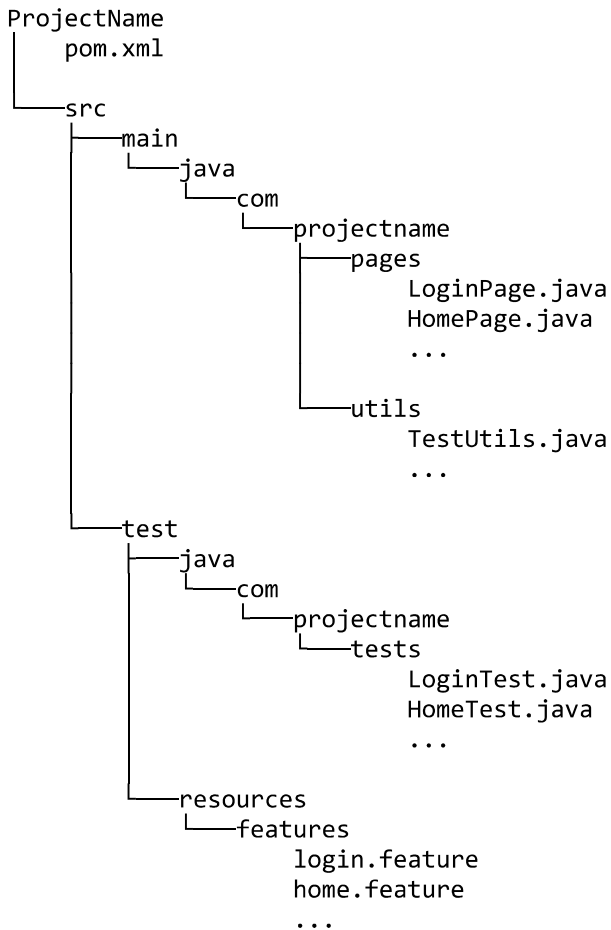


Copilot

Sure, I'd be happy to explain the folder structure for a Java Selenium project using Behavior Driven Development (BDD) and Page Object Model (POM). Here's a typical structure:



Here's what each part represents:

- **pom.xml:** This is your Project Object Model file for Maven. It includes all the dependencies needed for your project.
- **src/main/java:** This is where your main project code goes.
- **src/main/java/com/projectname/pages:** This is where you'll put your Page Object Model classes. Each class represents a page in your web application, and each method in the class represents an action you can take on that page.
- **src/main/java/com/projectname/utils:** This is where you can put utility classes, such as classes for reading from config files, common test functions, etc.
- **src/test/java:** This is where your test code goes.
- **src/test/java/com/projectname/tests:** This is where you'll put your test classes. Each test class will use methods from the Page classes to perform actions on the web application and assert that the application behaves as expected.
- **src/test/resources/features:** This is where you'll put your .feature files for Cucumber, which is commonly used in BDD. Each feature file represents a single feature of your application, and contains several scenarios that need to be tested for that feature.

This structure helps in maintaining a clean project setup and promotes code reusability and readability. It separates tests from page objects and features, making it easier to update tests as the application changes. It also makes it easier for new team members to understand the project and start contributing to it. Hope this helps! Let me know if you have any more questions. 😊