

Patronage 2018/2019

Back-end (JAVA)

Zadanie 1 - REST API – CRUD

Celem zadania jest zaprojektowanie i implementacja (bardzo uproszczonego) systemu rezerwacji sal konferencyjnych dla dowolnej liczby organizacji.

Wymagania

REST API

Komplet operacji CRUD (eng. C-reate, R-ead, U-pdate, D-elete) dla zasobów (eng. resources): organizacja, sala konferencyjna i rezerwacja sali konferencyjnej. Reprezentacja (format) danych: JSON.

Jeśli wysłane żądanie nie może zostać przetworzone, np. przy próbie dodania organizacji z istniejącą już w systemie nazwą lub niezgodną z definicją, serwer powinien zwrócić błąd HTTP (400) Bad request z opisem błędu.

Zakres informacyjny systemu:

tl;dr system ma obsługiwać wiele organizacji, dowolna z nich może mieć dowolną liczbę sal konferencyjnych z opcjonalnym wyposażeniem, każdą z sal można rezerwować (na razie bez jakiś specjalnych limitów).

Szczegółowa specyfikacja:

Atrybuty nie oznaczone jako opcjonalne są wymagane.

- organizacja:
 - o nazwa (unikalna w systemie, minimum 2 znaki, maksymalnie 20 znaków, nie może zawierać samych białych znaków)
- sala konferencyjna:
 - o nazwa (cechy takie same jak dla nazwy organizacji, np. „salka blue”)
 - o identyfikator (cechy takie same jak dla nazwy organizacji ale opcjonalny np. „1.33”)
 - o piętro (liczba 0-10)
 - o dostępna (true/false)
 - o ilość miejsc: siedzących, stojących
 - o ilość miejsc leżących (opcjonalny)
 - o ilość miejsc wiszących (opcjonalny) – p.s. tak, mamy hamaki :)
- wyposażenie sali konferencyjnej (opcjonalny):
 - o nazwa rzutnika (opcjonalny)
 - o czy jest telefon? (true/false) – jeśli jest to:
 - numer wewnętrzny (liczba naturalna, mniejsza niż 100, opcjonalny)
 - numer zewnętrzny (string, w formacie: +12 123456789, opcjonalny)
 - interfejs (USB lub bluetooth, opcjonalny)
- rezerwacja sali konferencyjnej:
 - o id rezerwującego (cechy takie same jak dla nazwy organizacji)
 - o początek i koniec rezerwacji

Rozwiązanie należy oprzeć o stabilną wersję frameworka spring-boot i dowolną wersję Java nie starszą niż 1.8. Projekt musi dać się zbudować i uruchomić przy pomocy narzędzia Gradle.

Do rozwiązania nie wymagamy użycia bazy danych (dane można przetrzymywać w pamięci systemu). Jeśli już jednak chcesz jakiejś użyć to wymagamy H2 (domyślnie dostępnej w spring-boot).

Wykonanie testów (dowolnego rodzaju) jest mile widziane.

Dostarczenie rozwiązania

- źródła aplikacji umieść w założonym przez siebie repozytorium na platformie Github (<https://github.com>)
- opis poleceń, wymaganych do uruchomienia swojej aplikacji, umieść w pliku README tak, żeby był dostępny na stronie głównej projektu na Github:
 - o jak zbudować aplikację?
 - o jak ją uruchomić?
 - o jak przy pomocy narzędzia curl dodać przykładową salę konferencyjną do systemu?
- **Link** do swojego repozytorium na github.com, zawierającego rozwiązanie, wyślij e-mailem na adres: patronage-java-leaders@intive.com i zatytułuj: „[Zadanie 1] Imię Nazwisko”

Materiały pomocnicze

- [REST API dobre praktyki](#)
- Szkielet projektu możesz utworzyć korzystając z [kreatora](#)
- [Wstęp do budowania usług REST w spring-boot](#)
- [Spring boot](#)
- [REST Validation using Bean Validation \(JSR303\)](#) – rozdział 2.
- [Przewodnik po Spring Framework](#)
- [Dokumentacja do narzędzia curl](#)

Powodzenia!