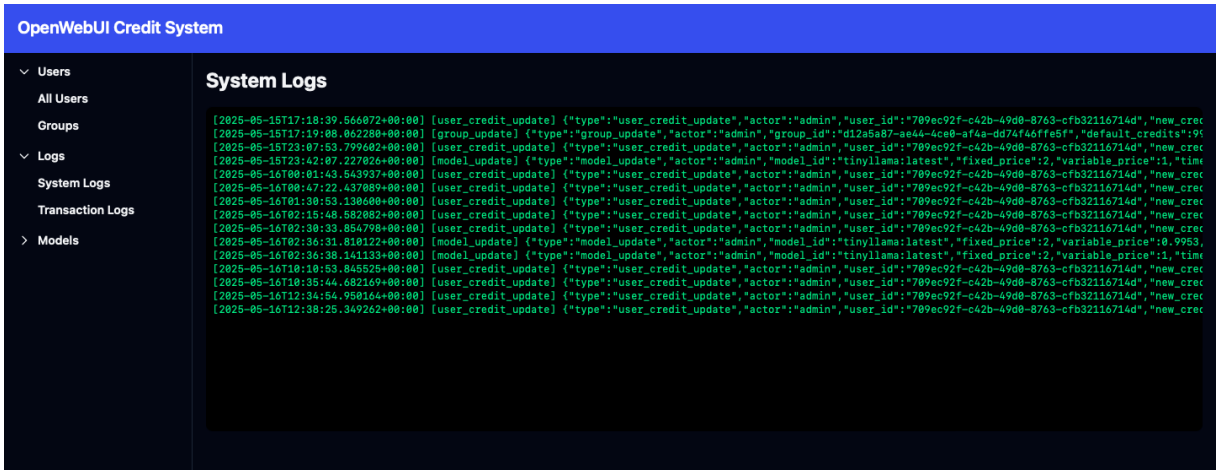
System Logs

This view records general system-level events, such as:

- Admin logins
- Configuration changes
- Role or permission modifications

System logs are useful for auditing and debugging administrative actions.

Figure 7: System Logs



Transaction Logs

This section displays all credit-related operations, including:

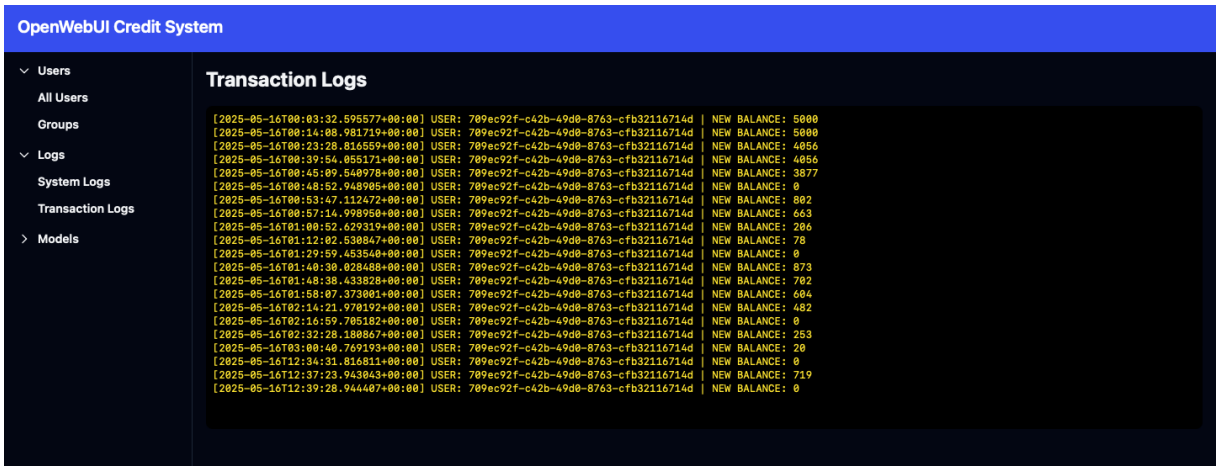
- Manual credit changes
- Credit deductions after queries
- Automatic resets or group-based allocations

Each entry typically includes a timestamp, user ID, action type, and credit change value.

🔍 Use filters or search functionality to narrow down log entries by date, user, or action.

All logs are read-only and cannot be edited, ensuring full traceability of system activity.

Figure 8: Transaction Logs



3.4 Model Pricing Management

Administrators can define and adjust the credit cost for each supported AI model. For every model, two pricing components are available:

- **Input Token Price** – the number of credits charged per input token
- **Output Token Price** – the number of credits charged per output token

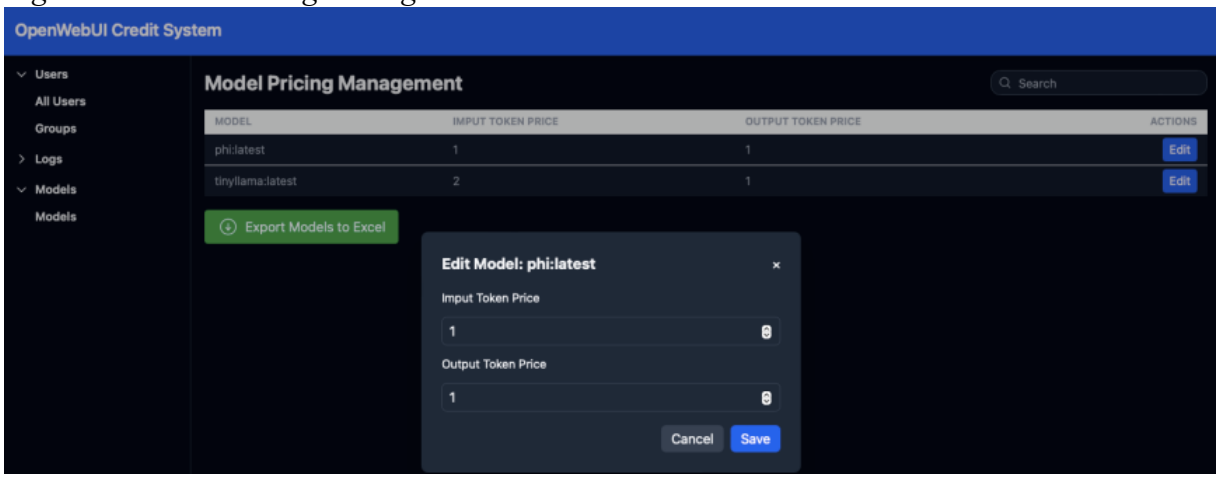
This allows fine-grained control of model usage costs and ensures fair credit deduction based on interaction volume.

Steps to manage model pricing:

1. In the **Credit Management** section, navigate to the **Models** page via the left-hand menu.
2. You'll see a list of all models currently available, with their input/output pricing.
3. Find the model you'd like to update.
4. Click the **Edit** button in the **Actions** column.
5. Adjust the **Input Token Price** and/or **Output Token Price** in the popup dialog.
6. Click **Save** to confirm the changes.

⚠️ **Note:** All pricing changes take effect immediately and will apply to the next user queries using the model.

Figure 9: Model Pricing Management Table



4 Filters – User Experience Overview

The OpenWebUI Credit System includes two essential filters that enhance the user experience by providing immediate feedback and preventing unwanted credit usage.

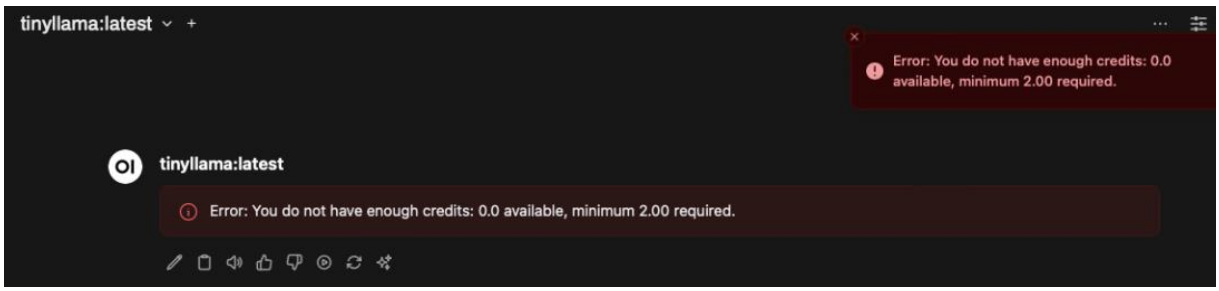
4.1 Enough Credits Filter

This filter prevents the user from sending a prompt if they do not have enough credits to cover the cost of the request.

If a user has insufficient credit, a system message appears in place of a model response, clearly explaining that the query was blocked due to lack of funds.

This feature protects users from unintended deductions and supports better credit planning.

Figure 10: Credits filter



4.2 Charging Credits Filter

This filter ensures that every time a user receives a response from the model, they are immediately informed about how many credits were used for that specific query.

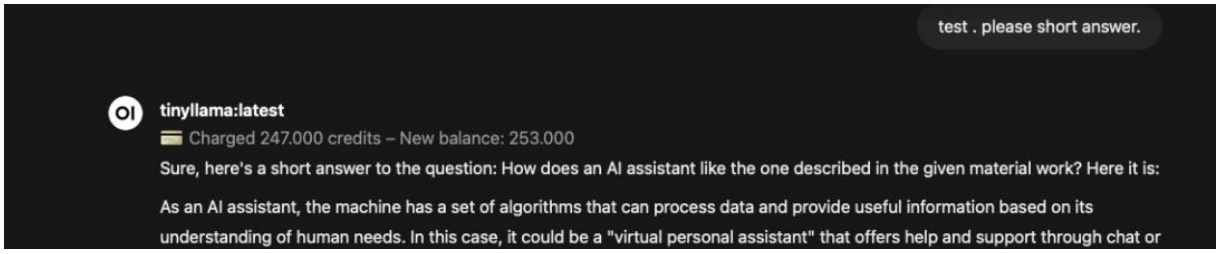
This message appears before the assistant's actual reply, ensuring that users are aware of:

- How much the query cost
- How many credits remain in their balance

This makes the system highly transparent and reinforces trust by helping users monitor their credit usage in real time.

⚠️ This message is fully automated and displayed for every prompt when the filter is enabled.

Figure 11: Query costs



4.3 Credit management Models Filter

Once a user sends a valid prompt and receives a response, the **Credit management Models** button (at the right end) allows users to manually trigger this credit information at any time. When clicked, it injects a summary of the currently active model's pricing directly into the chat window.

Figure 12: Model costs