



# Email Import

with RChilli

June 2024 | Version 1.0

Copyright © 2024, RChilli Inc.



## Purpose Statement

This document details integrating **Email Import** use case leveraging RChilli Email Integration.

## Disclaimer:

*Licensed Materials - Property of RChilli*

*For information about RChilli trademarks, copyrights, and patents, refer to the RChilli Intellectual Property page*

*(<https://www.rchilli.com/>) on the RChilli website. All other trademarks and registered trademarks are the property of their respective holders.*

*No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of RChilli Inc. Under the law, reproducing includes translating into another language or format.*

*Every effort has been made to ensure that the information in this manual is accurate. RChilli Inc. is not responsible for printing or clerical errors. Information in this document is subject to change without notice.*

*If you have any comments or feedback on our documentation, please send them to us at [support@rchilli.com](mailto:support@rchilli.com)*

## Contents

|   |           |
|---|-----------|
| <b>Introduction .....</b>   | <b>4</b>  |
| <b>Key Components and Flow.....</b>                                     | <b>5</b>  |
| <b>Benefits .....</b>   | <b>5</b>  |
| <b>Example Scenario .....</b>   | <b>6</b>  |
| <b>User Acceptance Testing (UAT) .....</b>                              | <b>7</b>  |
| <b>UAT Scenarios.....</b>   | <b>7</b>  |
| <b>UAT Scenario 1: Email Forwarding and Processing.....</b>             | <b>7</b>  |
| <b>UAT Scenario 2: JSON Data Storage.....</b>                           | <b>7</b>  |
| <b>UAT Scenario 3: Error Handling for Unsupported Formats.....</b>      | <b>8</b>  |
| <b>UAT Scenario 4: Performance Testing for High Volume Emails .....</b> | <b>8</b>  |
| <b>Test Cases .....</b>   | <b>9</b>  |
| <b>Test Case 1: Email Forwarding Test.....</b>                          | <b>9</b>  |
| <b>Test Case 2: JSON Data Storage Test .....</b>                        | <b>9</b>  |
| <b>Test Case 3: Unsupported Email Format Handling Test.....</b>         | <b>9</b>  |
| <b>Test Case 4: High Volume Email Processing Test.....</b>              | <b>10</b> |
| <b>Test Case 5: Email Attachment Handling Test .....</b>                | <b>10</b> |
| <b>Test Case 6: Email Without Attachments Test .....</b>                | <b>10</b> |
| <b>Test Case 7: Handling Large Attachment Files Test .....</b>          | <b>11</b> |
| <b>Test Case 8: Mixed Content Email Test .....</b>                      | <b>11</b> |
| <b>Test Case 9: Invalid Email Address Test.....</b>                     | <b>12</b> |
| <b>Test Case 10: Network Failure Handling Test .....</b>                | <b>12</b> |
| <b>Test Case 11: Duplicate Email Handling Test.....</b>                 | <b>13</b> |
| <b>Test Case 12: Email Content Parsing Test.....</b>                    | <b>13</b> |
| <b>Miscellaneous.....</b>   | <b>14</b> |
| <b>Integration Estimation.....</b>                                      | <b>14</b> |
| <b>Appendix .....</b>   | <b>15</b> |



## Introduction

The **Email Import** feature simplifies the process of adding candidates and jobs to your ATS by forwarding resumes and job descriptions via email.

The recruiter receives resumes in their email, forwards them to the RChilli Email Importer, and the system processes the content. The parsed data is then sent to a webhook, from which the user reads the JSON response and stores the information in their database.



## Key Components and Flow

### 1. Email Reception:

- **Action:** Recruiter receives a resume or job description via email.
- **Response:** Email is forwarded to the designated RChilli email address.

### 2. RChilli Email Importer:

- **Action:** Processes the email content and attachments.
- **Response:** Extracted data is sent to a webhook.

### 3. Webhook Integration:

- **Action:** Receives the parsed JSON data.
- **Response:** Data is processed and stored in the database.

### 4. Data Storage:

- **Action:** The parsed resume data is stored in the database, associated with the job application.
- **Functionality:** This data can be used for backend processing, reporting, and future reference.

## Benefits

1. **Efficiency:** Increases resume processing efficiency by 40%.
2. **Automation:** Reduces manual entry by automating the data extraction process.
3. **Accuracy:** Ensures accurate data parsing and storage.



## User Stories

### 1. Forward Resumes via Email

- a. As a recruiter, I want to forward resumes I receive via email to the RChilli Email Importer so that they can be automatically processed and added to the ATS.

### 2. Parse Email Content

- a. As a system, I need to parse the content of forwarded emails and attachments to extract relevant candidate and job information.

### 3. Store Parsed Data

- a. As a system, I need to store the extracted data in the database to maintain accurate candidate and job records.

### 4. Receive Webhook Response

- a. As a user, I want to receive a JSON response from the webhook after processing so that I can verify and store the information in my ATS.

## Example Scenario

1. **Scenario:** A recruiter named Jamie receives a resume via email.
2. **Action:** Jamie forwards the email to the RChilli Email Importer.
3. **Response:** The RChilli system processes the email and sends the parsed data to the webhook.
4. **Outcome:** The parsed data is stored in the ATS, and Jamie verifies the information



## User Acceptance Testing (UAT)

The **Email Import** UAT plan ensures the feature meets user requirements and business goals.

It validates that candidates can apply for jobs efficiently by send their resumes via email, which are then processed to populate application forms or stored in the database.

### UAT Scenarios

#### UAT Scenario 1: Email Forwarding and Processing

- **Objective:** Verify that an email with a resume can be forwarded and processed **successfully**.
- **Preconditions:** Recruiter has an email with a resume.
- **Steps:**
  - a. Forward the email to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The email content is parsed, and the JSON response is received.
- **Acceptance Criteria:**
  - a. Email is processed without errors.
  - b. JSON response includes correctly parsed fields.

#### UAT Scenario 2: JSON Data Storage

- **Objective:** Verify that the parsed JSON data can be stored in the database.
- **Preconditions:** Parsed JSON data from the webhook.
- **Steps:**
  - a. Read the JSON response from the webhook.
  - b. Store the data in the database.
- **Expected Result:** Data is stored accurately in the ATS database.
- **Acceptance Criteria:**



- a. Data is stored without errors.
- b. Database records are accurate.

### UAT Scenario 3: Error Handling for Unsupported Formats

- **Objective:** Verify that the system handles unsupported email formats gracefully.
- **Preconditions:** Email with an unsupported format.
- **Steps:**
  - a. Forward an email with an unsupported format to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result: Appropriate error message is generated.**
- **Acceptance Criteria:**
  - a. System does not crash.
  - b. Clear error messages are provided.

### UAT Scenario 4: Performance Testing for High Volume Emails

- **Objective:** Ensure the system can handle a high volume of emails efficiently.
- **Preconditions:** Multiple emails forwarded to the system.
- **Steps:**
  - a. Forward multiple emails to the RChilli Email Importer.
  - b. Monitor processing time and system response.
- **Expected Result: All emails are processed within acceptable time limits.**
- **Acceptance Criteria:**
  - a. Processing time is within acceptable limits.
  - b. System performance remains stable.





## Test Cases

### Test Case 1: Email Forwarding Test

- **Objective:** Verify that an email can be forwarded and processed successfully.
- **Preconditions:** Recruiter has an email with a resume.
- **Steps:**
  - a. Forward the email to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The email content is parsed, and the JSON response is received.
- **Pass Criteria:** Email is processed without errors, and the JSON response includes correctly parsed fields.

### Test Case 2: JSON Data Storage Test

- **Objective:** Verify that the parsed JSON data can be stored in the database.
- **Preconditions:** Parsed JSON data from the webhook.
- **Steps:**
  - a. Read the JSON response from the webhook.
  - b. Store the data in the database.
- **Expected Result:** Data is stored accurately in the ATS database.
- **Pass Criteria:** Data is stored without errors, and database records are accurate.

### Test Case 3: Unsupported Email Format Handling Test

- **Objective:** Verify that the system handles unsupported email formats gracefully.
- **Preconditions:** Email with an unsupported format.
- **Steps:**
  - a. Forward an email with an unsupported format to the RChilli Email Importer.
  - b. Monitor the webhook for the response.



- **Expected Result:** Appropriate error message is generated.
- **Pass Criteria:** System does not crash, and clear error messages are provided.

## Test Case 4: High Volume Email Processing Test

- **Objective:** Ensure the system can handle a high volume of emails efficiently.
- **Preconditions:** Multiple emails forwarded to the system.
- **Steps:**
  - a. Forward multiple emails to the RChilli Email Importer.
  - b. Monitor processing time and system response.
- **Expected Result:** All emails are processed within acceptable time limits.
- **Pass Criteria:** Processing time is within acceptable limits, and system performance remains stable.

## Test Case 5: Email Attachment Handling Test

- **Objective:** Verify that the system correctly handles emails with multiple attachments, focusing on the supported resume and job description formats.
- **Preconditions:** Recruiter has an email with multiple attachments (some supported, some unsupported).
- **Steps:**
  - a. Forward the email with multiple attachments to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** Only the supported attachments are processed, and the unsupported ones are ignored.
- **Pass Criteria:**
  - a. JSON response includes data from supported attachments.
  - b. Unsupported attachments are ignored without causing errors.

## Test Case 6: Email Without Attachments Test

- **Objective:** Ensure the system handles emails without attachments gracefully.



- **Preconditions:** Recruiter has an email without any attachments.
- **Steps:**
  - a. Forward the email without attachments to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The system identifies the absence of attachments and logs an appropriate message.
- **Pass Criteria:**
  - a. System logs a message indicating no attachments were found.
  - b. No errors are generated.

### Test Case 7: Handling Large Attachment Files Test

- **Objective:** Verify the system's ability to handle large file sizes in email attachments.  
**Preconditions:** Recruiter has an email with a large attachment file.
- **Steps:**
  - a. Forward the email with a large attachment to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The large attachment is processed within acceptable time limits.
- **Pass Criteria:**
  - a. Attachment is processed without errors.
  - b. Processing time is within acceptable limits.

### Test Case 8: Mixed Content Email Test

- **Objective:** Ensure the system correctly parses emails with both resume and job description attachments.
- **Preconditions:** Recruiter has an email with a resume and a job description as attachments.
- **Steps:**
  - a. Forward the email with both resume and job description attachments to the RChilli Email Importer.

- b. Monitor the webhook for the response.
- **Expected Result:** Both attachments are parsed, and the respective data is included in the JSON response.
- **Pass Criteria:**
  - a. JSON response includes parsed data from both attachments.
  - b. Data is accurate and correctly identified as resume or job description.

## Test Case 9: Invalid Email Address Test

- **Objective:** Verify the system's response to emails forwarded from an unregistered or invalid email address.
- **Preconditions:** Email forwarded from an invalid or unregistered email address.
- **Steps:**
  - a. Forward the email from an invalid or unregistered address to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The system identifies the invalid email address and logs an appropriate error message.
- **Pass Criteria:**
  - a. System logs a message indicating the email is from an invalid or unregistered address.
  - b. No further processing is done on the email content.

## Test Case 10: Network Failure Handling Test

- **Objective:** Verify how the system handles network failures during email processing.
- **Preconditions:** Simulate a network failure during the forwarding or webhook processing stage.
- **Steps:**
  - a. Forward the email to the RChilli Email Importer.
  - b. Simulate a network failure during the process.



- **Expected Result:** The system retries the operation or logs an appropriate error message.
- **Pass Criteria:**
  - a. System retries the operation as per the retry policy.
  - b. If retries fail, an error message is logged.

## Test Case 11: Duplicate Email Handling Test

- **Objective:** Verify the system's handling of duplicate emails.
- **Preconditions:** Forward the same email multiple times to the system.
- **Steps:**
  - a. Forward the same email multiple times to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The system processes the email once and identifies duplicates.
- **Pass Criteria:**
  - a. Duplicate emails are identified and logged.
  - b. Email content is processed only once.

## Test Case 12: Email Content Parsing Test

- **Objective:** Verify the system's ability to parse the email body content when resumes are embedded in the email rather than as attachments.
- **Preconditions:** Recruiter has an email with the resume content embedded in the email body.
- **Steps:**
  - a. Forward the email with embedded resume content to the RChilli Email Importer.
  - b. Monitor the webhook for the response.
- **Expected Result:** The system extracts and parses the embedded resume content.
- **Pass Criteria:**
  - a. Embedded resume content is accurately parsed.
  - b. JSON response includes data from the embedded content



## Miscellaneous

The integration depends on your application workflow. There might be different steps involved in your integration which may vary from application to application.

This document indicates basic steps which are involved, you can review your application and add other UAT, and test cases based upon your needs.

## Integration Estimation

**Disclaimer:** This estimation is based on a sample application with simple workflow. This may vary depending on the complexity of your application and the skill set/experience of the team involved in the integration.

1. **Development** – (14hr – 17hr)
  - a. Email Reception and Forwarding: 1hr
  - b. Email Parsing using RChilli: 2hr
  - c. Webhook Integration and JSON Processing: 5hr
  - d. Storing Data in Database: 3hr
  - e. Unit Test Cases: 2-3hr
  - f. Bug Fixing and Optimization: 2-3hr
2. **QA** – (5hr -7hr)
  - a. UAT: 1-1.5hr
  - b. Executing Test Cases: 4-5.5hr



## Appendix

Resume Parser API Details:

[https://docs.rchilli.com/kc/c\\_RChilli\\_resume\\_parser\\_rest\\_API](https://docs.rchilli.com/kc/c_RChilli_resume_parser_rest_API)

Get API key and Endpoints:

[https://docs.rchilli.com/kc/c\\_RChilli\\_resume\\_parser\\_API\\_authentication](https://docs.rchilli.com/kc/c_RChilli_resume_parser_API_authentication)

JD Parser API Detail:

[https://docs.rchilli.com/kc/c\\_RChilli\\_JD\\_parser](https://docs.rchilli.com/kc/c_RChilli_JD_parser)

Get API key and Endpoints:

[https://docs.rchilli.com/kc/c\\_RChilli\\_JD\\_parser\\_API\\_End\\_points](https://docs.rchilli.com/kc/c_RChilli_JD_parser_API_End_points)

Resume Parser - Email Integration:

[https://docs.rchilli.com/kc/c\\_RChilli\\_resume\\_parser\\_Email\\_inbox\\_inte](https://docs.rchilli.com/kc/c_RChilli_resume_parser_Email_inbox_inte)

JD Parser – Email Integration:

[https://docs.rchilli.com/kc/c\\_JD\\_Parser\\_email\\_integration](https://docs.rchilli.com/kc/c_JD_Parser_email_integration)

Postman Integration and Sample Code:

<https://documenter.getpostman.com/view/6003669/Szmmh1GQX?version=latest#0b486ceb-a8b6-496b-b53b-73c4adbc6987>

Resume Parser Response Schema:

[https://docs.rchilli.com/kc/c\\_RChilli\\_resume\\_parser\\_response\\_Schema](https://docs.rchilli.com/kc/c_RChilli_resume_parser_response_Schema)

JD Parser Error Code:



[https://docs.rchilli.com/kc/c\\_RChilli\\_JD\\_parser\\_error\\_code](https://docs.rchilli.com/kc/c_RChilli_JD_parser_error_code)

Resume Parser Error Code:

[https://docs.rchilli.com/kc/c\\_RChilli\\_Resume\\_parser\\_error\\_code](https://docs.rchilli.com/kc/c_RChilli_Resume_parser_error_code)

Language Supported:

<https://www.rchilli.com/languages>