

Dzień 3

- 1) Uwaga wstępna: kontroler nie powinien zwracać encji, a obiekt DTO. W przypadku wystąpienia jakiegokolwiek błędu powinniśmy otrzymać odpowiedź z opisem błędu w postaci obiektu JSON.
- 2) Stwórz metodę REST API, która zwraca listę sprintów. Dodatkowo niech metoda ta przyjmuje parametr żądania tasks typu boolean. W przypadku, gdy parametr tasks ma wartość true dodatkowo dołącz listę user stories (tylko pola nazwa i liczba story points).
- 3) Stwórz metodę REST API, która dodaje nowe user story do sprintu o podanym id.
- 4) Stwórz metodę REST API, która zwraca liczbę story points zrealizowaną w sprincie o podanym id.
- 5) Stwórz metodę REST API, która zwróci listę user stories zrealizowanych w danym sprincie o podanym id (odpowiedź powinna zawierać wszystkie dane user story oprócz opisu i załączników).
- 6) Stwórz metodę REST API, która zwróci opis danego user story o podanym id.
- 7) Stwórz metodę REST API, która dołączy nowy załącznik do user story o podanym id.
- 8) Stwórz metodę REST API, która pobierze załącznik do user story.
- 9) Stwórz metodę REST API, która dokona aktualizacji statusu danego Sprintu.
- 10) Stwórz metodę REST API, która usunie user story o podanym id.
- 11) Stwórz metodę REST API, która zwróci informacje o sprintach (nazwa, data od, data do, status) z podanego zakresu dat.
- 12) Stwórz metodę REST API, która zwraca listę user story posortowaną wg. nazwy user story. Dodatkowo metoda powinna obsługiwać stronicowanie tzn. należy do niej przekazać numer wielkość strony.
- 13) Napisz testy integracyjne do 3 wybranych powyższych metod.
- 14) Przejrzyj dokumentację API kursów walut, które są udostępniane przez Narodowy Bank Polski.
<https://api.nbp.pl/#kursyParams>
- 15) Stwórz kod, który pobierze kursy wszystkich walut z poprzedniego dnia.
- 16) Stwórz kod, który pobierze kurs USD z ostatnich 10 dni.
- 17) Napisz aspekt z advice typu Around, który wykona się dla wszystkich metod serwisowych. Przed rozpoczęciem wykonania danej metody należy zalogować jej parametry wejściowe, a po wykonaniu metody zwracany przez nią wynik.
- 18) Stwórz zdarzenie UserStoryCreatedEvent, które zostanie utworzone po dodaniu nowego user story. Odbiorcą tego zdarzenia niech będzie klasa SprintService, która doda user story do najnowszego sprintu, który ma status Pending.
- 19) Dodaj Spring Security do projektu.
- 20) Skonfiguruj Spring Security zgodnie z przykładem podanym na slajdach.
- 21) Dodaj dwóch użytkowników – 1 z rolą USER, 2 z rolą ADMIN.
- 22) Stwórz metodę REST API, która wyświetli aktualnie zalogowanego użytkownika oraz jego role.

Zapoznaj się z poniższymi materiałami:

- <https://www.baeldung.com/spring-mvc-annotations>
- <https://www.baeldung.com/exception-handling-for-rest-with-spring>
- <https://www.baeldung.com/rest-template>
- <https://www.baeldung.com/spring-aop>
- <https://www.baeldung.com/spring-events>