Concept - Multi-item Actions

Goal

The goal is to allow actions to act over more than one item, e.g. deleting multiple items at once etc.

The restriction is to make it possible with minimal intervention to the current API.

Jira issue: ☑ ~~MGNLUI-1515~~ - Allow actions to be executed on more items CLOSED

Considerations

- most of the actions will remain single-item restricted;
- multi-item actions are to be used (mostly) in the workbench;
- neither `Action` nor `ActionDefinition` interfaces specify anything about the items passed to the action - it's all done in the action's constructor;
- the `ActionExecutor` interface already expects multiple items in the `isAvailable(String actionName, Item... items)` method, and virtually also in the `execute(String actionName, Object... args)` method;

Proposal

Next: 5.1

The action availability described in the Concept - Action availability and access control will be extended with an **`availability.multiple`** property, set by default to **`false`**.

The `AbstractActionExecutor.isAvailable()` method will be changed to take the `availability.multiple` property into account in the availability evaluation (ATM it just returns false if the `items.length` is bigger than 1). Note (for the documentation): if more advanced availability evaluation is required (e.g. action should be available only if all the items are on the same level, or have the same parent), the Availability Rules should be used.

The `BrowserPresenter.executeAction()` method will be changed to pass the whole item list to the `actionExecutor` in such case, instead of just the first selected item. (There is already a TODO note with the above issue ID.)

The action is responsible to handle the items in a correct order, or solve the obvious dependencies (e.g. when deleting multiple nodes, where one node is a sub-node of the other one, delete them in a correct order to avoid exception, or handle such exception).

The action is responsible for informing the user about the result, i.e. whether it has successfully processed all the items, or failed on some of them. It is also responsible for the potential "transactional" processing, i.e. reverting all the items to the initial state, if the processing of one or more items fails. The recommended behaviour for most actions is "non-transactional" plus notifying the user about how many and which items couldn't be processed. The base support for this recommended behaviour will be implemented in new `AbstractMultiItemAction` class.

If the action supports multiple items, it must provide a constructor with a `List<Item>` parameter, but still MUST provide also the constructor with the `Item` parameter. The `ActionExecutor` will use the proper constructor (via the `ComponentProvider)` depending on the number of items selected.

As an example (and a best-practices model), the `DeleteItemAction` will be made multi-item supporting.

Future

- ☑ MGNLUI-1942 - Provide a general way/recommendation how to handle multiple items of different types OPEN

TODO - changes and enhancements after the 5.1 release