

Integration Document: RChilli Bulk Upload 8.0



What you will find...

- [*?* Introduction](#)
- [*?* RChilli Bulk Upload](#)
- [*?* Resume JSON Response on Webhook](#)
- [*?* Sample Code](#)
- [*?* Error Code](#)



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.

[RChilli Release Note Version 8.0.0](#)

If you have any comments or feedback on our documentation, please send them to us at: support@rchilli.com



Introduction

RChilli Inc., one of the leading Hiring AI tool and Analytics Solutions Providers, has helped businesses in over 30 countries reach their full potential. We provide a web API to parse resumes which converts resumes of any format to structured JSON.

RChilli Bulk Upload

RChilli Bulk Upload parsing gives you the ability to parse bulk resumes files. We provide you FTP details to upload your resumes. We process all your uploaded resumes on provided FTP and post their JSON output to your register webhook page.

RChilli Bulk Upload is a 4-step process:

Step 1: Request support@rchilli.com for FTP detail to upload files.

Step 2. Create **HTTP POST** capturing pages (**webhook page**) which should be capable of capturing the JSON and be publicly accessible.

Step 3. Register your web hook url at B2b panel and get ftp details

URL: <https://myaccount.rchilli.com/>

- 1) Select Integration (Demo) from the menu
- 2) Select Bulk Upload Integration option



Resume Parsing 8.0.0 < Bulk Upload Integration

JD Parsing 3.1 Resume Parsing JD Parsing Sample Code and Documentation

Taxonomy 3.0

Search and Match 3.0 ▾

Bulk Parsing (Demo)

Bulk Upload Integration

Email Inbox Integration

| | | | |
|----------------------|--|--|------------------------------------|
| FTP URL | <input type="text"/> | FTP Port | <input type="text"/> |
| FTP User name | <input type="text"/> | FTP Password | <input type="password"/> |
| API URL * | <input type="text" value="https://rest.rchilli.com/RChilliParser/Rchilli/parseRes"/> | Version * | <input type="text" value="8.0.0"/> |
| Webhook URL * | <input type="text"/> | <input type="button" value="Add Details"/> | |

Step 4. Access FTP with provided details, create folders for SubUserID and Upload files/zip files containing resumes into respective folders. We will process all files uploaded on provided FTP and post JSON output to your registered web hook pages.

Points to remember:

- 1. Zip File:** Zip file must contain **only resume files or one folder** containing resume files and it **must not contain any folder/subfolder hierarchy**.
- 2. Folders Name:** Create folders on FTP main folder with name as **your SubUserIDs**.

We will return the sub **folder name** inside the main FTP folder as **SubUserID** in our JSON response. If you upload files directly on the main folder then the userkey will be returned as **SubUserID**. You can upload Zip files directly to **FTP main folder** and we will return the zip file name



as **SubUserID**. (*Best way is to create a folder with your SubUserID and upload zipFile or files in it*)

E.g. Suppose you have a folder structure which looks like the following:

Folder Structure

FTP Access Folder (root folder)

Client1 (folder name will be returned as **SubUserID** in JSON for all files under it)

ZipFiles.zip

Resume Files...

Client2 (folder name will be returned as **SubUserID** in JSON for all files under it)

ZipFiles.zip

Resume Files...

Client3.zip (zipfile name will be returned as **SubUserID** in JSON for all files under it)

Resume file1 (**userKey** will be returned as **SubUserID** in JSON for files in the main folder)

Resume file2 (**userKey** will be returned as **SubUserID** in JSON for files in the main folder)

- 3. Zip File Size:** Always try to make a zip file of size upto 100mb for better response time. If you want to process files whose zip size is more than 100MB, then try to upload them on FTP in batches of size less than 100MB.



Resume JSON Response on Webhook

```
{  
  "RChilliBulkUpload": {  
    "SubUserId": "subuser id",  
    "Base64Data": "base64 data for processed file",  
    "FileName": "processed file name",  
    "ResumeParserData": {  
      "ResumeFileName": "",  
      "ResumeLanguage": {  
        "Language": "",  
        "LanguageCode": ""  
      },  
      "ParsingDate": "",  
      "ResumeCountry": {  
        "Country": "",  
        "Evidence": "",  
        "CountryCode": {  
          "IsoAlpha2": "",  
          "IsoAlpha3": "",  
          "UNCode": ""  
        }  
      }  
    },  
    "Name": {  
      "FullName": "",  
      "TitleName": "",  
      "FirstName": "",  
      "MiddleName": "",  
      "LastName": "",  
      "FormattedName": "",  
      "ConfidenceScore": 10  
    },  
    "DateOfBirth": "",  
    "Gender": "",  
    "FatherName": "",  
    "MotherName": "",  
    "MaritalStatus": "",  
    "Nationality": "",  
    "LanguageKnown": [  
      {  
    ]  
  }  
}
```



We Analyze Not Just Parse

```
        "Language": "",
        "LanguageCode": ""
    }
},
"UniqueID": "",
"LicenseNo": "",
"PassportDetail": {
    "PassportNumber": "",
    "DateOfExpiry": "",
    "DateOfIssue": "",
    "PlaceOfIssue": ""
},
"PanNo": "",
"VisaStatus": "",
"Email": [
    {
        "EmailAddress": "",
        "ConfidenceScore": 10
    }
],
"PhoneNumber": [],
"WebSite": [
    {
        "Type": "",
        "Url": ""
    }
],
"Address": [
    {
        "Street": "",
        "City": "",
        "State": "",
        "StateIsoCode": "",
        "Country": "",
        "CountryCode": {
            "IsoAlpha2": "",
            "IsoAlpha3": "",
            "UNCode": ""
        },
        "ZipCode": "",
    }
],
```



We Analyze Not Just Parse

```
        "FormattedAddress":"","  
        "Type":"","  
        "ConfidenceScore": 10  
    }  
},  
"Category":"","  
"SubCategory":"","  
"CurrentSalary": {  
    "Amount":"","  
    "Symbol":"","  
    "Currency":"","  
    "Unit":"","  
    "Text":""  
},  
"ExpectedSalary": {  
    "Amount":"","  
    "Symbol":"","  
    "Currency":"","  
    "Unit":"","  
    "Text":""  
},  
"Qualification":"","  
"SegregatedQualification": [  
    {  
        "Institution": {  
            "Name":"","  
            "Type":"","  
            "ConfidenceScore": 10,  
            "Location": {  
                "City":"","  
                "State":"","  
                "StateIsoCode":"","  
                "Country":"","  
                "CountryCode": {  
                    "IsoAlpha2":"","  
                    "IsoAlpha3":"","  
                    "UNCode":""  
                }  
            }  
        }  
    }  
],  
},
```




We Analyze Not Just Parse

```
"Degree": {
  "DegreeName": "",
  "NormalizeDegree": "",
  "Specialization": [],
  "ConfidenceScore": 10
},
"FormattedDegreePeriod": "",
"StartDate": "",
"EndDate": "",
"Aggregate": {
  "Value": "",
  "MeasureType": ""
}
},
],
"Certification": "",
"SegregatedCertification": [
{
  "CertificationTitle": "",
  "Authority": "",
  "CertificationCode": "",
  "IsExpiry": "",
  "StartDate": "",
  "EndDate": "",
  "CertificationUrl": ""
}
],
"SkillBlock": "",
"SkillKeywords": "",
"SegregatedSkill": [
{
  "Type": "",
  "Skill": "",
  "Ontology": "",
  "Alias": "",
  "FormattedName": "",
  "Evidence": "",
  "LastUsed": "",
  "ExperienceInMonths": 10
}
]
```



We Analyze Not Just Parse

```
],  
  "Experience": "",  
  "SegregatedExperience": [  
    {  
      "Employer": {  
        "EmployerName": "",  
        "FormattedName": "",  
        "ConfidenceScore": 10  
      },  
      "JobProfile": {  
        "Title": "",  
        "FormattedName": "",  
        "Alias": "",  
        "RelatedSkills": [],  
        "ConfidenceScore": 10  
      },  
      "Location": {  
        "City": "",  
        "State": "",  
        "StateIsoCode": "",  
        "Country": "",  
        "CountryCode": {  
          "IsoAlpha2": "",  
          "IsoAlpha3": "",  
          "UNCode": ""  
        }  
      }  
    },  
    {  
      "JobPeriod": "",  
      "FormattedJobPeriod": "",  
      "StartDate": "",  
      "EndDate": "",  
      "IsCurrentEmployer": "",  
      "JobDescription": "",  
      "Projects": [  
        {  
          "UsedSkills": "",  
          "ProjectName": "",  
          "TeamSize": ""  
        }  
      ]  
    }  
  ]
```



We Analyze Not Just Parse

```
}
],
"CurrentEmployer": "",
"JobProfile": "",
"WorkedPeriod": {
  "TotalExperienceInMonths": "",
  "TotalExperienceInYear": "",
  "TotalExperienceRange": ""
},
"GapPeriod": "",
"AverageStay": "",
"LongestStay": "",
"Summary": "",
"ExecutiveSummary": "",
"ManagementSummary": "",
"Coverletter": "",
"Publication": "",
"SegregatedPublication": [
  {
    "PublicationTitle": "",
    "Publisher": "",
    "PublicationNumber": "",
    "PublicationUrl": "",
    "Authors": "",
    "Description": ""
  }
],
"CurrentLocation": [],
"PreferredLocation": [],
"Availability": "",
"Hobbies": "",
"Objectives": "",
"Achievements": "",
"SegregatedAchievement": [
  {
    "AwardTitle": "",
    "Issuer": "",
    "AssociatedWith": "",
    "IssuingDate": "",
    "Description": ""
  }
]
```



We Analyze Not Just Parse

```
    }  
  ],  
  "References": "",  
  "CustomFields": "",  
  "EmailInfo": {  
    "EmailTo": "",  
    "EmailBody": "",  
    "EmailReplyTo": "",  
    "EmailSignature": "",  
    "EmailFrom": "",  
    "EmailSubject": "",  
    "EmailCC": ""  
  },  
  "Recommendations": [  
    {  
      "PersonName": "",  
      "CompanyName": "",  
      "Relation": "",  
      "PositionTitle": "",  
      "Description": ""  
    }  
  ],  
  "DetailResume": "",  
  "HtmlResume": "",  
  "CandidateImage": {  
    "CandidateImageData": "",  
    "CandidateImageFormat": ""  
  },  
  "TemplateOutput": {  
    "TemplateOutputFileName": "",  
    "TemplateOutputData": ""  
  },  
  "ApiInfo": {  
    "Metered": "",  
    "CreditLeft": "",  
    "AccountExpiryDate": "",  
    "BuildVersion": ""  
  }  
}
```



Note: For any queries, please email at support@rchilli.com

Sample Code for Webhook Page:

We have sample code for capturing JSON posted on webhook

- RChilli Http Webhook Sample Code Java ([Download](#))
- RChilli Http Webhook Sample Code C# ([Download](#))
- RChilli Http Webhook Sample Code Php ([Download](#))

Java

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // TODO Auto-generated method stub  
    request.setCharacterEncoding("UTF-8");  
    BufferedReader in = new BufferedReader(new InputStreamReader(request.getInputStream()));  
    String line = null;  
    StringBuilder sb= new StringBuilder();  
    while((line = in.readLine()) != null) {  
        sb.append(line+"\r\n");  
    }  
    String Output=sb.toString();  
}
```

C#

```
protected void Page_Load(object sender, EventArgs e)  
{  
    Page.Response.ContentType = "text/xml";  
    // Read json posted via HTTP  
    StreamReader reader = new StreamReader(Page.Request.InputStream);  
    String xmlData = reader.ReadToEnd();  
}
```

PHP

```
<?php  
  
if( $_POST["xmldata"]||$_POST["?xmldata"] )  
{  
    $jsonOutput =urldecode(utf8_encode($_POST["xmldata"]));  
    //Read json  
    $jsonOutput = json_decode($jsonOutput, true);  
}
```

Note:

Output JSON can be captured from Input Stream in C# and JAVA.
While in PHP Code, we post data in xmldata variable.

These sample codes can be used to receive data for any version but need to change output reading logic for version resume parser version 8.0 and for JD parser.

Download Sample Code for Resume Parser Version 8.0 from [here](#)



Download Sample Code for JD Parser Version 3.0 from [here](#)
Parsing API Error Code and Error Messages

| Error Code | Error Message |
|-------------------|---|
| 1001 | <i>Userkey is required</i> |
| 1002 | <i>Subuser ID is required</i> |
| 1003 | <i>Invalid userkey</i> |
| 1004 | <i>File name with file extension is required</i> |
| 1005 | <i>Key expired</i> |
| 1006 | <i>Resume credits exhausted</i> |
| 1007 | <i>Wrong resume url</i> |
| 1008 | <i>Not Text Content</i> |
| 1009 | <i>Resume File Extension Not Supported</i> |
| 1010 | <i>Resume File does not exist on this url</i> |
| 1011 | <i>Version Required</i> |
| 1012 | <i>Invalid Version</i> |
| 1013 | <i>This Version does not support your Agreement</i> |
| 1014 | <i>Corrupted data/ No Content Found</i> |
| 1015 | <i>Unable to parse the content</i> |
| 1016 | <i>No Resume Content Found</i> |
| 1017 | <i>Wrong Version</i> |
| 1018 | <i>File size is too large for processing</i> |
| 1019 | <i>Unable to detect language</i> |
| 1020 | <i>Unable to open PDF Parser</i> |
| 1021 | <i>File conversion error</i> |
| 1041 | <i>Abbyy Error: Not enough credits to process document. Please add more pages to your application's account</i> |
| 1042 | <i>Abbyy Error: Task Failed</i> |