Story Tests

MidPoint contains a set of "story tests". Such tests implement almost complete midPoint deployment scenarios with all the configuration, resources and policies. These tests have a configuration that is almost real-world but the individual resources and data are simplified so we are able to automatically execute them as integration tests. This helps to make sure that midPoint works in situations that are very close to reality.

Story Tests as Configuration Samples

As the story tests contain almost complete sophisticated scenarios they are not only interesting to developers but also to system engineers. The configuration used in story tests can be used as an inspiration for midPoint configuration in similar real-world scenarios. Therefore pages like this one document the story tests. Each page describes the basic idea of the test and the most important part of the configuration that is used in the test.

Documented story tests:

- LDAP Hierarchy Story Tests
- OrgSync Story Test
- Unix Story Test

Story Test Development

Story tests are integration tests. Therefore please see the Integration Tests page for a general introduction to midPoint integration tests. This is important information to understand the development of story tests.

All the story tests have some very basic common configuration (in src/test/resources/common). This common configuration mostly consists only from the objects that are essential for every midPoint instance such as system configuration, administrator user and superuser role, scanner tasks and so on. There is also couple of testing users that can be reused in the tests. But that's all. All other configuration much be loaded by each story test.

There is an AbstractStoryTest superclass that the story tests extend. This is just a convenience class that loads the common configuration. Otherwise it is almost empty.

The basic steps to develop a new story tests are:

- Create a test class, e.g. `TestSpaceStation`. This class should extend the `AbstractStoryTest` superclass. Make sure you have proper annotations on the class to initialize and clean up spring context (copy&paste from existing test classes).
- Create configuration subdirectory, e.g. `src/test/resources/space-station`. The `TEST_DIR` constant should point to that.
- Put all your configuration objects into that directory.
- Implement initSystem() method to load these objects. See existing test classes for examples how to do that. Important: do not forget to invoke the superclass method: `super.initSystem(...)`
- Initialize your resources in the initSystem() method. E.g. see Dummy Resource page. Use OpenDJController to start embedded LDAP server instance if needed. Use other existing test classes as examples.
- Write the test methods.
- Add your tests to testng-integration.xml

Test Execution

Story tests are **not** executed during normal build of the entire midPoint system (mvn clean install). Therefore you can create pretty intensive configuration without impacting midPoint build time. Individual tests are easy to execute from command-line even without using the special profile:

```
cd testing/story
mvn clean test -Dtest=TestSpaceStation
```

The story tests are executed in bamboo `longtest` plan which is executed usually once per day. Therefore if you modify a story test then keep an eye on that plan. You will be expected to fix the test if it fails.

See Also

- Integration Tests