

Marketing Campaign Scheduling - Test Plan

Document History- To maintain a list of changes being made

version	Date	Author	Description of change
1.0	10/06/2024	Kuldeep	

Approvers List- To track who has reviewed and sign off on the Test plan

NAME	ROLE	APPROVER/REVIEWER	Date
Kuldeep	QA		10/06/2024

INTRODUCTION

1.1. Purpose

This document describes the testing approach for marketing campaign scheduling and its integration with the email template and recipient services, allowing users to send scheduled email campaigns at specific dates and times.

OBJECTIVE

- Verify that users can create, configure, and schedule email campaigns.
- Ensure that scheduled campaigns are sent accurately and on time.
- Test the integration with the email template and recipient services.
- Validate the ability to handle large volumes of emails efficiently.
- Confirm that users can edit and cancel scheduled campaigns.

Scope

The scope of this test plan includes:

- Functional Testing
- Integration Testing
- End-to-End Testing

2. Test Environment

- Docker Compose setup using `docker-compose-e2e.yml` for running all services.
- Ensure services are running and accessible via:
 - Campaign service: <http://localhost:7070/swagger-ui/index.html>
 - Email template service: <http://localhost:7071/swagger-ui/index.html>

- Recipients service: <http://localhost:7072/swagger-ui/index.html>

3. User Stories

- Create a new email campaign with a scheduled send time and date.
- Select a recipient list for a scheduled campaign.
- Choose an email template for a scheduled campaign.
- Edit the name of scheduled campaigns.
- Cancel a scheduled campaign.

4. Test Scenarios (End -to-End Test)

- Verify that a user can create a new email campaign with a scheduled send time and date.
- Verify that the campaign is created successfully.
- Ensure that a user can select a recipient list for their scheduled campaign.
- Verify the recipient list is associated with the campaign.
- Verify that a user can choose an email template for their scheduled campaign.
- Verify the email template is associated with the campaign.
- Ensure that a user can edit the name of their scheduled campaigns.
- Verify the campaign name is updated.
- Verify that a user can cancel a scheduled campaign. (The Delete API is not present in the Swagger documentation, and the Status key, which holds information about activation/deactivation, is missing from the model therefore have been running creating a campaign with 0 schedule time (-ve test case))
- Verify the campaign is canceled and will not be sent.

5. Test Scenarios (Integration Tests)

- Verify the integration between the Campaign Scheduling service and the Recipient service.
- Verify that the recipient list is fetched and associated correctly.
- Verify the integration between the Campaign Scheduling service and the Email Template service.
- Verify that the email template is fetched and associated correctly.

6. Automation Implementation

- Tools and Frameworks
 - Rest-assured and SRP Pattern (Each service class can run independently, allowing for easy modifications and further changes.)
 - Maven
 - TestNG
 - Java
 - Jackson

7. Code (Folder) Structure:

- **Main**
 - The method has been implemented according to the action of the APIs, allowing these methods to be used with the required test data.
 - Campaign service
 - Email template service
 - Recipient service
 - Constants
 - Models - Serialization and deserialization
 - Request and response specification.
 - Utils
- **Resources**
 - Testng.xml
- **Test**
 - **DataProvider** - Test data are managed here according to the test case.
 - End to end test case
 - Integration test cases
 - Each test class case can runs independently and as a suites also.