

Hello World (D7)

Liu Xinan

January 12, 2015

Background

This drupal module is a modified version of the `helloworld_d7` module that comes with the assignment 2. I have added the mass import feature.

Helloworld

The screenshot shows the 'Mass Import User Records' tab selected in the Helloworld module interface. Below the tabs is a grey box with a right-pointing triangle and the text 'How should my data be formatted?'. Underneath is the section 'Choose your import method *' with two radio button options: 'Upload data file' and 'Enter data directly'. Below these is a 'Comment' section with a large text area and a small '//' icon in the bottom right corner. At the bottom of the form is a rounded 'Import' button.

Figure 1: Mass Import User Records

1 Basic Use Case

The `Mass Import User Records` feature is added in the form of menu tab alongside the existing `List User Records` and `Add User Record` tabs (as shown in [Figure 1](#)).

There is a fieldset which contains a guide on how the data should be formatted at the top of the form. It is collapsed by default. Click on the title to expand it.

This mass import feature supports two types of import method: `file upload` and `direct input`. Different type of line breaks: `\n`, `\r`, and `\r\n` are supported.

1.1 File Upload

Data file should be of plaintext `.txt` format and data should be structured in the stated way properly.

1. Choose the `Upload data file` radio button, a file upload form will appear.
2. Choose a file to upload.
3. (Optional) Enter *comment* for the users if necessary.
4. Click the `Import` button.

1.2 Direct Input

Data should be formatted according to the guide.

1. Choose the `Enter data directly` radio button, a text area will appear.
2. Paste or enter data into the text area.
3. (Optional) Enter *comment* for the users if necessary.
4. Click the `Import` button.

2 Miscellaneous

- I've found and fixed a small bug in the `helloworld_api`. Originally it does not sanitize the `name` and `comment` before output, which makes it vulnerable to script injections. Drupal's convention is to save everything that the user entered without escaping, but sanitize them before output. So I did not do any escaping when importing data. Instead, I added sanitize filters in the output function. This bug also exists in the `audit_log` menu of the `cvwobase_d7` module, I've fixed that also. However, since it is not part of this assignment, I'll not upload that module.
- I did not make use of the existing `add_user()` method. Instead, I implemented a `import_users()` method into the `helloworld_api`. This is because in case of error occurring during the actual writing to the database, I can rollback the whole transaction. If I choose to call `add_user()` iteratively, each `add_user()` call will start a transaction and commit it. In case of error, those successful commits won't be rolled back. And if the user sees the error and tries to upload the same file again, there will be duplicate entries in the database. Another reason is that it is not possible to nest transactions in another transaction.
- I did not perform validation on `name` so that it can support different types of characters, such as Chinese and Vietnamese.
- To easily find out the changes in code, please refer to the commit log on Github.

3 Accomplishment

Although I did not have time to add more features to it, I feel quite satisfied with this assignment. I've learnt how Drupal works and how to hook into it. I had fun with the form API.