2021-05-26

**Meteo.lt API**

**TEST PLAN**

Turinys

[Introduction 3](#_Toc72966472)

[Test Strategy 3](#_Toc72966473)

[Preconditions 3](#_Toc72966474)

[Suspension and Resumption criteria 3](#_Toc72966475)

[Item to be Tested 4](#_Toc72966476)

[Item not to be Tested 4](#_Toc72966477)

[Testing Approach 4](#_Toc72966478)

[Acceptance testing 4](#_Toc72966479)

[Test Tools 5](#_Toc72966480)

[Classification of defects 5](#_Toc72966481)

[Risk Issues 6](#_Toc72966482)

[Test Environment 6](#_Toc72966483)

[Test Deliverables 6](#_Toc72966484)

[Test scenarios 7](#_Toc72966485)

[Bug tickets 7](#_Toc72966486)

[Reports 7](#_Toc72966487)

[Planning risks 7](#_Toc72966488)

[Glossary 9](#_Toc72966489)

# Introduction

This test plan describes the testing approach to test <https://api.meteo.lt/> site. Test plan summarizes the overall effort required to test api.meteo.lt. Test plan will contain the details of test to be run during the testing cycle.

# Test Strategy

Tests will test (scope):

* If we get an API response using all of the methods (/places, /places/{place-code}, /places/{place-code}/forecasts, /places/{place-code}/forecasts/{forecast-type});
* Documentation: if heading is present and if all the methods above are described;

Testing will be focused on each method and documentation parts separable.

The following item not to be tested:

* Style guidelines

# Preconditions

In order to test the functionality certain conditions must be met:

* Testing scenarios must be developed on time leaving enough time for tester to execute
* The testing site must function to be able test it

# 

# Suspension and Resumption criteria

Testing activities can be stopped:

* in case there is a critical bug that must be fixed before moving forward with testing activities;
* If the website in general is unavailable.

Testing activities can be continued as soon as the critical defect is fixed and website is fully reachable for testing

# 

# Item to be Tested

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Likelihood** | **Impact** | **Priority** |
| Parameters | High | High | 1 |
| Data type of response | High | High | 1 |
| Object | Medium | Medium | 2 |

# Item not to be Tested

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Likelihood** | **Impact** | **Priority** |
|  |  |  |  |
|  |  |  |  |

# Testing Approach

For testing approach will be used Acceptance testing type of test.

## Acceptance testing

Acceptance Testing is a formal testing type where the objective is to verify that a system meets its specified requirements from the user point of view. Most often the use cases or similar end to end business scenarios are tested here. This testing level should be executed by customer and there should be an Acceptance Testing Coordinator available to plan and monitor the testing. Customer should include end users or other authorized entities to determine whether to accept a system or component. The actual testing should be completed in the environment as production like as possible but not in production if the production is already in use.

## Test Tools

Postman – API testing tool

Selenium – Web browser automation tool

IntelliJ – for writing the automated test cases

Browser – Chrome

## Classification of defects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Severity | Description | Impact on testing | Actions | Analysis and fix schedule |
| Critical defect | Defect causes that important functionalities are not achievable, that data disappears, performance deteriorates or parts of the system crash | showstopper for testing | Example: Needs actions immediately | Example: Defect analysis should be started immediately, fix delivery time ASAP |
| High defect | Defect means that a functionality does not work according to specifications or does not work at all | Essential function not functional | Example: Needs actions soon | Example: Defect analysis should be started latest in one to two days, fix delivery in one week |
| Medium defect | With Medium severity defect it is meant that some part of the system does not work normally but needs some special actions in order to get all the normal functionalities | Some part of function is not working | Example: Can be accepted, actions to be decided in the Steering group | Example: Defect analysis should be started in one week, fix delivery in two weeks |
| Low defect | With a Low severity defect it is meant a defect that is mainly cosmetic and does not have any effect on functionality | Cosmetic error, e.g. wrong error message | Example: Can be accepted | Example: Defect analysis should be started in one week if there is time and higher severity defects are fixed, fix delivery in one month |

## Risk Issues

There are several risk issues possible regarding testing activities:

* Reliability of the web hosting service
* Poorly documented modules
* Failure of services

# Test Environment

A windows environment with Google chrome 90.0

## Test Deliverables

Testing will provide specific deliverables. These deliverables fall into two basic categories:

* Test Scenarios
* Reports

## Test scenarios

Are defined as any functionality that can be tested.

## Bug tickets

In case defect is found, bug ticket must be registered in Excel. Information about the defect must be as detailed as possible and should contain the following information:

* Bug description
* Time of detection
* Environment (OS, computer)
* Browser
* Steps to reproduce the bug
* Expected behavior
* Screenshots/videos

## Reports

After test cases are executed report (in html and json formats) will be generated showing which Test scenario failed and which ones passed.

# Planning risks

Following criteria is the likely project risks:

* Unavailability of Website: Testing will be delayed until the website is reestablished. Possible contingency can be to increase number of tests or reduce number of test scenarios.
* Unavailability of Testing software: this can be caused by various issues on the testers computer when needed software cannot be lunched. It can lead to delay test scenario execution. Possible contingency can be to increase number of tests or reduce number of test scenarios.
* Time problem: There may not be enough time to complete all the test scenarios. In that case lowest priority test scenarios can be skipped.
* Large number of defects: in case there are many bugs it is difficult to make proper testing.

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Acceptance testing** | Testing conducted to enable a user/customer to determine whether to accept a product. Normally performed to validate if the software meets a set of agreed acceptance points |
| **Automated testing** | Testing software without manual intervention using tools like Selenium. |
| **Bug** | A defect in a software which causes it to perform in an unintended way |
| **Pass/fail criteria** | Decision rues used to determine whether a test item (function) or feature has passed or failed a test |
| **Severity** | The degree of impact that a defect has on the development or operation of the system |
| **Testing** | The process of exercising software to verify that it satisfies specified requirements. |
| **Test Scenario** | Ay functionality that can be tested. |
| **Test environment** | Hardware and software environment in which tests will be ran |
| **Test plan** | Document describing the scope, approach, resources and intended testing activities. It identifies test items, the features to be tested and risks that might arise. |
| **Test tool** | Computer software used in the testing |
| **Tester** | Technically skilled professional who is involved in the testing of a component or a system. |