import io.cucumber.java.en.\*;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.test.autoconfigure.web.servlet.AutoConfigureMockMvc;

import org.springframework.http.MediaType;

import org.springframework.test.web.servlet.MockMvc;

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.\*;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.\*;

@AutoConfigureMockMvc

public class ProductSteps {

@Autowired

private MockMvc mockMvc;

@Given("the following products exist:")

public void theFollowingProductsExist(io.cucumber.datatable.DataTable dataTable) throws Exception {

// Logic to insert products into the database using the provided data table

}

@When("I request to read the product with ID {int}")

public void iRequestToReadTheProductWithID(int id) throws Exception {

mockMvc.perform(get("/products/{id}", id))

.andExpect(status().isOk());

}

@Then("the response status should be {int}")

public void theResponseStatusShouldBe(int status) throws Exception {

// Logic to assert the response status

}

@Then("the product details should be:")

public void theProductDetailsShouldBe(io.cucumber.datatable.DataTable expectedData) throws Exception {

// Logic to verify that the returned product details match the expected data

}

@When("I request to read the product with ID {int}")

public void iRequestToReadTheNonExistingProductWithID(int id) throws Exception {

mockMvc.perform(get("/products/{id}", id))

.andExpect(status().isNotFound());

}

@Then("the response body should contain the message {string}")

public void theResponseBodyShouldContainTheMessage(String message) throws Exception {

// Logic to verify the response body contains the expected message

}

}