OxOOL ??????

本手冊將簡單介紹 OxOOL 模組結構與開發說明。

OxOOL ????????

這裡就以使用 `oxool-module-maker` 並以 `oxool-dev` 套件中預設之模組範本產生之新模組為例，重要的檔案與目錄重點說明如下:

- `ModuleConfiguration.md` ：模組配置檔說明。
- `module.xml.in` ：模組配置檔，用來在執行 `configure` 時產生模組 XML 檔。如果在 `oxool-module-maker` 所下的參數有誤或需要修改、需要調整 XML 檔中的內容或新增標籤等，則可根據 `ModuleConfiguration.md` 中的說明修改此檔並重新執行 `autogen.sh` 與 `configure` 。
- `module.spec.in` ：用來產生 rpm 軟體包的 spec 檔。
- `debian/*` ：用來產生 deb 軟體包的檔案。
- `admin/*` ：如果模組有後端管理介面，則放在此處。管理介面裡顯示的名稱定義在 `module.xml.in` 中的 `<adminItem>` 標籤，或是在執行 `oxool-module-maker` 時指定的 `--adminItem` 參數。可參考 README.md 中的說明。
- `src/*` ：模組的 C++ 源碼檔，會編譯成 .so 檔並在 OxOOL 執行時載入。
- `html/*` ：模組前端網頁介面。模組可以不必用 C++ 實作，而是直接透過網頁進行。預設的頁面為 index.html。
- `test.sh` ：模組開發期間測試時，只要執行此檔，並將模組的 XML 檔傳給 OxOOL 即可進行測試。可參考 OxOOL模組編譯手冊中的說明。

OxOOL ??? C++ ??????

`oxool-module-maker` 產生之新模組，預設在 src 下會有 `Module.cpp` 檔，裡面定義此模組的基礎類別，以模組名稱為類別名稱。裡面除了建構子與解構子之外，定義了幾個方法:

- `getVersion()` ：取得版本號。
- `initialize()` ：初始化模組。
- `handleRequest()` ：處理來自前端 Client 對模組的請求。根據模組需求去實作每個請求。
- `handleAdminRequest()` ：處理來自後端管理介面對模組的請求。根據模組需求去實作每個請求。
- `handleAdminMessage()` ：處理來自後端管理介面對模組傳送的 websocket 訊息。

Revision #1
Created 2023-09-05 02:58:47 UTC by Jeff Huang
Updated 2023-09-05 03:02:55 UTC by Jeff Huang