

Typy aplikacji systemowych

Typ aplikacji systemowej jest zależny od języka programowania, w którym została przygotowana funkcja. Wyróżnia się następujące typy:

- ⤴ **Aplikacja JavaScript** – wywoływany jest skrypt w języku JavaScript, wybór tego typu aplikacji systemowej wiąże się z podaniem treści kodu, w skrypcie do parametrów aplikacji można odwołać się za pomocą identyfikatorów zmiennych
- ⤴ **Aplikacja Java** – wywołuje funkcję z klasy w języku Java, wybór tego typu aplikacji systemowej wiąże się z podaniem nazwy pakietu wraz z klasą, w której zdefiniowana jest metoda **execute** implementująca działanie zadania automatycznego, parametry funkcji execute muszą być zgodne z parametrami określonymi w aplikacji, wszystkie parametry przekazane do funkcji execute są typu **AppParameter**
- ⤴ **Aplikacja Java z transakcją** – zastosowanie identyczne jak w przypadku Aplikacji Java, możliwe jest podanie jako jednego z parametrów aplikacji transakcji silnika Shark, która zostanie przekazana do metody execute
- ⤴ **Aplikacja systemowa** – umożliwia uruchamianie na serwerze dowolnej aplikacji (np. notepad, skrypt bash itp.) oraz przekazanie jej wartości zmiennych procesu, typ ten jednak nie umożliwia modyfikacji wartości zmiennych procesu, wybór tego typu aplikacji systemowej wiąże się z podaniem nazwy aplikacji wraz z pełną ścieżką, należy również wybrać jeden z trybów pracy:
 - ⤴ **Oczekiwanie na zakończenie aplikacji** - system czeka z zakończeniem zadania automatycznego do czasu zakończenia uruchomionej aplikacji
 - ⤴ **Automatyczne zakończenie aplikacji** - system automatycznie kończy zadanie zaplanowane i przechodzi do kolejnego zadania zgodnie z mapą





- ✦ **Aplikacja WebService** – umożliwia wywołanie dowolnej usługi sieciowej (WebService), aplikacja ta umożliwia zarówno odczyt jak i zapis zmiennych procesu, wybór tego typu aplikacji systemowej wiąże się z podaniem lokalizacji usługi sieciowej (ścieżki url do pliku definicji usługi sieciowej wsdl np. <http://localhost:58964/Service1.asmx?WSDL>)

