

# Moodle Plugin do Sphere Engine

Krzysztof Pastuszak

December 21, 2015

## Contents

<b>1</b>	<b>Wtyczki w Moodle</b>	<b>2</b>
1.1	Instalacja wtyczki . . . . .	2
1.2	Testowanie wtyczki . . . . .	2
1.3	Struktura wtyczki . . . . .	2
1.4	Warstwa prezentacji . . . . .	4
1.5	moodleform . . . . .	4
1.6	html_writer . . . . .	4
<b>2</b>	<b>Baza danych</b>	<b>5</b>
2.1	Aktualizacja bazy przy instalacji/aktualizacji wtyczki . . . . .	5
2.2	Baza danych z poziomu kodu . . . . .	5
<b>3</b>	<b>Wtyczka</b>	<b>5</b>
3.1	Koncepcja . . . . .	5
3.1.1	Dodawanie zadania . . . . .	5
3.1.2	Podgląd zadania . . . . .	5
3.1.3	Dodawanie zadania . . . . .	6
3.1.4	Wysyłanie zgłoszenia . . . . .	6
3.1.5	Dodawanie oceny . . . . .	6
3.1.6	Uwagi . . . . .	6
3.2	Implementacja . . . . .	6
<b>4</b>	<b>Zmiany w bazie danych</b>	<b>7</b>
<b>5</b>	<b>Inne sugestie</b>	<b>8</b>
5.1	Standard Learning Tools Interoperability (LTI) . . . . .	8

# 1 Wtyczki w Moodle

Wtyczka Sphere Assignment bazuje na wchodzącej w skład podstawowej wersji Moodle'a wtyczce Assignment. Obie wtyczki są typu Activity (jeden z rodzajów wtyczek Moodle'a).

## 1.1 Instalacja wtyczki

Przygotowana do instalacji wtyczka powinna zostać umieszczona w archiwum zip. Wtyczka i folder, w którym się znajduje, muszą mieć identyczne nazwy. Archiwum zip może mieć inną nazwę, jednak w trakcie instalacji trzeba będzie ją zmienić wypełniając pole "Save as...".

Proces instalacji:

Zalogowanie się jako administrator -> site administration (w domyślnej konfiguracji menu w lewej dolnej części ekranu) -> plugins -> install plugin -> install from zip -> wybór i wgranie pliku -> walidacja nazwy, wymaganej wersji moodle'a itd., głównie w oparciu o version.php -> instalacja elementów wtyczki -> aktualizacja bazy danych -> konfiguracja wtyczki.

Walidacja i instalacja odbywają się w sposób półautomatyczny, należy jedynie klikać "dalej"... Domyślne ustawienia konfiguracji są wystarczające do testów.

**UWAGA**

Aktualizacja bazy danych stanowi ostatni etap instalacji wtyczki. Jeżeli wystąpi błąd, moodle (z perspektywy administratora) zawiesza się na cyklu:

"instalacja elementów wtyczki zakończona powodzeniem. Aktualizuj bazę danych" i "Błąd aktualizacji bazy danych. Wróć". Przy każdej próbie aktualizacji bazy danych moodle generuje (na podstawie pliku "install.xml" w folderze /wtyczka/db) i uruchamia skrypty aktualizujące bazę danych. W razie wystąpienia błędów można zmienić zawartość pliku i ponowić próbę - jedyny problem stanowią tabele, które powstały przy poprzednim podejściu ("Table already exists..."). Można je oczywiście ręcznie usunąć/tymczasowo usunąć odpowiadający im fragment pliku "install.xml".

## 1.2 Testowanie wtyczki

Po zakończonej instalacji:

My courses->add course -> wpisanie nazwy itd. -> po wejściu w kurs "enroll users" - dodanie użytkownika z jakąś rolą, "finish enrolling", kliknięcie na rolę na liście użytkownika i dodanie brakujących (student, teacher, manager), przejście do kursu -> turn editing on -> "Add an activity or resource" -> wybór sphere assignment i rozpoczęcie testów...

## 1.3 Struktura wtyczki

W Moodle'u każda wtyczka znajduje się w folderze: /GłównyKatalogMoodle/mod/NazwaWtyczki

Nazwy wtyczki i folderu muszą być identyczne. Istotne pliki w głównym folderze:

- version.php  
Plik zawierający informacje o nazwie i wersji wtyczki, minimalnej wymaganej wersji Moodle i ustawieniach cron'a.

W przypadku wykrycia przez Moodle zmiany wersji wtyczki następuje przejście do ekranu aktualizacji wtyczek. Aktualizacja musi zostać zaakceptowana przez administratora. Zmiany w kodzie (bez zmiany nr wersji) zaczynają działać w momencie zapisania zaaktualizowanego pliku na dysku.

- view.php  
Podstawowy plik odpowiedzialny za wygląd głównej strony wtyczki - widzianej przez użytkownika po wyborze zadania z listy aktywności wchodzących w skład kursu.
- mod\_form.php  
Plik odpowiedzialny za stronę dodawania nowej aktywności (w tym wypadku zadania typu Sphere Assignment)
- submissionconfirmform.php  
Potwierdzenie aktualizacji/dodania rozwiązania przez studenta
- quickgradingform.php  
Dodawanie oceny rozwiązania przez nauczyciela
- submission\_form.php  
Strona, na której student może dodać/zaktualizować rozwiązanie  
Wszystkie wymienione wcześniej pliki postaci \*form.php dziedziczą po moodleform. Moodleform to nakładka na php QuickForms, która pozwala na generowanie formularzy i podstron (głównie na zasadzie addElement...)
- styles.css  
Arkusz css dla wtyczki.
- module.js  
Javascript używany do np. zwinienia/rozwinienia tabeli ze szczegółami dot. oceny zadania
- locallib.php  
Biblioteka funkcji wykorzystywanych przez wtyczkę, to tutaj należy można wstrzyknąć własny kod w procesy przetwarzania danych wzięte z wtyczki Assignment. Dołączona tutaj jest biblioteka spherelib.php
- spherelib.php  
wydzielone funkcje korzystające ze sphereApi
- upgradelib.php  
Funkcje wywoływane przy aktualizacji wersji wtyczki.
- gradingtable.php  
Oceny - interakcje z bazą danych
- feedbackplugin.php  
Dodawanie ocen
- externallib.php

Ważniejsze foldery:

- /NazwaWtyczki/db  
Pliki odpowiedzialne za aktualizację bazy danych przy instalacji/aktualizacji wtyczki. W pliku access.php wymagane uprawnienia.  
Instalacja bazy na install.xml  
Aktualizacja upgrade.php
- /NazwaWtyczki/lang  
Wersje językowe. Domyślne komunikaty itd. znajdują się w /NazwaWtyczki/lang/en/NazwaWtyczki.php
- /NazwaWtyczki/feedback  
Pliki wykorzystywane moduł Feedback (nie były zmieniane)
- /NazwaWtyczki/pix  
Wykorzystywane ikony
- /NazwaWtyczki/tests  
Testy
- /NazwaWtyczki/sphereengine  
Biblioteka od Roberta do komunikacji ze SphereAPI. Dołączona do pliku spherelib.php  
Jedyna modyfikacja - rozwiązanie problemu z html encode ( w Moodle'u przy generowaniu linków &przechodzi w &amp;#x26;
- /NazwaWtyczki/yui  
Wykorzystywane przez wtyczkę javascripty
- /NazwaWtyczki/submission  
Pliki i moduły wykorzystywane przez submission (dodawanie rozwiązań). Zawiera m.in. wbudowany edytor tekstu i dodawanie plików

## 1.4 Warstwa prezentacji

Podział elementów warstwy prezentacji między poszczególne pliki został przedstawiony przy okazji omawiania struktury wtyczki.

## 1.5 moodleform

Podstawowa forma budowania podstron wtyczki. Moodleform to nakładka na php QuickForms, która pozwala na generowanie formularzy i podstron (głównie na zasadzie addElement...).

## 1.6 html\_writer

Klasa odpowiedzialna za dodawanie elementów w postaci kodu html. Wyświetlenie html'a w inny sposób (np. 'echo('<p>')' będzie skutkowało wyświetleniem tekstu '<p>'.  
</p>

## 2 Baza danych

Wybór bazy danych i nawiązanie z nią połączenia następuje w momencie instalacji Moodle'a.

### 2.1 Aktualizacja bazy przy instalacji/aktualizacji wtyczki

Jak opisano wcześniej,

### 2.2 Baza danych z poziomu kodu

Zalecane jest korzystanie z gotowych funkcji dostarczonych przez api Moodle'a.

Data manipulation API:

[https://docs.moodle.org/dev/Data\\_manipulation\\_API](https://docs.moodle.org/dev/Data_manipulation_API)

Data definition API:

[https://docs.moodle.org/dev/Data\\_definition\\_API](https://docs.moodle.org/dev/Data_definition_API)

Funkcje z data manipulation API zapewniają funkcje zbliżone do "addslashes".

Twórcy Moodle'a nie zalecają bezpośredniego korzystania z komend SQL.

## 3 Wtyczka

### 3.1 Koncepcja

Wtyczka bazuje na wchodzącej w skład podstawowej wersji Moodle'a wtyczce Assign.

#### 3.1.1 Dodawanie zadania

W momencie tworzenia nowej instancji wyświetlana jest strona wygenerowana przez mod\_form.php. Dodałem do tej strony rozwijaną listę problemów do wyboru - wykorzystywane są funkcje ze spherelib.php (dołączony do locallib). Dane pobierane są ze SphereEngine i, jeżeli dane problemy nie były jeszcze w bazie, zapisywane do bazy danych (tabele setask\_problems (lista problemów, sphereApi->problems->all()), setask\_prob\_desc - zawiera szczegóły (zwracane przez sphereApi->get(kodProblemu))). Nie obsługowany został przypadek skasowania zadania po stronie SphereEngine.

Następnie, po zatwierdzeniu nowego zadania przez użytkownika, wywoływana jest funkcja add\_instance z locallib.php. Zapisywane są tam wszystkie szczegóły dotyczące zadania - terminy, limit podejść itp.. Zostawiłem wersję z Assign, dodałem jedynie odwołanie do spherelib (sphereAddInstance), gdzie zapisywane jest mapowanie między zadaniem i przypisanym do niego problemem (setask\_prob\_setask).

#### 3.1.2 Podgląd zadania

Strona wyświetlana jest w oparciu o view.php

### 3.1.3 Dodawanie zadania

Strona generowana jest za pomocą `submission_form.php`

Dodane zostało wczytanie i wyświetlenie listy kompilatorów.

Po próbie wysłania rozwiązania wyświetlana jest prośba o potwierdzenie - `submissionconfirmform.php`. Po wysłaniu wykonywane jest `add_attempt` z `local-lib.php` - Tutaj można dodać przechwycenie wysłanego rozwiązania i przygotować rekord do tabeli `setask_sub_data`

### 3.1.4 Wysyłanie zgłoszenia

Docelowo - skrypt `cron'a`, wysyłający zgłoszenia z kolejki i pobierający wyniki. Wersja wstępna - link do podstrony z AJAX'em wysyłającym zgłoszenia i pobierającym wyniki.

Zakładając realizację poprzednich podpunktów, wszystkie potrzebne dane są umieszczone w tabeli `setask_sub_data`. Wystarczy pobrać te dane i wysłać zgłoszenie z wykorzystaniem `spherelib`

### 3.1.5 Dodawanie oceny

We wtyczce Assign proces dodawania oceny wygląda następująco:

student wysła rozwiązanie -> wykładowca wchodzi na stronę z podglądem wysłanych rozwiązań -> ogląda i dodaje ocenę

Proponowane rozwiązanie:

Wykładowca oprócz rozwiązania widzi odpowiedź zwrotną ze `SphereEngine` - zapisaną do `setask_sub_result`

Są dwie możliwości:

- ocena jest automatycznie wstawiana, wykładowca tylko zatwierdza.  
Proponowany sposób wstawiania oceny - maksymalna liczba punktów za zadanie \* ocena ze `SphereEngine` w %
- Wykładowca ręcznie wstawia ocenę - może mieć sens w niektórych przypadkach - czasami są stosowane nietypowe modyfikatory punktacji

Proponowane rozwiązanie - dodatkowa opcja przy tworzeniu zadania, pozwalająca wybrać jeden z powyższych trybów.

### 3.1.6 Uwagi

Docelowo, zarówno problemy, jak i kompilatory, powinny być pobierane z bazy danych, z aktualizacją ze `SphereEngine` na żądanie/wg `cron'a`. Obecnie są zapisywane w bazie, ale lista wyświetlana jest na podstawie danych pobieranych przez `sphereApi`.

## 3.2 Implementacja

Zrealizowałem:

- Przygotowanie bazy danych - prawdopodobnie wersja ostateczna lub bliska ostatecznej
- Pobieranie i zapisywanie problemów

- Dostępne kompilatory
- Mapowanie problemy sphere - zadania (w aktualnej wersji mogą występować problemy, należy zweryfikować)

Należy zrealizować

- Kwestia tokenów ze SphereEngine.  
Aktualne rozwiązanie - do testów w spherelib hardkodowany jest mój token.  
Proponowane rozwiązanie:  
Klient wpisuje swój domyślny token przy konfiguracji wtyczki. Istnieje możliwość zmiany tokenu dla konkretnego zadania przy jego dodawaniu. Wszystkie rozwiązania wysyłane są z tym samym tokenem; użytkownicy są rozróżniani po stronie Moodle'a na podstawie mapowania między rozwiązaniami i ich identyfikatorami.
- Wysyłanie rozwiązań  
Zgodnie z propozycją z 3.1.4  
Próbowałem zrobić, ale nie udało mi się przygotować na czas działającej wersji.
- Dodawanie ocen  
Zgodnie z 3.1.5
- Mapowanie problemy sphere - zadania - do dopracowania

## 4 Zmiany w bazie danych

Wszystkie zmiany dotyczą nowo utworzonych tabel:

- "setask\_compilers" - dane dot. kompilatorów
- "setask\_problems" - dane o problemach (podstawowe)
- "setask\_prob\_comp" - mapowanie problemów i dostępnych dla nich kompilatorów (obecnie nie jest jeszcze wykorzystywane)
- "setask\_judges" - judge ze SphereEngine
- "setask\_testcases"
- "setask\_runtime\_info"
- "setask\_sub\_result" - wyniki ze SphereEngine
- "setask\_sub\_data" - dane do wysłania do ShereEngine
- "setask\_sub\_user" - przypisanie submission-user
- "setask\_prob\_setask" - mapowanie przypisany problem - zadanie
- "setask\_prob\_desc" - szczegółowe opisy problemów

Definicje tabel znajdują się w /setask/db/install.xml

## 5 Inne sugestie

### 5.1 Standard Learning Tools Interoperability (LTI)

Moodle umożliwia korzystanie z interfejsu LTI do obsługi zewnętrznych modułów.

LTI jest wykorzystywane obecnie przez wiele platform, kilka przykładów:

Specyfikacja standardu:

<http://www.imsglobal.org/activity/learning-tools-interoperability>

LTI z perspektywy Moodle'a:

[https://docs.moodle.org/22/en/LTI\\_Provider](https://docs.moodle.org/22/en/LTI_Provider)

[https://docs.moodle.org/22/en/External\\_tool](https://docs.moodle.org/22/en/External_tool)

Przykład użycia LTI w Moodle:

<https://www.youtube.com/watch?v=rEzKf27zgGU>