

## Test 1:

```
it('Should create an issue and validate it successfully', () => {
  // System finds modal for creating issue and does next steps inside of it
  cy.get('[data-testid="modal:issue-create"]').within(() => {
    // Type value to description input field
    cy.get('.ql-editor').type('TEST_DESCRIPTION');
    cy.get('.ql-editor').should('have.text', 'TEST_DESCRIPTION');

    // Type value to title input field
    // Order of filling in the fields is first description, then title on purpose
    // Otherwise filling title first sometimes doesn't work due to web page implementation
    cy.get('input[name="title"]').type('TEST_TITLE');
    cy.get('input[name="title"]').should('have.value', 'TEST_TITLE');

    // Open issue type dropdown and choose Story
    cy.get('[data-testid="select:type"]').click();
    cy.get('[data-testid="select-option:Story"]').wait(1000).trigger('mouseover').trigger('click');
    cy.get('[data-testid="icon:story"]').should('be.visible');

    // Select Baby Yoda from reporter dropdown
    cy.get('[data-testid="select:reporterId"]').click();
    cy.get('[data-testid="select-option:Pickle Rick"]').click();

    // Select Baby Yoda from assignee dropdown
    cy.get('[data-testid="form-field:userIds"]').click();
    cy.get('[data-testid="select-option:Lord Gaben"]').click();

    // Click on button "Create issue"
    cy.get('button[type="submit"]').click();
  });

  // Assert that modal window is closed and successful message is visible
  cy.get('[data-testid="modal:issue-create"]').should('not.exist');
  cy.contains('Issue has been successfully created.').should('be.visible');

  // Reload the page to be able to see recently created issue
  // Assert that successful message has dissappeared after the reload
  cy.reload();
  cy.contains('Issue has been successfully created.').should('not.exist');
```

```

// Assert than only one list with name Backlog is visible and do steps inside of it
cy.get('[data-testid="board-list:backlog"]')
  .should('be.visible')
  .and('have.length', '1')
  .within(() => {
    // Assert that this list contains 5 issues and first element with tag p has specified text
    cy.get('[data-testid="list-issue"]')
      .should('have.length', '5')
      .first()
      .find('p')
      .contains('TEST_TITLE')
      .siblings()
      .within(() => {
        //Assert that correct avatar and type icon are visible
        cy.get('[data-testid="avatar:Lord Gaben"]').should('be.visible');
        cy.get('[data-testid="icon:story"]').should('be.visible');
      });
  });

cy.get('[data-testid="board-list:backlog"]')
  .contains('TEST_TITLE')
  .within(() => {
    // Assert that correct avatar and type icon are visible
    cy.get('[data-testid="avatar:Lord Gaben"]').should('be.visible');
    cy.get('[data-testid="icon:story"]').should('be.visible');
  });
});

```

Test 2:

First of all install the faker dependency to your project, using following command in the terminal:

```
npm install @faker-js/faker --save-dev
```

Then add to the beginning of the file the following line:

```
import { faker } from '@faker-js/faker'
```

### And the test:

```
it('Should create another task and validate it successfully', () => {
  let title = faker.lorem.word();
  let description = faker.lorem.sentence();

  cy.get('[data-testid="modal:issue-create"]').within(() => {
    cy.get('.ql-editor').type(description);
    cy.get('input[name="title"]').type(title);
    cy.get('[data-testid="select:reporterId"]').click();
    cy.get('[data-testid="select-option:Baby Yoda"]').click();
    cy.get('[data-testid="select:priority"]').click();
    cy.get('[data-testid="select-option:Low"]').click();

    //assert all values are correctly visible
    cy.get('input[name="title"]').should('have.value', title);
    cy.get('.ql-editor').should('have.text', description);
    cy.get('[data-testid="select:type"]').should('have.text', 'Task');
    cy.get('[data-testid="select:reporterId"]').should('have.text', 'Baby Yoda');
    cy.get('[data-testid="select:priority"]').should('have.text', 'Low');
    cy.get('button[type="submit"]').click();
  });

  //Assert that modal window is closed and successful message is visible
  cy.get('[data-testid="modal:issue-create"]').should('not.exist');
  cy.contains('Issue has been successfully created.').should('be.visible');
  cy.reload();
  cy.contains('Issue has been successfully created.').should('not.exist');

  //Assert than only one list with name Backlog is visible and do steps inside of it
  cy.get('[data-testid="board-list:backlog"]').should('be.visible').and('have.length', '1').within(() => {
    //Assert that this list contains 5 issues and first element with tag p has specified text
    cy.get('[data-testid="list-issue"]')
      .should('have.length', '5').first()
      .find('p').contains(title);
    //Assert that correct avatar and type icon are visible
    cy.get('[data-testid="icon:task"]').should('be.visible');
```

```
});  
});
```