# BabyList Integration Test Plan

### Overview

This document is intended to outline the test strategy for Lovevery to test the integration with BabyList Registry product. Details are subject to change based on product implementation schedule.

*Intended For: Product Team, in order to agree on testing process*

Objective

Potential or existing customers will be able to register for Lovevery subscription products through the BabyList Registry Sites. These registries will provide opportunities for friends and family of the customer to purchase one or two subscription kits for the customer. An activation code or codes will be provided to the customer and Lovevery should identity whether they are an existing customer and then activate either the new or existing customer with one or two subscription kits based on what was purchased.

Product Phases

**Development**

i. Unit Tests to be written by developers

ii. API test and automation written and performed

iii. UI Automation to begin while in development phase, up to point of page object automation

**QA/Staging**

i. Complete automation tests based on page elements when they’re present

ii. Unit tests should be regularly passing or fail the build

iii. End-to-End testing performed

iv. Load and security testing and regression testing is performed.

**Production**

i. All automated testing (Unit, API, UI) should be consistently passing

ii. Components are functional and free from major defects from start to end of process

iii. Customers are using product without major issues

### II. Test Approach

Types of Testing to Be Performed

i. Unit testing will be written and performed as early as possible by developers to ensure code sanity at the most basic levels.

ii. Integration testing will be performed by QA. Need to determine if testing will be available for BabyList and LogicBroker.

iii. Load Testing performed by QA using API stress tool to measure where Orders API threshold lies.

iv. Security Testing performed by QA to check for process and site vulnerability.

v. Regression Testing performed by QA running existing automation tests to ensure existing functionality is not compromised.

Ordering approach

Testing will be performed on the basis of highest perceived risk level and criticality as identified by the product team and QA. The testing with highest risk basis will be performed and automated first and within the shortest time possible as identified by using the following risk matrix as a heuristic.

| Probability | Critical Impact | Marginal Impact | Negligible Impact |
| --- | --- | --- | --- |
| Frequently Occur | CRITICAL | HIGH | MEDIUM |
| Likely to Occur | HIGH | HIGH | MEDIUM |
| Occasionally Occur | HIGH | MEDIUM | LOW |
| Improbable | MEDIUM | LOW | LOW |

Automation approach

i. QA involvement during planning phase to help identify possible testing challenges and being outlining automation. Developers to add automation id’s to pages for any item that

a. Has verifiable data, i.e. pricing, customer name

b. Can and should be interacted with by automation, i.e. input fields, buttons

ii. UI Automation work to begin once functionality has been defined and completed when page elements are available to be interpreted by automated framework.

iii. API Automation work to begin once the api is considered dev complete, or valid endpoint and expected inputs and outputs are provided to QA

iv. Automation will follow the Just Barely Good Enough principle with concentration on tests that are considered to provide the highest value within the risk matrix.

### III. Scope

In Scope

i. Features developed by Lovevery, including Orders API, Notification Service and Subscription Manager

Out Of Scope

i. Other areas of Lovevery not affected by this change.

ii. Other areas of BabyList Registry product.

iii. Other areas of Logic Broker.

iv. Email Provider Service.

### IV. Test Strategy

Baby List to Orders API

*ASSUMPTIONS*

1. Registry can be submitted to Baby List for Lovevery products

2. Baby List consumer can purchase the product registered for by another user.

3. Upon purchase, correct order info is passed successfully to LogicBroker

4. LogicBroker sends the push notification via ebook to the Orders API

*ACTION ITEMS*

1. Test Payload from LogicBroker can be passed to Orders API

2. Test mandatory fields are required by LogicBroker

3. Test orders API creates subscription generation

4. Test orders API saves correct information in databases

*HIGH RISKS*

1. BabyList or Logic Broker unavailable

2. Data passed is not supported by Lovevery (payment method, currency, shipping country)

3. Data passed incorrectly or invalid data passed from 3rd party services

Subscription Activation

*ASSUMPTIONS*

1. Email Provider service will deliver notification if sent properly from the service

2. Actor may be given a code, a link or use the app

3. List of test activation codes is provided to QA

*ACTION ITEMS*

1. Test that code can be provided via email in embedded link and redirect user to Lovevery when clicked and modal appears.

2. Test that url can be navigated to directly and activation code can be provided and modal appears.

3. Test that activation code can be provided through the mobile application and modal appears.

4. Test that a new customer can activate a subscription.

5. Test that an existing customer can activate a subscription.

6. Test that multiple codes can be added to the same customer twice to equal one year of play.

*HIGH RISKS*

1. Subscription is activated for the wrong customer.

2. Subscription appears activated but does not register to subscription manager.

3. Subscription cannot be activated through all possible means of activation.

4. Customer type prevents activation (new vs existing)

5. Customer cannot receive all subscriptions purchased by gifters.

Subscription service

*ASSUMPTIONS*

1. Registry and Activation have already been successfully performed.

ACTION ITEMS

1. Test that each subscription type is honored and corresponds to correct customer.

2. Test that multiple subscriptions, up to two, can be added to an account

3. Test that proper prices, products and delivery dates are honored.

4. Test that proper subscription is applied regardless of activation method.

5. Test that multiple gifters can give subscriptions to the same customer.

*HIGH RISKS*

1. Subscriptions do not have proper pricing, dates or products

2. Subscription was applied to proper customer.

3. Multiple subscriptions can be properly applied, up to required limit.

### V. Test Deliverables

Project Deliverables

1. Test plan provided during development phase

2. High Level Test cases defined during development

3. Bug Reports at end of each sprint during development and qa/staging

4. Automated Tests of smoke and regression functionality, as well as unit tests, during development and qa/staging

### VI. Supported Environments

Lovevery Environments

Smoke automation will be run against test and staging environment after each deployment. Regression suite will run nightly.

Desktop Browser Support Matrix

|  | Windows | Mac | Linux |
| --- | --- | --- | --- |
| Firefox | 79.0.3 |  |  |
| Chrome |  |  |  |
| IE |  |  |  |
| Safari |  |  |  |

Mobile Device Support Matrix

|  | App Version 1 | App Version 2 | App Version 3 |
| --- | --- | --- | --- |
| iPhone 8 |  |  |  |
| iPhone 11 |  |  |  |
| Samsung S10 |  |  |  |
| Samsung Pixel 4 |  |  |  |