

Approval Tests in C++

NSWI126 Presentation



Agenda

- Introduction into the topic
- What is an Approval Test
- How does it differ from the other unit tests
- Example of usage

Introduction into the topic (by example)

- ❑ Clone the repository: <https://github.com/MichalKyjovsky/NSWI126>
- ❑ Open the cloned repository in your favourite IDE and navigate to the approval-test directory.
- ❑ Build the project (CMake) and run the application
- ❑ Execute the test cases in the ***test/integration_tests-catch2.cpp*** and observe the results

What is wrong with the previous approach

- Time spent on the expected result formatting
- Not automation-friendly
- Usually we do not test only string data
- Usage of raw assertions
- Hard to read test cases

```
// Arrange, Act
Sandwich s = createSandwichForTest();
// Assert
REQUIRE("Sourdough" == s.getBread());
REQUIRE(s.getCondiments().contains("Mayo"));
REQUIRE(s.getCondiments().contains("Pepper"));
REQUIRE(s.getCondiments().contains("Olive Oil"));
REQUIRE(s.getFillings().contains("Tomato"));
REQUIRE(s.getFillings().contains("Lettuce"));
REQUIRE(s.getFillings().contains("Cheddar"));
```

Solution: Approval Tests (by example)

- Navigate to the **test** directory within the cloned project
- Run the Test Case in the **integration_tests-approvals.cpp**
- Observe the difference

```
// Arrange, Act
Sandwich s =
createSandwichForTest();
// Assert
Approvals::verify(s);
```

```
sandwich {
  bread:
    "Sourdough", condiments : [ "Mayo", "Pepper", "Olive Oil"
    ],
                                fillings : [ "Tomato", "Lettuce",
    "Cheddar" ]
}
```

Approval Tests

- Framework that enables you to verify a chunk of output (string, image, object or file) in one operation
- Replaces usage of vast amount of assertions dedicated for each element
- Open Source
- Project Repository: [link](#)
- Official documentation: [link](#)

Approval Tests - Pros

- + Convention over Configuration approach
- + Simple to use
- + Scrubber
- + Convenient updates of expected results (approved)
- + Cooperation with leading testing frameworks (doctest, googletest, catch2, etc.)
- + Well documented tool
- + CI supported

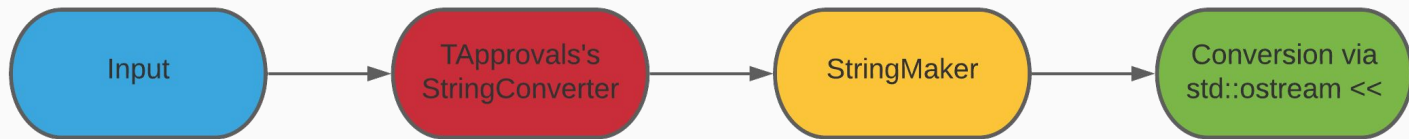
Approval Tests - Cons

- Limited amount of supported Reporters
- Platform issues
- Adjustments to the configuration may be tricky

Approval Tests - String conversions

- Approval Tests converts all tested objects into a String
- Ability to handle vast amount of objects (files, images, sounds, objects)

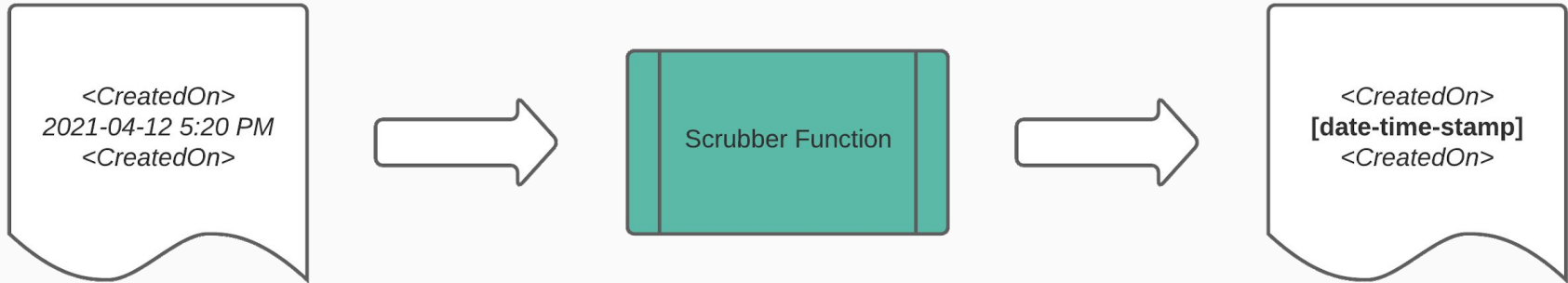
Approval Test - Text Processing



Approval Tests - Scrubber

- How to solve variable inputs (dates, numeric values,...)

Approval Tests - Scrubber



Q/A - If any

Thank you for
your attention