

VBA SOURCE CODE BOOK

NATURAL LANGUAGE INPUT



Add Fred Fredders 12234 Main St. Los Angeles
CA 90213 Fred@AnyCompany.com 310 222-
1212

The following contact has been added
Contact Name: Fred Fredders
Email: Fred@AnyCompany.com
Phone #: 310 222-1212
Address: 12234 Main St.
City: Los Angeles
State: CA
Zip: 90213



Excel For Freelancers

The Future Of Excel Is Here! Natural Language Input



[DOWNLOAD
APPLICATION](#)



[VIEW
TRAINING](#)

by: Randy Austin

ABOUT THE AUTHOR

A two-time Microsoft MVP & lifetime Excel enthusiast, Randy Austin founded Excel For Freelancers in 2017. Excel For Freelancers quickly became the most prominent resource Excel for developers to learn how to turn their passion for Excel into profits by building & selling their own excel-based applications for passive & recurring income.

With over 442,000 YouTube subscribers, 33,000,000 video views, 250+ comprehensive training videos, and a thriving 60,000 member Facebook community, Excel For Freelancers has positioned itself as the #1 Excel developers resource in the world.

Get free content, training, and downloads just by clicking any of the free resources below:



[WEBSITE](#)



[YOUTUBE](#)



[FACEBOOK](#)



[TWITTER](#)



[DISCORD](#)



[INSTAGRAM](#)



[TELEGRAM](#)



[RUMBLE](#)



Microsoft®
Most Valuable
Professional



OUR COURSES & PRODUCTS



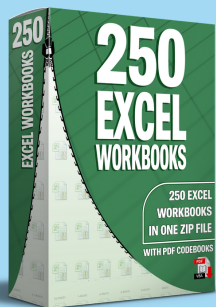
This comprehensive program will take you through a 12-phase process that will turn your enthusiasm for Excel into passive income.

[Click here to learn more](#)



16 hour masterclass that will teach you the tips, tricks and techniques on how to create a dynamic single-click dashboard, and a ton more

[Click here to learn more](#)



Incredible Package of 250 of my BEST Applications now with PDF VBA Codebooks packed into a SINGLE ZIP File which also includes the "250 Workbook Library".

[Click here to learn more](#)



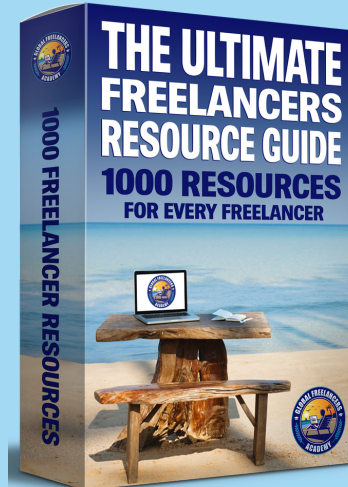
With 1000 live links, continuously updating content, sort-able and filterable items, you will always have exactly what you need, when you need it.

[Click here to learn more](#)



This incredible 13-hour freelancer masterclass will teach anyone how to be a successful freelancer with my proven 9-Phase 'Financial Freedom Roadmap' and includes 30+ downloads and exercises along with a ton of resources for anyone to reach their goals through freelancing.

[Click here to learn more](#)



I've put together all the Freelancing essentials, from freelance tools to freelance templates, in this Ultimate Freelancer's Resource Guide for freelancers at any stage in their career. With 1,000 Live Links, and a single click-to-update application, you will always have the most current and up-to-date information at your fingertips.

[Click here to learn more](#)

Table of Contents

Projects.....	2
VBAProject.....	2
Documents.....	2
Admin.....	2
(Declarations).....	2
Contacts.....	3
(Declarations).....	3
ConvFlow_Sheet.....	4
(Declarations).....	4
Entries.....	5
(Declarations).....	5
Worksheet_Change [Sub].....	5
ThisWorkbook.....	6
(Declarations).....	6
Transactions.....	7
(Declarations).....	7
Modules.....	8
Contact_Macros.....	8
(Declarations).....	8
ContactData [Sub].....	8
IntialReq_Maros.....	10
(Declarations).....	10
CheckResponse [Sub].....	10
InitialRequest [Sub].....	10
Transaction_Macros.....	12
(Declarations).....	12
AddTransaction [Sub].....	12
TransSummary_RequestDetails [Sub].....	13

1 Option Explicit

2

1 Option Explicit

2

1
2

Option Explicit


```
1 Option Explicit
```

```
2
```

```
3 Private Sub Worksheet_Change(ByVal Target As Range)
```

```
4     If Not Intersect(Target, Range("C4" )) Is Nothing And Range("C4" ).Value <> Empty
```

```
5         Then InitialRequest
```

```
6     End Sub
```

1 Option Explicit

2

1	Option Explicit
2	

```

1 Option Explicit
2 Dim AIResp As String, AIBody As String, Prompt As String, Response As String,
   ParsedAnswer As String
3 Dim Request As String, PromptCond As String, RespArr() As String, NotifText As String
4 Dim ContArr() As String, ContName As String
5 Dim ContRow As Long, ContCol As Long
6
7 Sub ContactData(RequestType As String)
8     Request = Entries.Range("C4").Value
9     PromptCond = Entries.Range("O5").Value
10    Prompt = Replace(Replace(PromptCond & Request & "\n If any information is missing,
   do not add sample or made-up data", vbNewLine, "\n"), Chr(10), "\n")
11    'Debug.Print Prompt
12    oXMLHTTP.Open "POST", "https://api.openai.com/v1/completions", False 'Set Open
   Posts For Text
13    AIBody = "{" & "model" & ":" & " " & Model & " ", "prompt" & ":" & " " & Prompt & " ",
   "temperature" & ":" & Temp & " ", "max_tokens" & ":" & 3500 & "}" 'Request Body For
   Text Response
14    oXMLHTTP.setRequestHeader "Content-Type", "application/json" 'Set Header Type
15    oXMLHTTP.setRequestHeader "Authorization", "Bearer " & APIKey 'Set Header
   Authorization
16    'Debug.Print AIBody
17    oXMLHTTP.send AIBody
18    Response = oXMLHTTP.responseText 'Extract Response
19    'Debug.Print Response
20    CheckResponse 'Check Response Conditions
21    ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "{" & Chr(34) &
   "text" & Chr(34) & ":" & Chr(34)) - 9) 'Remove Left Json Response
22    ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34) &
   "index" ) - 1) 'Remove Right Json Response
23    If InStr(ParsedAnswer, "\n\n") > 0 Then ParsedAnswer = Right(ParsedAnswer, Len(
   ParsedAnswer) - InStr(ParsedAnswer, "\n\n") - 3) 'Remove anything to the left of
   \n\n
24    RespArr = Split(ParsedAnswer, "\n")
25    'Check If headers were included in response
26    If InStr(ParsedAnswer, "Contact Name") > 0 Then ContArr = Split(RespArr(1), "|" )
   Else ContArr = Split(RespArr(0), "|" )
27    If RequestType = "Add Data" Then 'Add Data (new records)
28        ContRow = Contacts.Range("A99999").End(xlUp).Row + 1 'First Available Row For
   New Contacts
29    Else 'Update/Delete Contact (Find Contact Row)
30        ContName = Trim(ContArr(0)) 'Contact Name 'First Name in the array
31        ContRow = 0 'Reset Contact Row to zero initially
32        On Error Resume Next
33        ContRow = Contacts.Range("Cont_Name").Find(ContName, , xlValues, xlWhole).Row
   'Extract Row From Contact Names (If found)
34        On Error GoTo 0
35        If ContRow = 0 Then
36            MsgBox "Contact not found, Please check if the name exist or possible
   spelling differences"
37            Exit Sub
38        End If
39        'Check for Delete or Add/Update
40        If RequestType = "Delete Data" Then
41            Contacts.Range(ContRow & ":" & ContRow).Delete 'Delete Contact Row
42            NotifText = ContName & " has been deleted from Contacts"
43            GoTo Notification 'Skip To User Notification
44        End If
45    End If
46    NotifText = "The following contact has been " 'Set Notification pretext
47    If RequestType = "Update Data" Then NotifText = NotifText & "updated" & vbCrLf

```

1

```
1
48 Else: NotifText = NotifText & "added" & vbCrLf
49 For ContCol = 1 To 7
50     If Trim(ContArr(ContCol - 1)) <> "" And InStr(ContArr(ContCol - 1), "---" ) = 0
51         Then Contacts.Cells(ContRow, ContCol).Value = Trim(ContArr(ContCol - 1))
52         'Add/Update Contact Information & remove any spaces before/after
53         NotifText = NotifText & Contacts.Cells(2, ContCol).Value & ": " & Contacts.Cells
54         (ContRow, ContCol).Value & vbCrLf 'Update Contact Text w/ headers & values
55     Next ContCol
56     Notification:
57     Entries.Shapes("ResponseShp").TextFrame2.TextRange.Text = NotifText 'Update
58     Notification Shape
59 End Sub
```

```

1 Option Explicit
2 Public RequestType As String, APIKey As String, Temp As String, Model As String, Lang
  As String, Response As String
3 Public oXMLHTTP As Object
4 Dim AIResp As String, AIBody As String, Prompt As String, ParsedAnswer As String,
  Context As String
5 Dim Request As String, PromptCond As String, RespArr() As String, Database As String
6
7
8 Sub CheckResponse()
9   'Check For Issues in response and respond accordingly
10  If InStr(Response, "!DOCTYPE html" ) > 0 Then 'This will appear when there is a
    site issue or an API issue
11    MsgBox "The OpenAI Server appears to be having issues. Please try again a bit
    later"
12    End
13    Exit Sub
14  End If
15  If InStr(Response, "You didn't provide an API key" ) > 0 Then 'Missing API Key
16    MsgBox "No API Key was added. You can find your API key at " & vbCrLf &
    "https://platform.openai.com/account/api-keys."
17    End
18    Exit Sub
19  End If
20  If InStr(Response, "Incorrect API key provided" ) > 0 Then 'Incorrect API Key
21    MsgBox "Incorrect API key provided. You can find your API key at " & vbCrLf &
    "https://platform.openai.com/account/api-keys."
22    End
23    Exit Sub
24  End If
25  If InStr(Response, "maximum context length" ) > 0 Then 'Content submitted is too
    large
26    MsgBox "You have reached the maximum content length. Please clear the history
    above to move forward"
27    End
28    Exit Sub
29  End If
30 End Sub
31
32 Sub InitialRequest()
33   APIKey = Admin.Range("D4" ).Value 'Secret Key
34   Model = Admin.Range("D5" ).Value 'Model
35   Temp = Admin.Range("D6" ).Value 'Temperature
36   If [Language] <> "English" & [Language] <> "" Then Lang = "Please make sure
    response are in " & [Language] & " language" Else Lang = ""
37   If Model = "" Then Model = "text-davinci-003" 'Set Default Model
38   If Temp = "" Then Temp = "0.4" 'Set Default Temp
39   Request = Entries.Range("C4" ).Value
40   PromptCond = Entries.Range("O4" ).Value
41   Prompt = PromptCond & "Text:\n" & Request
42   Set oXMLHTTP = CreateObject("MSXML2.ServerXMLHTTP" )
43   oXMLHTTP.Open "POST" , "https://api.openai.com/v1/completions" , False 'Set Open
    Posts For Text
44   AIBody = "{" "model" ": " & Model & " ", " "prompt" ": " & Prompt & " ",
    " "temperature" ": " & Temp & " , " "max_tokens" ": 3500}" 'Request Body For
    Text Response
45   oXMLHTTP.setRequestHeader "Content-Type" , "application/json" 'Set Header Type
46   oXMLHTTP.setRequestHeader "Authorization" , "Bearer " & APIKey 'Set Header
    Authorization
47   'Debug.Print AIBody
48   oXMLHTTP.send AIBody

```

1

```
1
49 Response = oXMLHTTP.responseText 'Extract Response
50 Debug.Print Response
51 CheckResponse 'Run Macro To Check For Response Conditions
52 ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "{" & Chr(34) &
    "text" & Chr(34) & ":" & Chr(34)) - 9) 'Remove Left Json Response
53 ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34) &
    "index" ) - 1) 'Remove Right Json response
54 If InStr(ParsedAnswer, "\n\n" ) > 0 Then ParsedAnswer = Right(ParsedAnswer, Len(
    ParsedAnswer) - InStr(ParsedAnswer, "\n\n" ) - 3) 'Remove anything to the left of
    \n\n
55 Debug.Print ParsedAnswer
56 RespArr = Split(ParsedAnswer, "," ) 'Split
57 RequestType = Trim(RespArr(0)) 'Determind Request Type
58 Database = Trim(RespArr(1)) 'Determine Database
59
60 Select Case RequestType
61     Case Is = "Add Data" , "Update Data" , "Delete Data"
62         If Database = "Contacts" Then ContactData (RequestType)
63         If Database = "Transactions" Then AddTransaction (RequestType)
64     Case Is = "Summarize Data"
65         If Database = "Transactions" Then TransSummary_RequestDetails
66     End Select
67 End Sub
```

```

1 Option Explicit
2 Public DateContext As String, ThisDay As String, ThisMonth As String, LastMonth As
String, NextMonth As String, ThisYear As String
3 Dim AIResp As String, AIBody As String, Prompt As String, ParsedAnswer As String
4 Dim Request As String, PromptCond As String, RespArr() As String, NotifText As String
5 Dim TransArr() As String, RequestType As String, CriteriaData() As String, TransData As
String
6 Dim LastRow As Long, LastResultRow As Long, ResultRow As Long, ResultCol As Long
7 Dim TransRow As Long, TransCol As Long, CritNumb As Long
8
9 Sub AddTransaction(RequestType As String)
10     'Add Current Date Context
11     ThisDay = "The current date is " & Format(Entries.Range("S2").Value, "mm/dd/yyyy"
) & "\n" 'Set the current date for context
12     ThisMonth = "The current month start date is " & Format(Entries.Range("T2").Value,
"mm/dd/yyyy" ) & "\n"
13     LastMonth = "Last month start date is " & Format(Entries.Range("U2").Value,
"mm/dd/yyyy" ) & "\n"
14     NextMonth = "Next month start date is " & Format(Entries.Range("V2").Value,
"mm/dd/yyyy" ) & "\n"
15     ThisYear = "The current start of year date is " & Format(Entries.Range("W2").Value
, "mm/dd/yyyy" ) & "\n"
16     DateContext = ThisDay & ThisMonth & LastMonth & NextMonth & ThisYear & "\n" 'Set
the current date for context
17     Request = Entries.Range("C4").Value & "\n" 'Current Request
18     PromptCond = Entries.Range("O6").Value & "\n" & DateContext 'Promt With Date
Context
19     Prompt = Replace(Replace(Replace(PromptCond & Request, vbNewLine, "\n" ), Chr(10),
"\n" ), "$", "" ) 'Combine Prompt Conditions & Request, replace new lines and
remove $
20     'Debug.Print Prompt
21     oXMLHTTP.Open "POST", "https://api.openai.com/v1/completions", False 'Set Open
Posts For Text
22     AIBody = "{" & "model" & ":" & " " & Model & " ", "prompt" & ":" & " " & Prompt & " ",
"temperature" & ":" & " " & Temp & " ", "max_tokens" & ":" & " 3500"}" 'Request Body For
Text Response
23     oXMLHTTP.setRequestHeader "Content-Type", "application/json" 'Set Header Type
24     oXMLHTTP.setRequestHeader "Authorization", "Bearer " & APIKey 'Set Header
Authorization
25     Debug.Print AIBody
26     oXMLHTTP.send AIBody
27     Response = oXMLHTTP.responseText 'Extract Response
28     Debug.Print Response
29     CheckResponse 'Run Macro To Check For Response Conditions
30     If InStr(Response, "maximum context length" ) > 0 Then
31         MsgBox "You have reached the maximum context length. Please clear the history
above to move forward"
32         Exit Sub
33     End If
34     ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "{" & Chr(34) &
"text" & Chr(34) & ":" & Chr(34)) - 9)
35     ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34) &
"index" ) - 1)
36     'If InStr(ParsedAnswer, "\n\n") > 0 Then ParsedAnswer = Right(ParsedAnswer,
Len(ParsedAnswer) - InStr(ParsedAnswer, "\n\n") - 3) 'Remove anything to the left
of \n\n
37     If Left(ParsedAnswer, 2) = "\n" Then ParsedAnswer = Right(ParsedAnswer, Len(
ParsedAnswer) - 2) 'Remove First \n if existing
38     'Set Criteria
39     If InStr(ParsedAnswer, "Amount" ) > 0 Then 'Response Contains more than 1 rows
(first row is header)

```

1 2


```

1 2
40     RespArr = Split(ParsedAnswer, "\n" ) 'Split First/Second Row
41     TransArr = Split(RespArr(1), "|" ) 'Extract Second Row into array
42     Else
43     TransArr = Split(ParsedAnswer, "|" ) 'Extract only row into array
44     End If
45     'If InStr(ParsedAnswer, "Amount") > 0 Then TransArr = Split(RespArr(1), "|") Else
TransArr = Split(RespArr(0), "|") 'Split Based on Transaction Columns (with and
without headers)
46     TransRow = Transactions.Range("A99999" ).End(xlUp).Row + 1 'First available row
47     NotifText = "The following transaction has been added:" & vbCrLf
48     For TransCol = 1 To 4
49     Transactions.Cells(TransRow, TransCol).Value = Trim(TransArr(TransCol - 1))
50     NotifText = NotifText & Transactions.Cells(2, TransCol).Value & ": " &
Transactions.Cells(TransRow, TransCol).Value & vbCrLf 'Update Transaction Text
w/ headers & values
51     Next TransCol
52     Entries.Shapes("ResponseShp" ).TextFrame2.TextRange.Text = NotifText
53     End Sub
54
55     Sub TransSummary_RequestDetails()
56     Request = Entries.Range("C4" ).Value & "\n" 'Original Request
57     ThisDay = "The current date is " & Format(Entries.Range("S2" ).Value, "mm/dd/yyyy"
) & "\n" 'Set the current date for context
58     ThisMonth = "The current month start date is " & Format(Entries.Range("T2" ).Value,
"mm/dd/yyyy" ) & "\n"
59     LastMonth = "Last month start date is " & Format(Entries.Range("U2" ).Value,
"mm/dd/yyyy" ) & "\n"
60     NextMonth = "Next month start date is " & Format(Entries.Range("V2" ).Value,
"mm/dd/yyyy" ) & "\n"
61     ThisYear = "The current start of year date is " & Format(Entries.Range("W2" ).Value
, "mm/dd/yyyy" ) & "\n"
62     PromptCond = Entries.Range("O7" ).Value & "\n" & DateContext 'Promt With Date
Context
63     Prompt = Replace(Replace(Replace(PromptCond & Request, vbCrLf, "\n" ), Chr(10),
"\n" ), "$" , "" )
64     oXMLHTTP.Open "POST" , "https://api.openai.com/v1/completions" , False 'Set Open
Posts For Text
65     AIBody = "{" "model" ": " " & Model & "" , " "prompt" ": " " " & Prompt & "" ,
" "temperature" ": " & Temp & "" , " "max_tokens" ": 3500}" 'Request Body For
Text Response
66     oXMLHTTP.setRequestHeader "Content-Type" , "application/json" 'Set Header Type
67     oXMLHTTP.setRequestHeader "Authorization" , "Bearer " & APIKey 'Set Header
Authorization
68     Debug.Print AIBody
69     oXMLHTTP.send AIBody
70     Response = oXMLHTTP.responseText 'Extract Response
71     Debug.Print Response
72     CheckResponse 'Check Response Conditions
73     ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "[{" & Chr(34) &
"text" & Chr(34) & ":" & Chr(34)) - 9) 'Remove Left Json Response
74     ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34) &
"index" ) - 1) 'Remove Right Json Response
75     If InStr(ParsedAnswer, "\n\n" ) > 0 Then ParsedAnswer = Right(ParsedAnswer, Len(
ParsedAnswer) - InStr(ParsedAnswer, "\n\n" ) - 3) 'Remove anything to the left of
\n\n
76     If Left(ParsedAnswer, 2) = "\n" Then ParsedAnswer = Right(ParsedAnswer, Len(
ParsedAnswer) - 2) 'Remove First line return (if any)
77     Debug.Print ParsedAnswer
78     If InStr(ParsedAnswer, "No response" ) > 0 Then
79     MsgBox "No response from server"
80     Exit Sub

```

1 2

```

1 2
81 End If
82 With Transactions
83   Transactions.Range("I3:S9999").ClearContents 'Clear Previous Criteria
84   'Set Criteria
85   If InStr(ParsedAnswer, "Amount") > 0 Then 'Response Contains more than 1 rows
86     (first row is header)
87     RespArr = Split(ParsedAnswer, "\n") 'Split First/Second Row
88     CriteriaData = Split(RespArr(1), "|") 'Extract Second Row into array
89   Else
90     CriteriaData = Split(ParsedAnswer, "|") 'Extract only row into array
91   End If
92   For CritNum = 0 To 3
93     .Cells(3, 9 + CritNum).Value = Trim(CriteriaData(CritNum)) 'Remove spaces
94   Next CritNum
95   If .Range("I3").Value <> Empty Then .Range("I3").Value = ">=" & .Range("I3").
96     Value 'Add Greater than or equal to From date
97   If .Range("J3").Value <> Empty Then .Range("J3").Value = "<=" & .Range("J3").
98     Value 'Add Less Than or equal to To date
99   LastRow = .Range("A9999").End(xlUp).Row
100   If LastRow < 4 Then
101     MsgBox "No transaction data exists"
102     Exit Sub
103   End If
104   If .Range("L3").Value <> "Expense" Or .Range("L3").Value <> "Income" Then .
105     Range("L3").ClearContents 'Clear any criteria other than Expense and income
106   .Range("A2:D" & LastRow).AdvancedFilter xlFilterCopy, CriteriaRange:=.Range(
107     "I2:M3"), CopyToRange:=.Range("P2:S2"), Unique:=True
108   LastResultRow = .Range("S9999").End(xlUp).Row
109   If LastResultRow < 3 Then
110     MsgBox "No transaction data meets that criteria"
111     Exit Sub
112   End If
113   'Add Transaction data in a string (include header)
114   TransData = "" 'Clear Out Transaction Data String
115   For TransRow = 2 To LastResultRow
116     For TransCol = 16 To 19
117       TransData = TransData & .Cells(TransRow, TransCol).Value & " | " 'Add
118       Spaces & Column Separator
119     Next TransCol
120     TransData = TransData & "\n" 'Add Transaction amounts
121     'Add Row Separator
122   Next TransRow
123   TransData = TransData & "With a total of " & Transactions.Range("V1").Value &
124     "\n" 'Add in actual total (ChatGPT seems bad at math)
125   Debug.Print TransData
126   PromptCond = Entries.Range("O8").Value 'Prompt Requirements
127   Prompt = PromptCond & "\n\n" & TransData & "\n\n" & Request & DateContext 'Add
128   Original Prompt Conditions, with Transaction data, Original Request and Date
129   Context
130   oXMLHTTP.Open "POST", "https://api.openai.com/v1/completions", False 'Set Open
131   Posts For Text
132   AIBody = "{" & "model" & ": " & Model & ", " & "prompt" & ": " & Prompt & "
133     ", " & "temperature" & ": " & Temp & ", " & "max_tokens" & ": 3500}" 'Request Body
134   For Text Response
135   oXMLHTTP.setRequestHeader "Content-Type", "application/json" 'Set Header Type
136   oXMLHTTP.setRequestHeader "Authorization", "Bearer " & APIKey 'Set Header
137   Authorization
138   'Debug.Print AIBody
139   oXMLHTTP.send AIBody
140   Response = oXMLHTTP.responseText 'Extract Response

```

1 2

```
129 1 2 ParsedAnswer = Right(Response, Len(Response) - InStrRev(Response, "{" & Chr(34)
      & "text" & Chr(34) & ":" & Chr(34)) - 9) 'Remove Left Json Response
130 ParsedAnswer = Left(ParsedAnswer, InStrRev(ParsedAnswer, Chr(34) & "," & Chr(34)
      & "index" ) - 1) 'Remove Right Json Response
131 If InStr(ParsedAnswer, "\n\n" ) > 0 Then ParsedAnswer = Right(ParsedAnswer, Len(
      ParsedAnswer) - InStr(ParsedAnswer, "\n\n" ) - 3) 'Remove anything to the left
      of \n\n
132 If Left(ParsedAnswer, 2) = "\n" Then ParsedAnswer = Right(ParsedAnswer, Len(
      ParsedAnswer) - 2) 'Remove First line return (if any)
133 ParsedAnswer = Replace(Replace(ParsedAnswer, "\n", vbNewLine), "\" & Chr(34),
      Chr(34))
134 With Entries.Shapes("ResponseShp" )
135     .TextFrame2.TextRange.Text = ParsedAnswer
136     .Visible = msoCTrue
137 End With
138 End With
139 End Sub
```

A

AddTransaction, 11, 12
 Admin, 10
 AdvancedFilter, 14
 AIBody, 8, 10, 12-14
 AIResp, 8, 10, 12
 APIKey, 8, 10, 12-14

C

Cells, 9, 13, 14
 CheckResponse, 8, 10-13
 Chr, 8, 11-13, 15
 ClearContents, 14
 ContactData, 8, 11
 Contacts, 8, 9
 ContArr, 8, 9
 ContCol, 8, 9
 Context, 10
 ContName, 8
 ContRow, 8, 9
 CopyToRange, 14
 CreateObject, 10
 CriteriaData, 12, 14
 CriteriaRange, 14
 CritNumb, 12, 14

D

Database, 10, 11
 DateContext, 12-14
 Debug, 11-14
 Delete, 8

E

Empty, 5, 14
 Entries, 8-10, 12-15
 Explicit, 2-8, 10, 12

F

Find, 8
 Format, 12, 13

I

InitialRequest, 5, 10
 InStr, 8-15
 InStrRev, 8, 11-13, 15
 Intersect, 5

L

Lang, 10
 Language, 10
 LastMonth, 12, 13
 LastResultRow, 12, 14
 LastRow, 12, 14
 Left, 8, 11-13, 15
 Len, 8, 11-13, 15

M

Model, 8, 10, 12-14
 MsgBox, 8, 10, 12-14
 msoCTrue, 15

N

NextMonth, 12, 13
 Notification, 8, 9
 NotifText, 8, 9, 12, 13

O

Open, 8, 10, 12-14
 oXMLHTTP, 8, 10-14

P

ParsedAnswer, 8, 10-15
 Print, 11-14
 Prompt, 8, 10, 12-14
 PromptCond, 8, 10, 12-14

R

Range, 5, 8, 10, 12-14
 Replace, 8, 12, 13, 15
 Request, 8, 10, 12-14
 RequestType, 8, 10-12
 RespArr, 8, 10-14
 Response, 8, 10-15
 responseText, 8, 11-14
 ResultCol, 12
 ResultRow, 12
 Right, 8, 11-13, 15
 Row, 8, 13, 14

S

send, 8, 10, 12-14
 setRequestHeader, 8, 10, 12-14
 Shapes, 9, 13, 15
 Split, 8, 11, 13, 14

T

Target, 5
 Temp, 8, 10, 12-14
 Text, 9, 13, 15
 TextFrame2, 9, 13, 15
 TextRange, 9, 13, 15
 ThisDay, 12, 13
 ThisMonth, 12, 13
 ThisYear, 12, 13
 Transactions, 13, 14
 TransArr, 12, 13
 TransCol, 12-14
 TransData, 12, 14
 TransRow, 12-14
 TransSummary_RequestDetails, 11, 13
 Trim, 8, 9, 11, 13, 14

U

Unique, 14

V

Value, 5, 8-10, 12-14
 vbCrLf, 8-10, 13
 vbNewLine, 8, 12, 13, 15
 Visible, 15

W

Worksheet_Change, 5

X

xlFilterCopy, 14
 xlUp, 8, 13, 14
 xlValues, 8
 xlWhole, 8

Thank You!

This source code was created and made available to help you gain a better understanding of how VBA is used to create amazing Excel-based applications.

Thank you so much for your continued shares, likes and support. It really helps.



Excel For Freelancers