

Getting Started
With



RChilli Web API
Installation and Integration

Contents

1 Introduction to Web API.....	
1.1 Brief Introduction	
1.2 Feature List	
2 Installation steps.....	
2.1 To Install.....	
3 RChilli IIS screen overview.....	
4 Activities on IIS screen.....	
5 How to use it.....	
6 API Integration	

Introduction to Web API

RChilli's Resume Parser web API is an enterprise solution designed keeping in mind the requirements of Job boards, corporate website or any other web application who are looking to develop or integrate resume parsing capability in it. RChilli's web API is an easy to integrate Web API which fulfills this requirement and ensures the highly scalable resume parsing capabilities with the option of self hosting or hosting on cloud.

With resume parsing capability, companies can increase the user experience for candidates by offering the option to directly upload resumes on their corporate website.

Job boards can process resumes much faster and send the resume details to their database quickly and leaving the managing the parsing accuracy and speed capabilities improvements on us and can focus on their activities of bringing more clients and candidates on job board.

Features We Provide

- Easy integration with any job portals, corporate websites and any web application.
- Fasten the resume uploading process in job portals.
- Support major resume formats (doc, docx, rtf, HTML,pdf,odt etc).
- Available in major sample codes (DotNet, C#, PHP, Java etc).
- Sample code available in Dot Net, C#, PHP and Java

Rchilli Web API – Installation Configuration of IIS(Internet Information Services) server

RChilli's Web API requires windows based server OS (**Windows 2003, 2008 R2**) and IIS to be installed on the server to function, if these IIS is not installed on your server, please follow the given below steps to install and configure IIS on **windows 2008 R2**.

IIS Installation

Step 1: Click on **Start** button, right click on **Computer** and click on **Manage**.

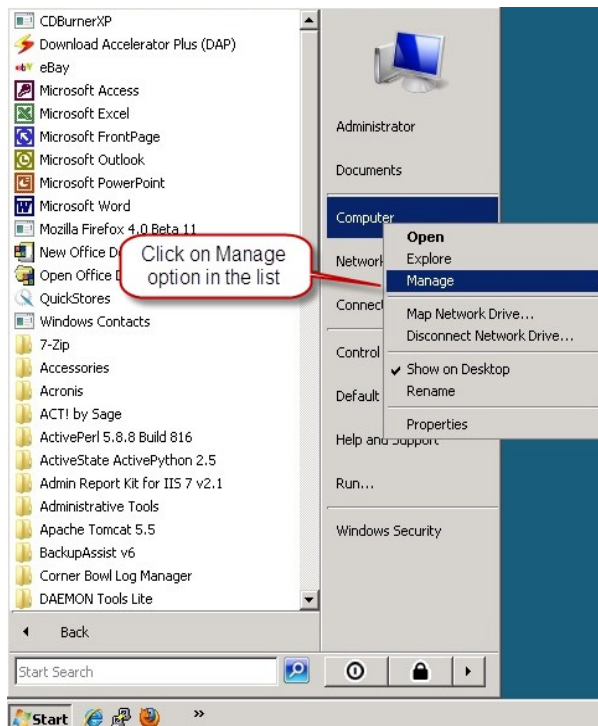


Image 1.

Step 2: It will open **Server Management** window.



Image 2.

Under server manager, select **Roles**, it will show the various information like Roles summary , DNS server etc on right hand side of the server management window.

Now click on **Add Roles** link in the role summary section.

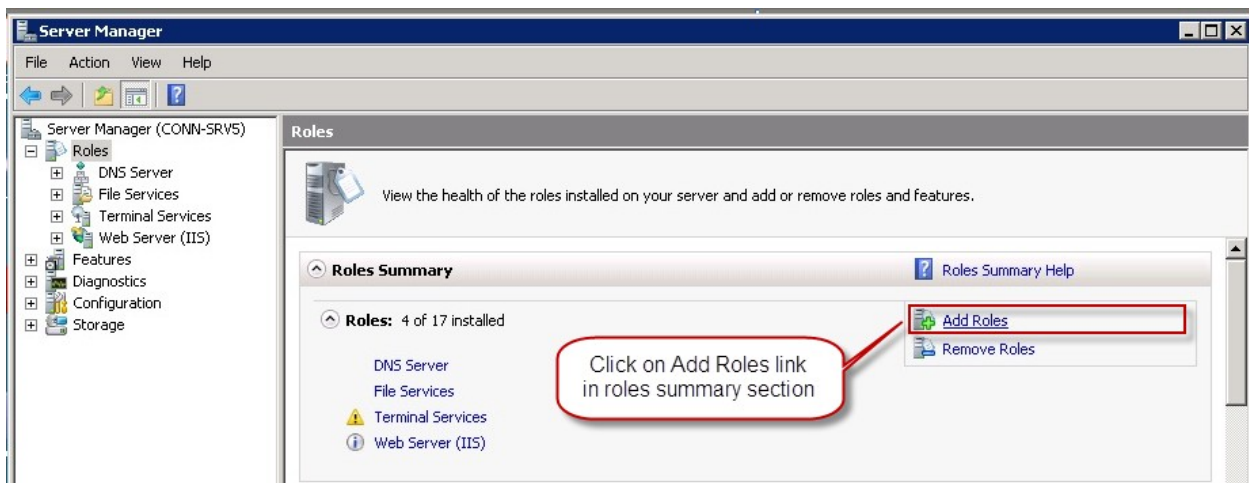


Image 3.

Step 3: Clicking on Add roles will launch the “Add Roles Wizard”

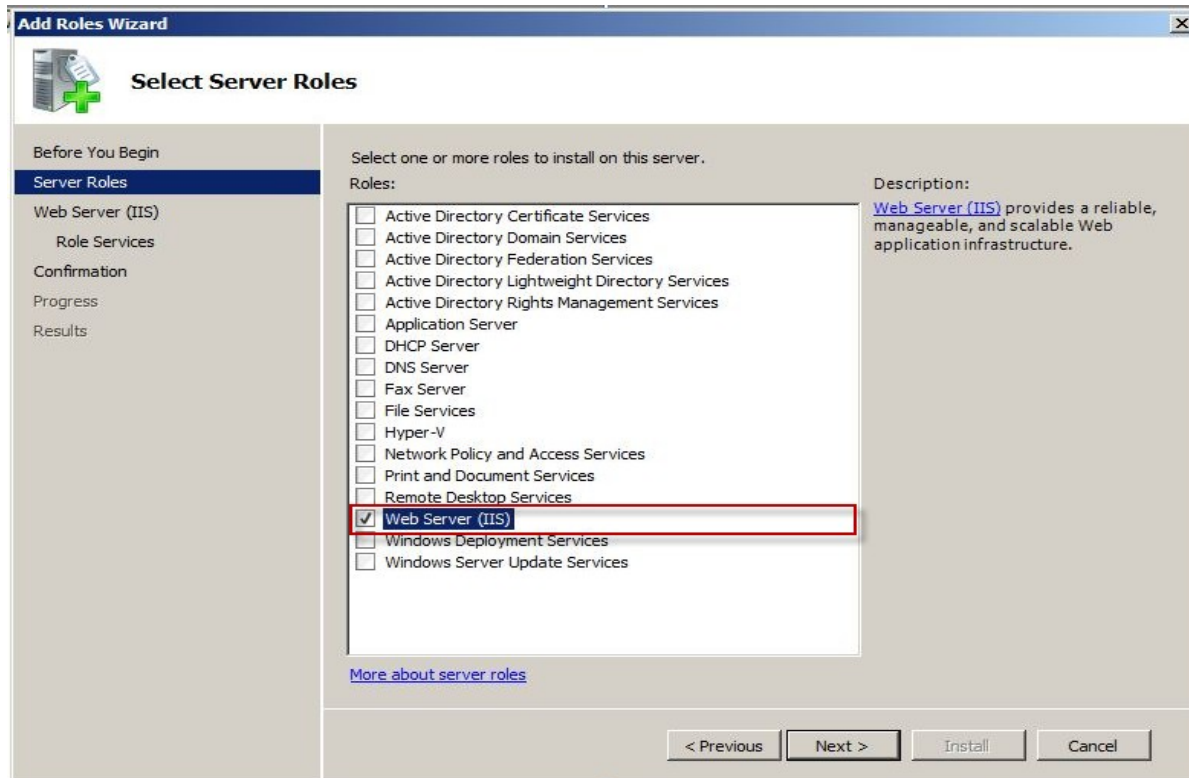


Image 4.

In Add roles Wizard, select the **web server (IIS)** option in the roles list and click on **Next button**.

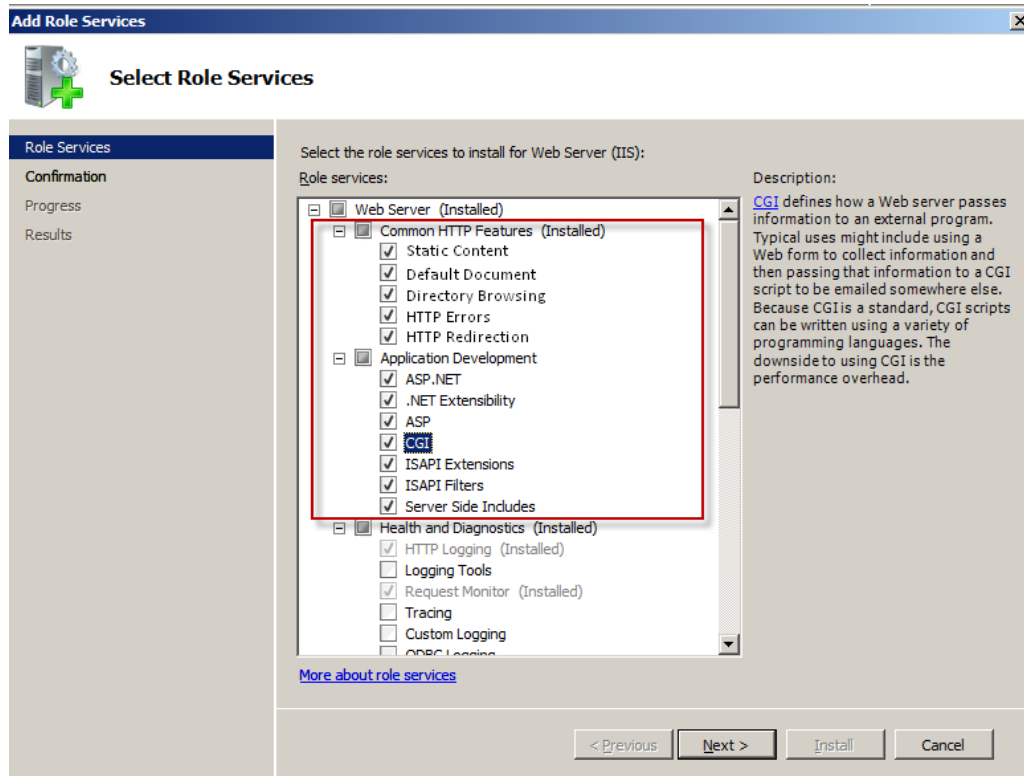


Image 5.

Clicking on Next button will launch “**Add Role services**”, now select all the above selected services as shown in the above image and click on Next button this will install the IIS server on your machine.

IIS Configuration

Once IIS is installed on your machine, the IIS Manager will be launched automatically (or you can launch the IIS Manager from the start Menu option). Once the IIS Manager is launched, right click on Sites option in left pane and click on **Add web site** in right click option list.

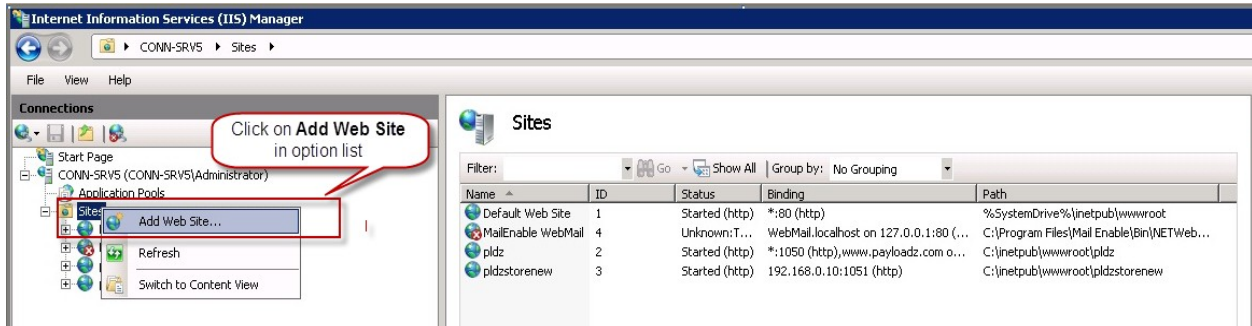


Image 6.

Clicking on Add Web Site option will launch “Add Web Site” option Window where you need to fill information related to **site name** and **path** where Rchilli’s API will be installed.

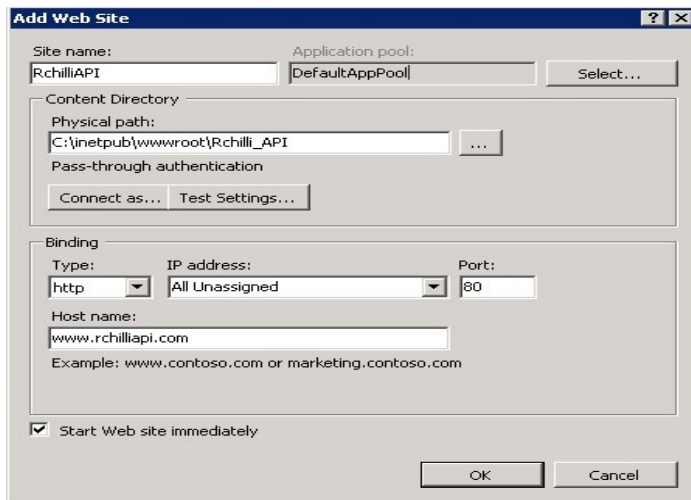


Image 7.

In Add Web Site option Window, you can see certain fields that needs to fill to complete the configuration.

Site Name – Here you need to fill your **site name** so that it can be identifiable with others.

Site name:
RchilliAPI

Application Pool: Click on **Select** button and select the **DefaultAppPool** option under the dropdown options.

Application pool:
DefaultAppPool Select...

Physical Path: Select button to select the location in your server where your RChilli's web API folder is residing (should be inside wwwroot folder).



Under sub heading of Binding, you can see some options Type, IP Address, Port & Host Address.

Here you need to fill some information like for

Type – you can select here **http**, **https** under type dropdown, select the option as per your server configuration.

IP Address: Keep the value “**All unassigned**” in this dropdown.

Port: Based on your selection of Type selected (http, https), you have to change the value to **80** for **http** & **143** for **https**.

Once completed the information in the fields, click on “**OK**” button and API will be added under the sites pane.

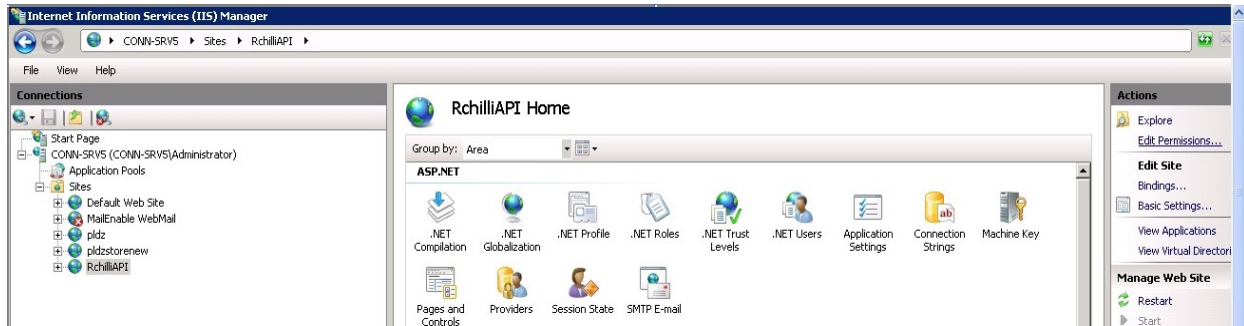


Image 8.

Now click on **Edit permissions** link the right hand pane under **Action**

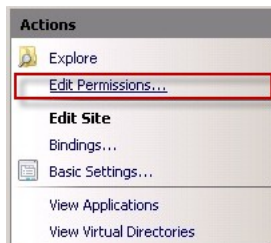


Image 9.

Clicking on Edit permissions link will launch **API's properties window**, now please click on **Edit** button in the properties window.

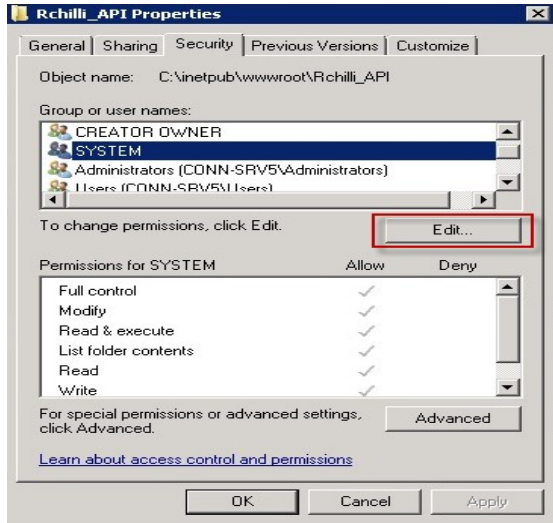


Image 10.

Now click on **Add** button in the permissions for the API folder and click on **Add** button

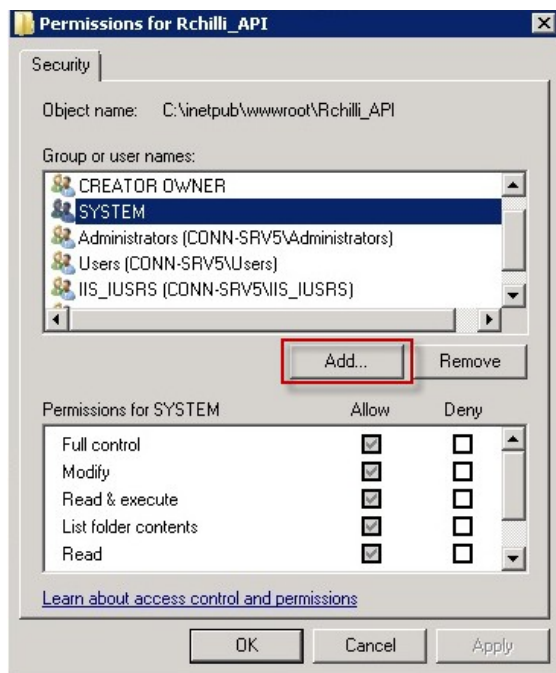


Image 11.

Clicking on **Add** button will launch select User or Groups windows, now click on **Advanced** button on the window.

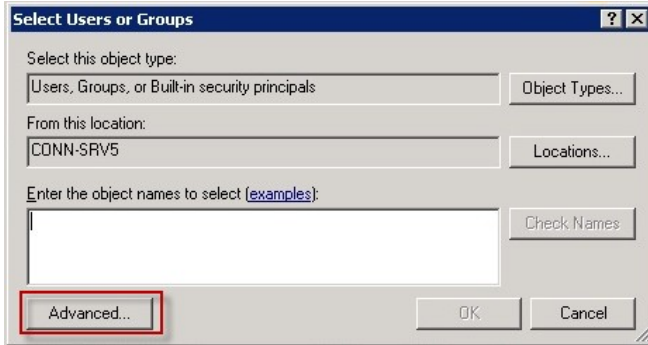


Image 12.

Clicking on Advanced button will launch another “**Select User or Groups**” button where you need to click on “**Find Now**” button.

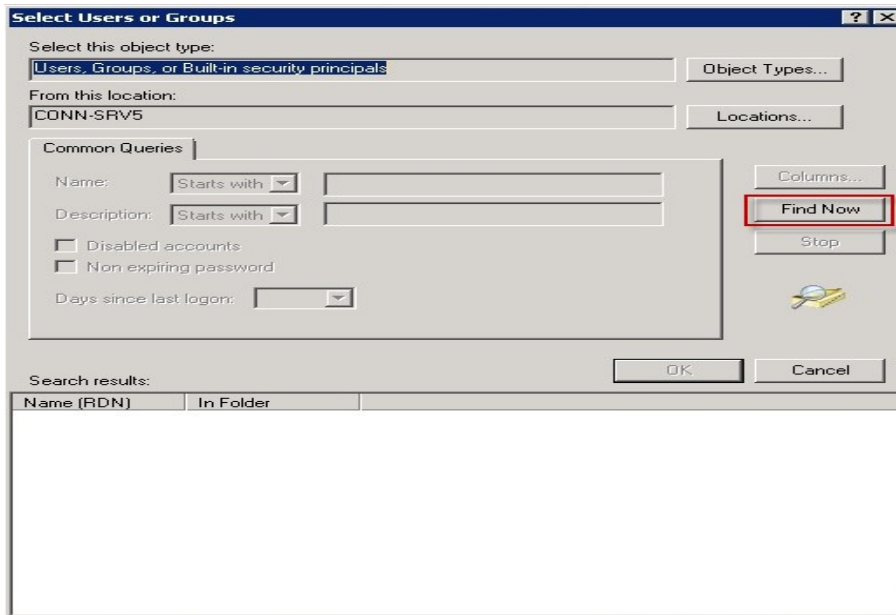
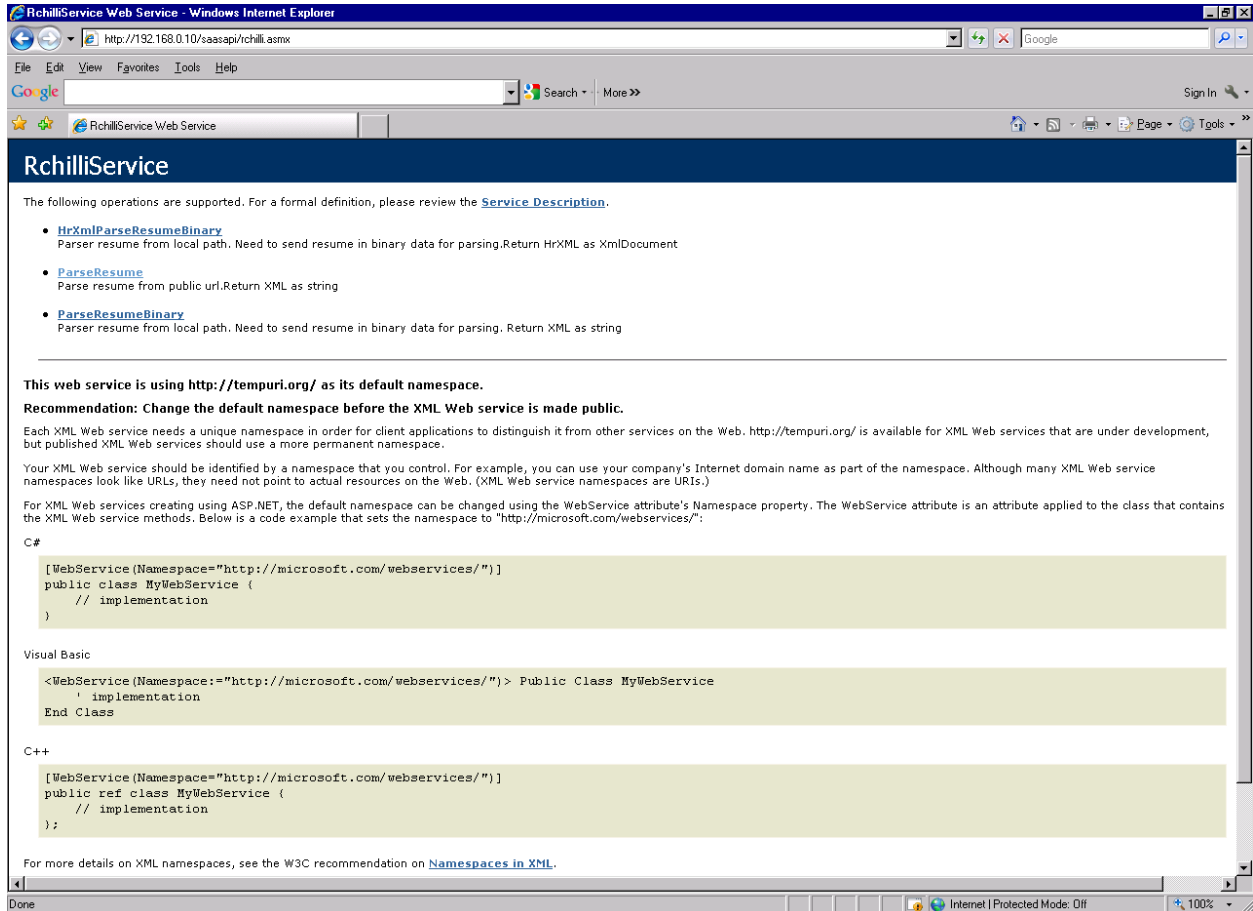


Image 13.

Here you have to select specific users or Group for permissions (like Administrator, Administrators & Everyone) and these users or groups will be added to the Permissions window.

Image 16.

Now, your API configuration is completed and you are ready to use RChilli's web API for service.



RchillService Web Service - Windows Internet Explorer

http://192.168.0.10/saasapi/rchill.asmx

File Edit View Favorites Tools Help

Google Search More >> Sign In

RchillService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [HrXmlParseResumeBinary](#)
Parser resume from local path. Need to send resume in binary data for parsing.Return HrXML as XmlDocument
- [ParseResume](#)
Parse resume from public url.Return XML as string
- [ParseResumeBinary](#)
Parser resume from local path. Need to send resume in binary data for parsing. Return XML as string

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

For more details on XML namespaces, see the W3C recommendation on [Namespaces in XML](#).

Done Internet | Protected Mode: Off 100%

Image 17.

API Integration

There are two method you can use for parsing resume using RChilli web service hosted at your server.

- Parsing Resume by Resume URL
- Parsing Resume by sending Binary Data

Parsing Resume by Resume URL

Input

Sr. no	SOAP XML NODE	Constrain Type	Value Info	Values
1	Resume URL	string	Public URL along with file name	Required
2	Version	string	"4.0"	Required

NOTE: Please read the document thoroughly.

Description: The Transaction Detail API is a RESUME PARSER WEB API which used to get the detail of the resume in the xml format .For this the user pass the

"URL along with filename like exp.(<http://www.xyz.com/resume.doc>) and Version 4.0 and get xml string output"

Note:- In XML of version 4.0 we return data in CData

Soap Request to send:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ParseResume xmlns="http://tempuri.org/">
      <url>Public URL where resume file (DOC/DOCX/HTM/HTML/TXT/RTF/ODT
      Format is uploaded. </url>
      <version> 4.0 </version>
    </ParseResume>
  </soap:Body>
</soap:Envelope>
```

Request URL Where Soap String to be sent:

<http://www.yourserverurl/RChilli.asmx?op=ParseResume>

Output

The Soap response Get Back

```
<ResumeParserData>
<ResumeFileName><![CDATA[Candidate ]]></ResumeFileName>
<ParsingDate><![CDATA[Candidate ParsingDate]]></ParsingDate>
<TitleName><![CDATA[Candidate TitleName]]></TitleName>
<FirstName><![CDATA[Candidate FirstName]]></FirstName>
<Middlename><![CDATA[Candidate Middlename]]></Middlename>
<LastName><![CDATA[Candidate LastName]]></LastName>
<Email><![CDATA[Candidateemail@mail.com]]></Email>
<Phone><![CDATA[Candidate Phone]]></Phone>
<Mobile><![CDATA[Candidate Mobile]]></Mobile>
<FaxNo><![CDATA[Candidate FaxNo]]></FaxNo>
<LicenseNo><![CDATA[Candidate LicenseNo]]></LicenseNo>
<PassportNo><![CDATA[Candidate PassportNo]]></PassportNo>
<!--Visa Status will be provided in future release-->
<VisaStatus />
<Address><![CDATA[Candidate Address]]></Address>
<City><![CDATA[Candidate City]]></City>
<State><![CDATA[Candidate State]]></State>
<ZipCode><![CDATA[Candidate ZipCode]]></ZipCode>
<!--PermanentAddress will be provided in future release-->
<PermanentAddress />
<!--PermanentCity will be provided in future release-->
<PermanentCity />
<!--PermanentState will be provided in future release-->
<PermanentState />
<!--PermanentZipCode will be provided in future release-->
<PermanentZipCode />
<!--CorrespondenceAddress will be provided in future release-->
<CorrespondenceAddress />
<!--CorrespondenceCity will be provided in future release-->
```

```
<CorrespondenceCity />
<!--CorrespondenceState will be provided in future release-->
<CorrespondenceState />
<!--CorrespondenceZipCode will be provided in future release-->
<CorrespondenceZipCode />
<Category><![CDATA[Candidate Category]]></Category>
<SubCategory><![CDATA[Candidate SubCategory]]></SubCategory>
<DateOfBirth><![CDATA[Candidate DateOfBirth]]></DateOfBirth>
<Gender><![CDATA[Candidate Gender]]></Gender>
<FatherName><![CDATA[Candidate FatherName]]></FatherName>
<MotherName><![CDATA[Candidate MotherName]]></MotherName>
<MaritalStatus><![CDATA[Candidate MaritalStatus]]></MaritalStatus>
<Nationality><![CDATA[Candidate Nationality]]></Nationality>
<CurrentSalary><![CDATA[Candidate CurrentSalary]]></CurrentSalary>
<ExpectedSalary><![CDATA[Candidate ExpectedSalary]]></ExpectedSalary>
<Qualification><![CDATA[Candidate Qualification]]></Qualification>
<SegrigatedQualification>
<EducationSplit>
<University><![CDATA[Candidate University]]></University>
<Degree><![CDATA[Candidate Degree]]></Degree>
<Year><![CDATA[Candidate Year]]></Year>
</EducationSplit>
</SegrigatedQualification>
<Skills><![CDATA[Candidate Skills]]></Skills>
<SkillsKeywords>
<BehaviorSkills><![CDATA[Candidate BehaviorSkills]]></BehaviorSkills>
<SoftSkills><![CDATA[Candidate SoftSkills]]></SoftSkills>
<OperationalSkills>
<SkillSet>
<Skill><![CDATA[Candidate Skill]]></Skill>
<ExperienceInMonths > <![CDATA[Candidate ExperienceInMonths]]></ExperienceInMonths>
</SkillSet>
</OperationalSkills>
</SkillsKeywords>
<LanguageKnown><![CDATA[Candidate LanguageKnown]]></LanguageKnown>
```

```
<Experience><![CDATA[Candidate Experience]]></Experience>
<SegrigatedExperience>
<WorkHistory>
<Employer><![CDATA[Candidate Employer]]></Employer>
<JobProfile><![CDATA[Candidate JobProfile]]></JobProfile>
<JobLocation><![CDATA[Candidate JobLocation]]></JobLocation>
<JobPeriod><![CDATA[Candidate JobPeriod]]></JobPeriod>
<StartDate><![CDATA[Candidate StartDate]]></StartDate>
<EndDate><![CDATA[Candidate EndDate]]></EndDate>
<JobDescription><![CDATA[Candidate JobDescription]]></JobDescription>
<Projects>
<ProjectName><![CDATA[Candidate ProjectName]]></ProjectName>
<UsedSkills><![CDATA[Candidate UsedSkills]]></UsedSkills>
<TeamSize><![CDATA[Candidate TeamSize]]></TeamSize>
</Projects>
</WorkHistory>
</SegrigatedExperience>
<CurrentEmployer><![CDATA[Candidate CurrentEmployer]]></CurrentEmployer>
<JobProfile><![CDATA[Candidate JobProfile]]></JobProfile>
<WorkedPeriod><![CDATA[Candidate WorkedPeriod]]></WorkedPeriod>
<GapPeriod><![CDATA[Candidate GapPeriod]]></GapPeriod>
<!--Number of Job Changed will be provided in future release-->
<NumberOfJobChanged />
<!--Average Stay will be provided in future release-->
<AverageStay />
<!--Availability will be provided in future release-->
<Availability />
<!--Competency Related fields will be provided in future release-->
<Competency>
<CompetencyName />
<Evidence />
<LastUsed />
<Description />
</Competency>
<Hobbies><![CDATA[Candidate Hobbies]]></Hobbies>
```

```
<Objectives><![CDATA[Candidate Objectives]]></Objectives>
<Achievements><![CDATA[Candidate Achievements]]></Achievements>
<References><![CDATA[Candidate References]]></References>
<!--Preferred Location will be provided in future release-->
<PreferredLocation />
<!--Certification will be provided in future release-->
<Certification />
<!--UniqueID will be provided in future release-->
<UniqueID />
<!--CustomFields will be provided in future release-->
<CustomFields />
<!--Email Info Related Fields will be provided in future release-->
<EmailInfo>
<EmailFrom />
<EmailTo />
<EmailSubject />
<EmailBody />
<EmailCC />
<EmailReplyTo />
<EmailSignature />
</EmailInfo>
<!--WebSites Info Related Fields will be provided in future release-->
<WebSites>
<Website />
</WebSites>
<!--Recommendation Field Provide values for Linkedin Resumes-->
<Recommendations>
<Recomendation>
<PersonName />
<PositionTitle />
<CompanyName />
<Relation />
<Description />
</Recomendation>
</Recommendations>
```

```
<DetailResume><![CDATA[Candidate DetailResume]]></DetailResume>
<htmlresume><![CDATA[Candidate htmlresume]]></htmlresume>
<maskresume><![CDATA[Candidate maskresume]]></maskresume>
</ResumeParserData>
```

To use Resume Parser API, Some essential information to be needed as follows :

- Make sure all the nodes must be filled in. The above table structure defines the required parameters whose values must not be empty and rest of the parameters may have their values empty accordingly. Use the above mentioned SOAP string in your application for processing.

Following messages will appear due to error or wrong information :

File Not Load : This message comes from API when user has not entered the correct URL, or if the file is not in the format of DOC, DOCX, TXT, HTML, RTF.

Note the things below :

- You must enter the correct URL along with the resume file name which you want to parse.
- Always enter the valid version to get the result.

Error Code

1004	Wrong URL
1005	Not Text Content, Please save this file in proper format and send.
1006	Resume File Extension not supported
1007	Resume File does not Exist On This URL
1008	Version is Required (2.1,2.6,2.8,2.9,3.1,3.1.1,3.1.2,3.1.3,3.2,4.0)
1010	Corrupted data
1012	Invalid Version

Parsing Resume by sending Binary Data

Input

Sr. No	SOAP XML NODE	Constraint Type	Value Info	Values
1	File data	base64Binary	Resume Data	Required
2	File type	string	Extension of the resume file	Required
3	Version	string	"4.0"	Required

NOTE : Please read the document thoroughly.

Description: The Transaction Detail API is a RESUME PARSER WEB API which used to get the detail of the resume in the xml format .For this the user pass the

"resume as binary data and get the xmloutput"

Soap Request to send:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd =
"http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ParseResumeBinary xmlns="http://tempuri.org/">
      <filedata>base64Binary</filedata>
      <filetype>string</filetype>
      <version>string</version>
    </ParseResumeBinary>
  </soap:Body>
</soap:Envelope>
```

Request URL Where Soap String to be sent:

<http://www.yourserverurl/RChilli.asmx>

Output

The Soap response Get Back

```
<ResumeParserData>
<ResumeFileName><![CDATA[Candidate ]]></ResumeFileName>
<ParsingDate><![CDATA[Candidate ParsingDate]]></ParsingDate>
<TitleName><![CDATA[Candidate TitleName]]></TitleName>
<FirstName><![CDATA[Candidate FirstName]]></FirstName>
<Middlename><![CDATA[Candidate Middlename]]></Middlename>
<LastName><![CDATA[Candidate LastName]]></LastName>
<Email><![CDATA[Candidateemail@mail.com]]></Email>
<Phone><![CDATA[Candidate Phone]]></Phone>
<Mobile><![CDATA[Candidate Mobile]]></Mobile>
<FaxNo><![CDATA[Candidate FaxNo]]></FaxNo>
<LicenseNo><![CDATA[Candidate LicenseNo]]></LicenseNo>
<PassportNo><![CDATA[Candidate PassportNo]]></PassportNo>
<!--Visa Status will be provided in future release-->
<VisaStatus />
<Address><![CDATA[Candidate Address]]></Address>
<City><![CDATA[Candidate City]]></City>
<State><![CDATA[Candidate State]]></State>
<ZipCode><![CDATA[Candidate ZipCode]]></ZipCode>
<!--PermanentAddress will be provided in future release-->
<PermanentAddress />
<!--PermanentCity will be provided in future release-->
<PermanentCity />
<!--PermanentState will be provided in future release-->
<PermanentState />
<!--PermanentZipCode will be provided in future release-->
<PermanentZipCode />
<!--CorrespondenceAddress will be provided in future release-->
<CorrespondenceAddress />
<!--CorrespondenceCity will be provided in future release-->
```

```
<CorrespondenceCity />
<!--CorrespondenceState will be provided in future release-->
<CorrespondenceState />
<!--CorrespondenceZipCode will be provided in future release-->
<CorrespondenceZipCode />
<Category><![CDATA[Candidate Category]]></Category>
<SubCategory><![CDATA[Candidate SubCategory]]></SubCategory>
<DateOfBirth><![CDATA[Candidate DateOfBirth]]></DateOfBirth>
<Gender><![CDATA[Candidate Gender]]></Gender>
<FatherName><![CDATA[Candidate FatherName]]></FatherName>
<MotherName><![CDATA[Candidate MotherName]]></MotherName>
<MaritalStatus><![CDATA[Candidate MaritalStatus]]></MaritalStatus>
<Nationality><![CDATA[Candidate Nationality]]></Nationality>
<CurrentSalary><![CDATA[Candidate CurrentSalary]]></CurrentSalary>
<ExpectedSalary><![CDATA[Candidate ExpectedSalary]]></ExpectedSalary>
<Qualification><![CDATA[Candidate Qualification]]></Qualification>
<SegrigatedQualification>
<EducationSplit>
<University><![CDATA[Candidate University]]></University>
<Degree><![CDATA[Candidate Degree]]></Degree>
<Year><![CDATA[Candidate Year]]></Year>
</EducationSplit>
</SegrigatedQualification>
<Skills><![CDATA[Candidate Skills]]></Skills>
<SkillsKeywords>
<BehaviorSkills><![CDATA[Candidate BehaviorSkills]]></BehaviorSkills>
<SoftSkills><![CDATA[Candidate SoftSkills]]></SoftSkills>
<OperationalSkills>
<SkillSet>
<Skill><![CDATA[Candidate Skill]]></Skill>
<ExperienceInMonths > <![CDATA[Candidate ExperienceInMonths]]></ExperienceInMonths>
</SkillSet>
</OperationalSkills>
</SkillsKeywords>
<LanguageKnown><![CDATA[Candidate LanguageKnown]]></LanguageKnown>
```

```
<Experience><![CDATA[Candidate Experience]]></Experience>
<SegrigatedExperience>
<WorkHistory>
<Employer><![CDATA[Candidate Employer]]></Employer>
<JobProfile><![CDATA[Candidate JobProfile]]></JobProfile>
<JobLocation><![CDATA[Candidate JobLocation]]></JobLocation>
<JobPeriod><![CDATA[Candidate JobPeriod]]></JobPeriod>
<StartDate><![CDATA[Candidate StartDate]]></StartDate>
<EndDate><![CDATA[Candidate EndDate]]></EndDate>
<JobDescription><![CDATA[Candidate JobDescription]]></JobDescription>
<Projects>
<ProjectName><![CDATA[Candidate ProjectName]]></ProjectName>
<UsedSkills><![CDATA[Candidate UsedSkills]]></UsedSkills>
<TeamSize><![CDATA[Candidate TeamSize]]></TeamSize>
</Projects>
</WorkHistory>
</SegrigatedExperience>
<CurrentEmployer><![CDATA[Candidate CurrentEmployer]]></CurrentEmployer>
<JobProfile><![CDATA[Candidate JobProfile]]></JobProfile>
<WorkedPeriod><![CDATA[Candidate WorkedPeriod]]></WorkedPeriod>
<GapPeriod><![CDATA[Candidate GapPeriod]]></GapPeriod>
<!--Number of Job Changed will be provided in future release-->
<NumberOfJobChanged />
<!--Average Stay will be provided in future release-->
<AverageStay />
<!--Availability will be provided in future release-->
<Availability />
<!--Competency Related fields will be provided in future release-->
<Competency>
<CompetencyName />
<Evidence />
<LastUsed />
<Description />
</Competency>
<Hobbies><![CDATA[Candidate Hobbies]]></Hobbies>
```

```
<Objectives><![CDATA[Candidate Objectives]]></Objectives>
<Achievements><![CDATA[Candidate Achievements]]></Achievements>
<References><![CDATA[Candidate References]]></References>
<!--Preferred Location will be provided in future release-->
<PreferredLocation />
<!--Certification will be provided in future release-->
<Certification />
<!--UniqueID will be provided in future release-->
<UniqueID />
<!--CustomFields will be provided in future release-->
<CustomFields />
<!--Email Info Related Fields will be provided in future release-->
<EmailInfo>
<EmailFrom />
<EmailTo />
<EmailSubject />
<EmailBody />
<EmailCC />
<EmailReplyTo />
<EmailSignature />
</EmailInfo>
<!--WebSites Info Related Fields will be provided in future release-->
<WebSites>
<Website />
</WebSites>
<!--Recommendation Field Provide values for Linkedin Resumes-->
<Recommendations>
<Recomendation>
<PersonName />
<PositionTitle />
<CompanyName />
<Relation />
<Description />
</Recomendation>
</Recommendations>
```

```
<DetailResume><![CDATA[Candidate DetailResume]]></DetailResume>
<htmlresume><![CDATA[Candidate htmlresume]]></htmlresume>
<maskresume><![CDATA[Candidate maskresume]]></maskresume>
</ResumeParserData>
```

To use Resume Parser API, Following are the essential info to be needed-

- Make sure all the nodes must be filled in. The above table structure defines the required parameters whose values must not be empty and rest of the parameters may have their values empty accordingly.
- Use the above mentioned SOAP string in your application for processing.

Following messages will appear due to error or wrong information:

File Not Load: This message comes from API when user has not entered the correct URL, or if the file is not in the format of DOC, DOCX, TXT, HTML, RTF.

Note the things below:

You must enter the correct URL along with the resume file name which you want to parse.

Always enter the valid version to get the result.

Error Code

1004	Wrong URL
1005	Not Text Content, Please save this file in proper format and send.
1006	Resume File Extension not supported
1007	Resume File does not Exist On This URL
1008	Version is Required (2.1,2.6,2.8,2.9,3.1,3.1.1,3.1.2.3.1.3,3.2,4.0)
1010	Corrupted data
1012	Invalid Version

Note :-

You can do certain setting for api in Config [File:-](#)

1. **SecretKey** :- Add your domain licence key into value.

```
<add key="SecretKey" value=""/><!--Enter your DomainLicense Key -->
```

2. **SpecialWord** :- You can add the special words which cannot be name in the resume

```
<add key="specialWords" value=",ASSISTANT,EXECUTIVE,"/>
```

3. **Commonwords** :- You can add the commonwords which cannot be name in the resume

```
<add key="commonwords" value="is,am,are,for,to,be,myself,he,she,me"/>
```

4. **Titles** :- You can add the titles keyword of the candidate which can appear in resume

```
<add key="Titles" value="DR,DR.,MR,MR.,MS,MS.,MRS,MRS.,ER,ER.,JR,JR."/>
```

5. **CarriageReturn** :- Set it 'True' to get line breaks as in the resume

```
<add key="CarriageReturn" value="True"/>
```

6. **Maskcontent** :- Set it 'True' to get maskresume tag in xml which contain resume content without contact info.

```
<add key="MaskContent" value="False"/>
```

7. **HtmlResume** :- To get resume content In Html format set it "True")

```
<add key="HtmlResume" value="False"/>
```

8. **Multiemail** :- To get multiple emails set it "True"

```
<add key="Multiemail" value="True"/>
```

9. **MultiPhone** :- To get multiple phone set it "True"

```
<add key="Multiphone" value="True"/>
```

10. **Writelog**:- This will write all error log files in log folder where api is hosted

```
<add key="WriteLog" value="True"/>
```

11. **LargeFieldInBaseBinary** :- Set this true if you want to get large fields in base binary.

```
<add key="LargeFieldInBaseBinary" value="False"/>
```

12. **Save Binary File** :- Set this as true if you are using parseresumebinary method and want to save the resume file too. Files will be saved in Resume folder

```
<add key="savebinaryfile" value="False"/>
```

Thank You.

If you need any help in integration contact us:-

Email:- support@rchilli.com

or you can live chat on Rchilli.com