OxOOL v4 模組編譯手冊

一、環境準備

- 參考 技術手冊_____，先行編譯 oxool-community 以產生 oxool 開發相關套件。
  - 請留意: 此步驟編譯 oxool-community 時應使用 `autogen.sh`，不需再執行 `configure`。`autogen.sh` 預設會開啟 `--enable-debug` 並載入 ModuleTesting 模組供開發測試使用。
  - 透過 `rpmbuild` 或 `debuild` 產生 `oxool` 與 `oxool-dev` 安裝套件。
- 將產生出來之 oxool-dev 套件安裝於系統內。 oxool-dev 內含編譯模組所需之標頭檔、模組範本與 `oxool-module-maker` 程式。

二、使用 oxool-module-maker 產生模組基本檔案

- 執行 oxool-module-maker 以產生新模組的基本檔案。參數說明如下:
  - `--module-name=name`：模組名稱。必須符合正規表示式: `^[\w\@#]+$`，也就是所有的大小寫英文字母、數字 0-9、底線以及 # @ 三種符號的組合。注意減號 '-' 不能使用，而底線雖然可以使用但不建議。
  - `--serviceURI=serviceURI`：此模組所使用的網址路徑。若結尾有 / 表示此模組會處理一系列指令。例如 "/oxool/user/" 表示此模組可能會處理 /oxool/user/add 、 /oxool/user/del 、 /oxool/user/update 等一系列動作。若結尾沒有 / 則為固定的網址路徑。
  - `--version=version`：指定版號。預設為 "0.0.1"。
  - `--summary=summary`：指定 Summary，用於產生出來的 rpm 檔。
  - `--description=Description`：指定 Description，用於產生出來的 deb 檔。
  - `--author=author`：模組作者。預設會去找使用者全域 git 設定中的 user.name 與 user.email 兩個欄位。
  - `--license=license`：所使用的授權條款。預設為 MPLv2.0。
  - `--adminPrivilege=true/false`：此模組的網址路徑是否需要 admin 權限。預設為 false。
  - `--adminIcon=icon`：在後台管理介面中所使用的圖示（請參考 getbootstrap.com 。預設值為 "bug-fill"。
  - `--adminItem=text`：在後台管理介面中所使用的文字。
  - `--template-path=path`：指定模組範本路徑，也就是產生新模組時要從哪裡複製檔案過來。預設會從 `/usr/share/oxool-devel/module-template`

中複製模組檔案。
- - output-path=path :
指定產生的模組要放在哪個路徑。預設為使用者的家目錄。要留意不能用 ~
符號。最好使用完整路徑。
執行範例：

oxool-module-maker --module-name="samplemod" --serviceURI="/oxool/samplemod/" --summary="A sample moudle service on OxOOL" --description="A sample module service on OxOOL" --output-path="/home/oxool/git"

完成後在指定的 output-path 中會產生一個 git repository。裡面的內容是 template-path
參數指定的路徑裡的內容。

三、編譯與測試模組
用 autogen.sh 產生 configure 檔，然後用 configure
去確認該有的標頭檔、函式庫等是否都存在，並產生 Makefile 檔。
用 make 編譯。
測試模組前，先在 oxool-community 中執行 make run 啟動 oxool 服務。
啟動後，在模組目錄中執行 ./test.sh <模組的 XML 檔>。 test.sh 會呼叫 oxool 服務中的 ModuleTesting 模組，並將 XML 路徑傳送過去。 ModuleTesting 模組則會通知 oxool 載入模組的
. so 檔（模組編譯完成後可以在 .libs/ 目錄下找到此模組的 .so 檔）。此時就可以透過
serviceURI 指定的網址路徑執行模組。例如在瀏覽器中輸入：
http://127.0.0.1:9980/oxool/samplemod/。
模組修改後需重新編譯並重新執行 test.sh 以便把新的 .so 檔載入 oxool 服務中。

Revision #1
Created 2023-08-02 06:14:05 UTC by Jeff Huang
Updated 2023-09-05 02:58:37 UTC by Jeff Huang