* Simple APIs
  + Chart, bubble chart

    Description automatically generatedAn API allows 2 programs to talk to each other.
  + APPLICATION PROGRAMING INTERFACE
  + Rest APIs
    - Graphical user interface, text

      Description automatically generated**Re**presentational **S**tate **T**ransfer **APIs**

OTT8zmG8ogXyrbq2dVOhrrnaEUdYgbMQxwraXFVbxoxB

<https://api.us-south.speech-to-text.watson.cloud.ibm.com/instances/ce69c85b-390b-4e09-8656-444e73945c4b>

{

"apikey": "OTT8zmG8ogXyrbq2dVOhrrnaEUdYgbMQxwraXFVbxoxB",

"iam\_apikey\_description": "Auto-generated for key crn:v1:bluemix:public:speech-to-text:us-south:a/f0503188ab024d6c8a2d51b47e933513:ce69c85b-390b-4e09-8656-444e73945c4b:resource-key:230f6ef9-c39a-4786-8dea-03d840ca36b9",

"iam\_apikey\_name": "Auto-generated service credentials",

"iam\_role\_crn": "crn:v1:bluemix:public:iam::::serviceRole:Manager",

"iam\_serviceid\_crn": "crn:v1:bluemix:public:iam-identity::a/f0503188ab024d6c8a2d51b47e933513::serviceid:ServiceId-8222c844-0180-492c-912a-4803abbc46ec",

"url": "https://api.us-south.speech-to-text.watson.cloud.ibm.com/instances/ce69c85b-390b-4e09-8656-444e73945c4b"

}

* URL
  + - Uniform resource locator
    - Scheme : Protocol
    - Base URL : www……
    - Route: Location o the web server
* Table

  Description automatically generatedRequest Message

Table

Description automatically generated

* Table

  Description automatically generatedStatus Code
* Table

  Description automatically generatedHTTP Methods
* Graphical user interface, text, application, email

  Description automatically generatedRequests
* **Extracting Data from webpage.**
  + Web scraping
    - Requests and Beautiful Soup
    - From bs4 import BeautifulSoup
    - Store html as a string
    - Soup = BeautifulSoup(html, ‘thml5lib)

Text

Description automatically generated

* Reading CSV Files
  + **Import pandas as pd**
  + **file = “fileExample.csv”**
  + **df = pd.read+csv(file)**
  + **df.columns = [‘name’, ‘phone number’ etc]** #This function adds headers to data if the data doesn’t have headers.
* Reading JSON files
  + **Import json**
  + **With open(‘filesample.json’, ‘r’) as openfile:**
    - **Json\_object = json.load(openfile)**
* Reading XML Files (xml needs to be parsed inorder for it to be used)
  + **Import pandas as pd**
  + **Import xml.etree.ElementTree as etree**
  + **Tree = etree.parse(“fileExample.xml”)**
  + **Root = tree.getroot()**
  + **Columns = [“Name”, “Phone Number”, “Birthday”]**
  + **Df = pd.DataFrame(columns = columns)**
  + **For node in root:**
    - **Name = node.find(“name”).text**
    - **Phonenumber = node.find(“phonenumber”).text**
    - **Birthday = node.find(“birthday”).text**
  + **Df = df.append(pd.Series([name, phonenumber, birthday], index = columns) ignore\_index = True)**