



Laboratorium 1

Jakarta Servlet

Studenci w ramach zajęć laboratoryjnych zapoznają się ze specyfikacją Servlet będącą jednym z podstawowych elementów platformy Jakarta EE (i nie tylko) służącą obsłudze żądań HTTP.

Kontrolery zapewniające interakcję z użytkownikiem powinny być zrealizowane w postaci servletów, natomiast wszelkie serwisy, repozytoria i inne zarządzane obiekty powinny być ustawiane w odpowiednich zasięgach. Realizując zadanie należy pamiętać o poprawnym rozdzieleniu interakcji z użytkownikiem od logiki biznesowej oraz dostępu do danych.

Należy zrealizować następujące zadania:

1. Należy zaimplementować mechanizm zarządzania użytkownikami. Użytkownicy są przechowywani w pamięci aplikacji. Użytkownik powinien mieć możliwość pobrania listy użytkowników oraz wybranego użytkownika w formacie JSON. Wskazanie na użytkownika odbywa się za pomocą identyfikatora użytkownika. Jako identyfikatory należy wykorzystać UUID. Podczas uruchamiania, aplikacja powinna wgrać przynajmniej 4 różnych użytkowników testowych. Poprawne wykonanie zadania wymaga m. in. właściwej dekompozycji na komponenty, poprawnego kontraktu z klientem (adresy zasobów, nagłówki, kody odpowiedzi) oraz poprawnego zarządzania stanem obiektów zarządzanych. (1 + 1 punkty)
2. Należy zaimplementować mechanizm zarządzania avatarami użytkowników. Avatary powinny być przechowywane jako pliki graficzne w ramach systemu plików w katalogu wskazanym w konfiguracji. Użytkownik powinien mieć możliwość wgrania (o ile nie istnieje), nadpisania, pobrania i usunięcia avatara. Każdy użytkownik ma maksymalnie tylko jednego avatara (nie jest przechowywana historia avatarów). Wskazanie na avatara odbywa się za pomocą identyfikatora użytkownika. Poprawne wykonanie zadania wymaga m. in. właściwej dekompozycji na komponenty, poprawnego kontraktu z użytkownikiem (adresy zasobów, nagłówki, kody odpowiedzi), poprawnego zarządzania stanem obiektów zarządzanych oraz modyfikowalnej konfiguracji bez potrzeby zmiany źródeł klas (np. za pomocą odpowiedniego deskryptora). (2 + 2 punkty)

Uwaga: Podczas prezentacji rozwiązania należy pokazać, że żądania HTTP wykonują się poprawnie. Przygotowywanie warstwy widoku nie jest tutaj konieczne. Żądania należy udokumentować w standardowym pliku `request.http` obsługiwanym przez IntelliJ IDEA HTTP Client plugin lub Visual Studio Code REST Client plugin.

Uwaga: Zgodnie z wytycznymi Java/Jakarta EE aplikacja nie powinna mieć dostępu do systemu plików a wszelkie operacje powinna realizować przez specjalny obiekt zgodny z architekturą Jakarta Connectors. Ponieważ w standardowych bibliotekach nie istnieje gotowe rozwiązanie, nie jest to konieczne w realizacji zadania.



Uwaga: Podczas tego zadania dostęp do systemu należy zrealizować bezpośrednio z wykorzystaniem pakietu `java.nio.file` (interfejs `Path` oraz klasy `Files` i `Paths`).

Uwaga: Podczas realizacji zadania pomijane są kwestie uwierzytelniania i autoryzacji. Oznacza to, że żadne z metod nie są zabezpieczone (np.: możliwość podmieniania avatara dowolnego użytkownika). Kwestie autoryzacji i uwierzytelnienia będą poruszane w ramach innego zadania.

Uwaga: Wszystkie rozwiązania należy zachować na potrzeby kolejnych zadań, chyba, że w kolejnych instrukcjach będzie napisane inaczej.