**テストコード解説**

UserController

このコードは、Spring Frameworkを使用したWebアプリケーションのコントローラークラスであるUserControllerのテストクラスです。テストの目的は、各種メソッドが正しく動作するかを検証することです。

テストクラスのアノテーション@WebMvcTest(UserController.class)は、UserControllerクラスのみを対象としてWebアプリケーションのテストを実行することを示しています。

各テストメソッドの概要は以下の通りです：

testDisplayList()：ユーザー一覧画面が正しく表示されることを検証します。UserServiceのsearchAll()メソッドがモックされ、テストデータを返すように設定されます。

testDisplayDetail()：ユーザー詳細画面が正しく表示されることを検証します。UserServiceのsearch()メソッドがモックされ、指定されたユーザーIDに対応するユーザー情報を返すように設定されます。

testDisplayAdd()：ユーザー登録画面が正しく表示されることを検証します。UserServiceとのやり取りは行われません。

testCreateUser\_Success()：ユーザーの正常な登録ができることを検証します。UserServiceのcreateUser()メソッドがモックされ、指定されたユーザーオブジェクトを返すように設定されます。

testCreateUser\_ValidationError()：ユーザー登録時にバリデーションエラーが発生する場合を検証します。UserServiceとのやり取りは行われません。

testDeleteUser()：指定したユーザーの削除ができることを検証します。UserServiceのdeleteUser()メソッドがモックされ、指定されたユーザーIDに対応するユーザーが削除されるように設定されます。

testDisplayEdit()：ユーザー編集画面が正しく表示されることを検証します。UserServiceのsearch()メソッドがモックされ、指定されたユーザーIDに対応するユーザー情報を返すように設定されます。

testUpdateUser\_Success()：ユーザーの正常な更新ができることを検証します。UserServiceのupdateUser()メソッドがモックされ、指定されたユーザーオブジェクトを返すように設定されます。

testUpdateUser\_ValidationError()：ユーザー更新時にバリデーションエラーが発生する場合を検証します。UserServiceとのやり取りは行われません。

testUpdateUser\_OptimisticLockException()：ユーザー更新時に楽観的排他制御エラー（OptimisticLockException）が発生する場合を検証します。UserServiceのupdateUser()メソッドがモックされ、OptimisticLockExceptionをスローするように設定されます。

テストでは、各メソッドが期待される結果と一致するかどうかをMockMvcを使用して検証しています。例えば、status().isOk()はHTTPステータスコードが200であることを、view().name("user/list")はビューの名前が"user/list"であることを検証します。

verify()メソッドは、UserServiceのメソッドが指定の回数呼び出されたかどうかを検証するために使用されます。

これらのテストは、UserControllerの各メソッドが正しく動作し、意図した結果を返すことを確認するために実行されます。テストの結果が期待通りであれば、アプリケーションの信頼性を高めることができます。