



Capture and Load Candidates from Job Board

with RChilli

June 2024 | Version 1.0

Copyright © 2024, RChilli Inc.



Purpose Statement

This document details integrating the **Capture and Load Candidates from Job Board** use case leveraging a RChilli browser plugin.

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



Contents

Introduction	4
Key Components and Flow	5
Benefits	6
User Stories	7
User Acceptance Testing (UAT)	8
UAT Scenario 1: Auto Capture Candidate Profile	8
UAT Scenario 2: Edit the Parsed Candidate Information	8
UAT Scenario 3: Load into the HRIS/ATS	9
UAT Scenario 4: Error Handling for Unsupported Job Boards	9
Test Cases	10
Test Case 1: Auto Capture Candidate Profile	10
Test Case 2: Edit the Parsed Candidate Information	11
Test Case 3: Load into the HRIS/ATS	11
Test Case 4: Error Handling for Unsupported Job Boards	12
Test Case 5: Validate Browser Compatibility	12
Test Case 6: Ensure Secure Data Transmission	13
Test Case 7: Validate Extension Response Time	13
Test Case 8: Verify Handling of Incomplete Profiles	14
Test Case 9: Verify Extension Updates and Maintenance	14
Miscellaneous	15
Integration Estimation	15



Introduction

The **Capture and Load Candidates from Job Board** feature enables recruiters to directly add candidates from given job boards (LinkedIn, Monster, CareerBuilder etc.) as soon as the RChilli browser plugin identifies that a candidate profile is opened in browser. Utilizing the Resume Parser the system automates the extraction of the candidate information and then loads it to your HRIS/ATS, significantly reducing the time and effort required for recruiters to manually add candidates.

Capture and Load Candidates from Job Board simplifies load candidate process by:

1. **Capture Candidate:** RChilli browser plugin automatically captures candidate information from the Job Board.
2. **Parsing Resume:** Then RChilli browser plugin instantly processes the candidate information using the RChilli Parser.
3. **Load to HRIS/ATS:** The recruiter can review and decide to load the candidate information into the HRIS/ATS



Key Components and Flow

1. Recruiter Interaction:

- **Action:** Recruiters open the RChilli browser plugin and navigate to a candidate profile on the job board.
- **Response:** This initiates the RChilli browser plugin to capture the candidate information from the candidate profile.

2. Candidate Information Upload:

- **Action:** The selected candidate information is sent to the Resume Parser for parsing.
- **Response:** Resume is parsed, and an option is given to the recruiter to add this or ignore, if recruiter adds it then following candidate information is displayed in RChilli browser plugin
 - personal details (name, contact information)
 - education, experience
 - known languages
 - certifications
 - skills.
- **Functionality:** Recruiters can review and modify any of the given information as necessary.

3. Load into the HRIS/ATS:

- **Action:** Recruiter triggers the RChilli browser plugin to load the candidate information into the HRIS/ATS.
- **Response:** The RChilli browser plugin loads all the candidate information to the HRIS/ATS automatically.
- **Functionality:** The log is displayed to the recruiter, where they can check the status of the candidate.

4. Data Storage:

- **Action:** Logs which include File Name, Status, Date, Message is stored.



- **Functionality:** This data can be used for status, reporting, and future reference.

Benefits

1. **Time Efficiency:** This reduces the time recruiters spend manually adding the candidate in the ATS.
2. **User-Friendly:** Enhances the recruiter experience by reducing the complexity of manually adding a candidate. Recruiters simply visit the candidate profile while the RChilli browser plugin is open, and the candidate is automatically added to the HRIS/ATS.



User Stories

1. Capture Candidate

- **As a system**, I need to capture the candidate information from the job board when candidate's profile is open in the browser.

2. Parse Candidate Information and Map into Fields

- **As a system**, I need to parse the captured candidate information, and display the parsed information.

3. Load into the HRIS/ATS

- **As a recruiter**, I should be able to review/update the displayed candidate information and load it into the **HRIS/ATS**.

4. Display Logs

- **As a system**, I need to store the logs (File Name, Date, Status etc.) of Candidates which have been loaded into the HRIS/ATS.

Example Scenario

1. **Scenario:** A recruiter named Alex finds a candidate on the job board and wants to load the candidate information to the HRIS/ATS.
2. **Action:** Alex opens RChilli browser plugin.
3. **Response:** RChilli browser plugin automatically captures and parses the candidate information and displays the parsed information to Alex.
4. **Action 2:** Alex reviews the parsed information and triggers the RChilli browser plugin to load candidate information into the HRIS/ATS.
5. **Response 2:** RChilli browser plugin loads the candidate information into the HRIS/ATS automatically.
6. **Outcome:** Alex finds the candidate with all the information in the HRIS/ATS immediately.



User Acceptance Testing (UAT)

The **Capture and Load Candidates from Job Board** UAT plan ensures the feature meets user requirements and business goals. It validates the candidate profile is exported to the ATS.

UAT Scenarios

UAT Scenario 1: Auto Capture Candidate Profile

- **Objective:** Verify that candidate's profile is auto captured and parsed
- **Preconditions:** Candidate Profile is opened on the job board.
- **Steps:**
 1. Open the RChilli browser plugin.
 2. Wait for the RChilli browser plugin to capture and parse the candidate information.
 3. Verify that candidate information is parsed and displayed in the RChilli browser plugin
- **Expected Result:** The candidate's profile is parsed, and its information is displayed in the RChilli browser plugin.
- **Acceptance Criteria:**
 - Candidate's Profile is Parsed.
 - All the candidate information is displayed inside the RChilli browser plugin.

UAT Scenario 2: Edit the Parsed Candidate Information

- **Objective:** Validate that the Recruiter can edit any field of the parsed candidate information in RChilli browser plugin.
- **Preconditions:** Candidate Profile is opened on the job board.
- **Steps:**
 1. Open the RChilli browser plugin.



2. Wait for the RChilli browser plugin to capture and parse the candidate information.
 3. Verify that candidate information is parsed and displayed in the RChilli browser plugin
 4. After the parsed candidate information is displayed in the RChilli browser plugin verify that fields are editable and any new information can be added or any existing information can be removed
- **Expected Result:** The candidate fields are populated with the parsed candidate information, and they are editable.
 - **Acceptance Criteria:**
 - Parsed candidate fields are editable inside the RChilli browser plugin.

UAT Scenario 3: Load into the HRIS/ATS

- **Objective:** Validate the candidate information is added to the ATS.
- **Preconditions:** Candidate Profile is opened on the job board.
- **Steps:**
 - Open the RChilli browser plugin.
 - Wait for the RChilli browser plugin to capture and parse the candidate information.
 - RChilli browser plugin automatically displays the parsed candidate information
 - After reviewing trigger, the load into the HRIS/ATS.
 - Review the candidate information in the HRIS/ATS
- **Expected Result:** The candidate is available in the HRIS/ATS now.
- **Acceptance Criteria:** All Candidate information is Exported to HRIS/ATS

UAT Scenario 4: Error Handling for Unsupported Job Boards

- **Objective:** Ensure the system handles unknow websites.



- **Preconditions:**
 1. There will be a predefined list of websites for which the RChilli browser plugin will do the candidate profile scan and capture
- **Steps:**
 1. Navigate to the any website which is not present in the list.
 2. Open the RChilli browser plugin.
 3. Wait for few seconds for the RChilli browser plugin to not take any action.
- **Expected Result:** The RChilli browser plugin should not capture any data from the website and should not parse it.
- **Acceptance Criteria:** RChilli browser plugin does not display any parsed information.

Test Cases

Test Case 1: Auto Capture Candidate Profile

- **Objective:** Verify that candidate's profile is auto captured and parsed.
- **Preconditions:**
 1. Candidate profile is opened on the job board.
 2. RChilli browser plugin is installed and active.
- **Steps:**
 1. Open the candidate profile on the job board.
 2. Launch the RChilli browser plugin.
 3. Wait for the RChilli browser plugin to capture and parse the candidate information.
 4. Verify that the candidate information is parsed and displayed in the RChilli browser plugin
- **Expected Result:** The candidate's profile is parsed, and its information is displayed in the RChilli browser plugin.
- **Pass Criteria:**



1. Candidate's profile is parsed
2. All candidate information is displayed inside the RChilli browser plugin.

Test Case 2: Edit the Parsed Candidate Information

- **Objective:** Validate that the recruiter can edit any field of the parsed candidate information in the RChilli browser plugin.
- **Preconditions:**
 1. Candidate profile is opened on the job board.
 2. RChilli browser plugin is installed and active.
- **Steps:**
 1. Open the candidate profile on the job board.
 2. Launch the RChilli browser plugin.
 3. Wait for the RChilli browser plugin to capture and parse the candidate information.
 4. Verify that the candidate information is parsed and displayed in the RChilli browser plugin.
 5. Edit the fields of the parsed candidate information and add new information or remove existing information.
- **Expected Result:** The candidate fields are populated with the parsed candidate information, and they are editable.
- **Pass Criteria:**
 1. Parsed candidate fields are editable inside the RChilli browser plugin.
 2. Any new information can be added, and existing information can be removed.

Test Case 3: Load into the HRIS/ATS

- **Objective:** Validate the candidate information is added to the ATS.
- **Preconditions:**
 1. Candidate profile is opened on the job board.



2. RChilli browser plugin is installed and active.
- **Steps:**
 1. Launch the RChilli browser plugin.
 2. Wait for the RChilli browser plugin to capture and parse the candidate information.
 3. Review the parsed candidate information displayed in the RChilli browser plugin.
 4. Trigger the load into the HRIS/ATS.
 5. Verify the candidate information in the HRIS/ATS.
 - **Expected Result:** The candidate is available in the HRIS/ATS.
 - **Pass Criteria:** All candidate information is exported to HRIS/ATS.

Test Case 4: Error Handling for Unsupported Job Boards

- **Objective:** Ensure the system handles unknown websites.
- **Preconditions:** There is a predefined list of websites for which the RChilli browser plugin will do the candidate profile scan and capture.
- **Steps:**
 1. Navigate to any website which is not present in the list.
 2. Launch the RChilli browser plugin.
 3. Wait a few seconds for the RChilli browser plugin to not take any action.
- **Expected Result:** The RChilli browser plugin should not capture any data from the website and should not parse it.
- **Pass Criteria:** RChilli browser plugin does not display any parsed information.

Test Case 5: Validate Browser Compatibility

- **Objective:** Ensure the browser extension works across different browsers.
- **Preconditions:** RChilli Browser Extension installed on various browsers (Chrome, Firefox, Edge).



- **Steps:**
 1. Open a candidate profile in different browsers.
 2. Open the RChilli Browser Extension and capture candidate data.
- **Expected Result:** Extension should function correctly across all supported browsers.
- **Pass Criteria:** Consistent performance and functionality in all tested browsers.

Test Case 6: Ensure Secure Data Transmission

- **Objective:** Ensure that data transmission between the browser extension and ATS is secure.
- **Preconditions:** Secure connections configured (HTTPS).
- **Steps:**
 1. Capture and export candidate data.
 2. Monitor the data transmission process.
- **Expected Result:** Data is transmitted securely without any breaches.
- **Pass Criteria:** No data breaches or insecure transmissions detected.

Test Case 7: Validate Extension Response Time

- **Objective:** Measure the response time of the browser extension during candidate data capture and export.
- **Preconditions:** Browser extension installed and operational.
- **Steps:**
 1. Open a candidate profile and trigger data capture.
 2. Measure the time taken for data capture and export processes.
- **Expected Result:** Processes should be completed within acceptable response time limits.
- **Pass Criteria:** Response times meet predefined performance criteria.



Test Case 8: Verify Handling of Incomplete Profiles

- **Objective:** Ensure the system handles incomplete candidate profiles gracefully.
- **Preconditions:** Incomplete candidate profiles on job boards.
- **Steps:**
 1. Open an incomplete candidate profile.
 2. Attempt to capture and export the profile data.
- **Expected Result:** System should handle incomplete data without errors.
- **Pass Criteria:** Incomplete profiles are processed correctly, with appropriate notifications for missing data.

Test Case 9: Verify Extension Updates and Maintenance

- **Objective:** Ensure the browser extension can be updated and maintained without affecting functionality.
- **Preconditions:** Older version of the browser extension installed.
- **Steps:**
 1. Update the browser extension to the latest version.
 2. Verify that all features function correctly post-update.
- **Expected Result:** Extension updates should not disrupt functionality.
- **Pass Criteria:** Successful update with all features working correctly.



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 (54hr – 51hr)

- Data Selection - 2hr
- Data Filtration - 2hr
- Parse using RChilli AI – 1hr
- Read Json, Mapping and populate on UI – 16hr
- Export to ATS - 10hr
- Logs - 4hr
- Unit Test cases – 6hr
- Bug fixing and optimization – 3-4hr

2. QA – (8hr -10hr)

- UAT – 1-2hr
- Executing Test cases – 7-8hr