

Dzień 2

- 1) Stwórz nowy projekt w którym oprócz zależności `spring-boot-starter-web` dodasz także zależność pozwalającą na korzystanie ze Spring Data JPA oraz pamięciowej bazy H2.
- 2) Stwórz w pliku `schema.sql` strukturę tabel `Sprint` oraz `UserStory`. Przykładowa struktura danych może być następująca:
 - Sprints
 - Nazwa sprintu
 - Data rozpoczęcia
 - Data zakończenia
 - Opis celu sprintu
 - Status (Pending, In progress, Finished, Canceled)
 - Lista user stories
 - User Stories
 - Nazwa user story
 - Opis user story
 - Lista załączników (pliki binarne)
 - Liczba story points
 - Status (To do, In progress, Review, Done)
 - Lista sprintów do których należy/należało user story (może być więcej niż jeden, gdyż czasami user story nie jest ukończone w czasie pierwotnego sprintu).
- 3) Stwórz encję dla tabel `Sprints` oraz `User Stories`.
- 4) Stwórz plik `data.sql`, w którym stworzysz 1 Sprinta z 4 User Stories.
- 5) Stwórz odpowiednie interfejsy dla repozytoriów obsługujących `Sprint` oraz `User Story`.
- 6) Stwórz metodę serwisową zapisującą nowy sprint oraz drugą metodę zapisującą nowe user story. Weź przy tym pod uwagę następujące kryteria walidacji danych:
 - Sprint
 - Wszystkie pola muszą być wypełnione oprócz celu sprintu oraz listy user stories.
 - Data rozpoczęcia \leq Data zakończenia
 - Wartość statusu musi być jedną z 4 wymienionych wartości
 - User Story
 - Pola lista załączników, story points oraz lista sprintów nie są obowiązkowe.
 - Jeżeli nie przekazano statusu to domyślnie ustawiany jest status To do.
- 7) Jeżeli któryś z warunków określonych w punkcie 6 nie jest spełniony to należy wyrzucić odpowiedni wyjątek, który spowoduje rollback transakcji.
- 8) Napisz odpowiednie testy integracyjne, które zweryfikują poprawność implementacji punktów nr 6 i 7.
- 9) Stwórz metodę serwisową, która odczyta wszystkie user stories ze sprintu o podanym id. Napisz odpowiedni test.
- 10) Stwórz metodę serwisową, która zwróci listę wszystkich sprintów realizowanych w zadanym czasie (od-do). Napisz odpowiedni test.
- 11) Stwórz metodę serwisową, która zliczy liczbę story pointów zrealizowanych w danym sprincie. Trzeba w tym przypadku wziąć pod uwagę tylko user stories, które zostały w pełni zrealizowane czyli mają status Done. Zapytanie dla tej metody napisz w języku JPQL. Napisz odpowiedni test.
- 12) Stwórz klasę, która utworzy 100 user stories z randomowymi danymi.
- 13) Zmień interfejs dla User Story tak, aby obsługiwał stronicowanie.

- 14) Napisz test, który sprawdzi poprawność pobierania stron i ich zawartość.
- 15) Napisz test, który oprócz stronicowania obsługuje sortowanie dla metody pobierającej dane o sprintach, i sortuje je wg. daty rozpoczęcia sprintu. Na początku powinien znaleźć się najnowszy sprint.
- 16) Stwórz metodę, która jest transakcyjna i zapisuje w jednym kroku nowy sprint oraz listę user stories.
- 17) Włącz debugowanie transakcji i przejrzyj logi aplikacji.

Jeśli udało Ci się przejść przez wszystkie powyższe 17 punktów i wciąż masz czas 😊 to proponuję, aby przyjrzeć się poniższemu problemowi (to ten sam projekt, który był opisywany w dniu nr 1) i spróbować stworzyć podstawowe metody dla repozytoriów i serwisów, które będą wg. Ciebie potrzebne do zrealizowania opisanych wymagań.

Firma SH jest producentem oprogramowania. Realizuje swoje usługi w formie projektów wykonywanych na rzecz klientów zewnętrznych z wielu różnych krajów. Naszym celem będzie zaprojektowanie aplikacji, która będzie wspomagać zarządzanie takim portfelem projektów.

Nasi klienci mogą mieć swoje oddziały w różnych krajach i miastach (musimy umieć je rozróżniać). Każdy z nich może zlecić nam dowolną liczbę projektów. Projekty realizowane są w różnych technologiach przez różne zespoły. Zespoły należą do jednego z 3 departamentów nazwanych roboczo Java, .NET i SAP. Interesuję nas jednak nie tylko język programowania czy główna technologia ale również bardziej szczegółowy opis stosu technologicznego każdego z projektów. Po każdym miesiącu wystawiana jest klientowi faktura. To co nas interesuje przede wszystkim to pozycje faktury i termin płatności oraz jej aktualny status (przygotowana, wysłana, zapłacona).

Zespoły to pracownicy firmy SH, którzy mogą być zatrudnieni na jeden z 3 rodzajów umów: etat, B2B oraz zlecenie. Pracownicy pełnią w zespołach różne role np. programista, analityk, architekt, tester, manager, itd. Pracownik w różnych projektach może pełnić różne role.

Projekty realizowane są w sprintach, które mogą trwać od 1 do 4 tygodni. Każdy sprint składa się z zestawu user stories, które zostały ukończone w ramach danego sprinta.

Dodatkowo trzeba wziąć pod uwagę, że firma SH ma 2 oddziały. Pierwszy w Gliwicach, drugi w Opolu. Pracownicy powinni być przypisani do jednego z tych oddziałów.

Zapoznaj się z poniższymi materiałami:

- <https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa>
- <https://www.baeldung.com/spring-data-derived-queries>
- <https://www.baeldung.com/spring-data-jpa-query>
- <https://www.baeldung.com/spring-data-jpa-pagination-sorting>
- <https://www.baeldung.com/transaction-configuration-with-jpa-and-spring>