

Excel & VBA Sample

- [API-NG Excel VBA Sample Code](#)
 - [Prerequisites](#)
 - [Installation](#)
 - [How to run](#)
 - [JSON-RPC](#)
 - [RESCRIPT \(JSON\)](#)
 - [Code Snippets](#)
 - [Calling API-NG](#)
 - [Calling API-NG via JSON-RPC](#)
 - [Calling API-NG via RESCRIPT](#)
 - [Ascertain the event type id for Horse Racing using listEventTypes](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Get next available horse racing market and runner information using listMarketCatalogue](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Get available back prices for the next horse racing Market using listMarketBook](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Place a bet on first runner from next horse racing market using placeOrders](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Other Common Code](#)

API-NG Excel VBA Sample Code

Prerequisites

- Microsoft Excel 2007 or later

Installation

None required

Clone the repository at <https://github.com/betfair/API-NG-sample-code/tree/master/vba>

How to run

Open the Excel workbook. Obtain an app key and session token and enter them into sheet **Example** cells **B3** and **B4** respectively.

JSON-RPC

- Click **Clear**
- Click **Go (JSON-RPC)** button

RESCRIPT (JSON)

- Click **Clear**
- Click **Go (RESCRIPT)** button

Code Snippets

Calling API-NG

```

Function SendRequest(Url, AppKey, Session, Data) As String
    On Error GoTo ErrorHandler:
    Dim xhr: Set xhr = CreateObject("MSXML2.XMLHTTP")

    With xhr
        .Open "POST", Url & "/", False
        .setRequestHeader "X-Application", AppKey
        .setRequestHeader "Content-Type", "application/json"
        .setRequestHeader "Accept", "application/json"
    End With

    If Session <> "" Then
        xhr.setRequestHeader "X-Authentication", Session
    End If

    xhr.send Data
    SendRequest = xhr.responseText

    If xhr.Status <> 200 Then
        Err.Raise vbObjectError + 1000, "Util.SendRequest", "The call to API-NG was unsuccessful. Status code:
" & xhr.Status & " " & xhr.statusText & ". Response was: " & xhr.responseText
    End If

    Set xhr = Nothing
    Exit Function

ErrorHandler:
    HandleError
End Function

```

Calling API-NG via JSON-RPC

```

Dim Request: Request = "{""jsonrpc"": ""2.0"", ""method"": ""SportsAPING/v1.0/listEventTypes"", ""params"":
{""filter"":{}}, ""id"": 1}"
Dim Url: Url = "https://api.betfair.com/json-rpc/"

Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(Url, "your app key", "your session
token", Request)

```

Calling API-NG via RESCRIPT

```

Dim Request: Request = "{""filter"":{}}"
Dim Url: Url = "https://api.betfair.com/rest/v1.0/listEventTypes/"
Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(Url, "your app key", "your session
token", Request)

```

Ascertain the event type id for Horse Racing using listEventTypes

Common

```

Function GetListEventTypesRequestString() As String
    GetListEventTypesRequestString = "{"filter":{}}"
End Function

Function GetEventTypeIdFromEventTypes(ByVal EventTypes As Object) As String
    GetEventTypeIdFromEventTypes = "0"

    Dim Index As Integer
    For Index = 1 To EventTypes.Count Step 1
        Dim EventType: Set EventType = EventTypes.Item(Index).Item("eventType")
        If EventType.Item("name") = "Horse Racing" Then
            GetEventTypeIdFromEventTypes = EventType.Item("id")
            Exit For
        End If
    Next
End Function

```

JSON-RPC

```

Dim Request: Request = MakeJsonRpcRequestString(ListEventTypesMethod, GetListEventTypesRequestString())
Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(), "", Request)
Dim EventTypeResult: Set EventTypeResult = ParseJsonRpcResponseToCollection(ListEventTypesResponse)
Dim EventTypeId: EventTypeId = GetEventTypeIdFromEventTypes(EventTypeResult)

```

RESCRIPT

```

Public Const ListEventTypesMethod As String = "listEventTypes"
Dim Request: Request = MakeJsonRpcRequestString(ListEventTypesMethod, GetListEventTypesRequestString())
Dim ListEventTypesResponse As String: ListEventTypesResponse = SendRequest(GetRestUrl() + ListEventTypesMethod, GetAppKey(), "", Request)
Dim EventTypeResult: Set EventTypeResult = ParseRestResponseToCollection(ListEventTypesResponse)
Dim EventTypeId: EventTypeId = GetEventTypeIdFromEventTypes(EventTypeResult)

```

Get next available horse racing market and runner information using listMarketCatalogue

Common

```

Function GetListMarketCatalogueRequestString(ByVal EventTypeId As String) As String
    Dim dateNow As Date: dateNow = Format(Now, "yyyy-mm-dd hh:mm:ss")
    GetListMarketCatalogueRequestString = "{"filter":{"eventTypeIds":["" & EventTypeId & ""],"marketCountries":["GB"],"marketTypeCodes":["WIN"],"marketStartTime":{"from":"" & dateNow & ""},"sort":["FIRST_TO_START"],"maxResults":"1","marketProjection":["RUNNER_DESCRIPTION"]}"
End Function

Function GetMarketIdFromMarketCatalogue(ByVal Response As Object) As String
    GetMarketIdFromMarketCatalogue = Response.Item(1).Item("marketId")
End Function

```

JSON-RPC

```

Dim Request: Request = MakeJsonRpcRequestString(ListMarketCatalogueMethod, GetListMarketCatalogueRequestString(EventTypeId))
Dim ListMarketCatalogueResponse As String: ListMarketCatalogueResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(), "", Request)
Dim MarketCatalogue: Set MarketCatalogue = ParseJsonRpcResponseToCollection(ListMarketCatalogueResponse)
Dim MarketId: MarketId = GetMarketIdFromMarketCatalogue(MarketCatalogue)

```

RESCRIPT

```

Public Const ListMarketCatalogueMethod As String = "listMarketCatalogue"
Dim Request: Request = GetListMarketCatalogueRequestString(EventTypeId)
Dim ListMarketCatalogueResponse As String: ListMarketCatalogueResponse = SendRequest(GetRestUrl() +
ListMarketCatalogueMethod, GetAppKey(), "", Request)
Dim MarketCatalogue: Set MarketCatalogue = ParseRestResponseToCollection(ListMarketCatalogueResponse)
Dim MarketId: MarketId = GetMarketIdFromMarketCatalogue(MarketCatalogue)

```

Get available back prices for the next horse racing Market using listMarketBook

Common

```

Function GetListMarketBookRequestString(ByVal MarketId As String) As String
    GetListMarketBookRequestString = "{" & "marketIds":["" & MarketId & ""], "priceProjection": {"priceData":
["EX_BEST_OFFERS"]} & "}"
End Function

Function GetSelectionIdFromMarketBook(ByVal Response As Object) As String
    Dim Runners As Object: Set Runners = Response.Item(1).Item("runners")
    GetSelectionIdFromMarketBook = Runners.Item(1).Item("selectionId")
    Set Runners = Nothing
End Function

Function GetAvailableToBackForSelection(ByVal SelectionId As String, ByVal Response As Object) As Collection
    Dim Runners As Object: Set Runners = Response.Item(1).Item("runners")

    Dim Index As Integer
    For Index = 1 To Runners.Count Step 1
        Dim Id: Id = Runners.Item(Index).Item("selectionId")
        If Id = SelectionId Then
            Set GetAvailableToBackForSelection = Runners.Item(Index).Item("ex").Item("availableToBack")
            Exit For
        End If
    Next

    Set Runners = Nothing
End Function

```

JSON-RPC

```

Dim Request: Request = MakeJsonRpcRequestString(ListMarketBookMethod, GetListMarketBookRequestString(MarketId))
Dim ListMarketBookResponse As String: ListMarketBookResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(), "",
Request)

Dim MarketBook: Set MarketBook = ParseJsonRpcResponseToCollection(ListMarketBookResponse)
Dim SelectionId: SelectionId = GetSelectionIdFromMarketBook(MarketBook)

Dim AvailableToBack As Object: Set AvailableToBack = GetAvailableToBackForSelection(SelectionId, MarketBook)

```

RESCRIPT

```

Public Const ListMarketBookMethod As String = "listMarketBook"
Dim Request: Request = GetListMarketBookRequestString(MarketId)
Dim ListMarketBookResponse As String: ListMarketBookResponse = SendRequest(GetRestUrl() + ListMarketBookMethod,
GetAppKey(), "", Request)
Dim MarketBook: Set MarketBook = ParseRestResponseToCollection(ListMarketBookResponse)
Dim SelectionId: SelectionId = GetSelectionIdFromMarketBook(MarketBook)
Dim AvailableToBack As Object: Set AvailableToBack = GetAvailableToBackForSelection(SelectionId, MarketBook)

```

Place a bet on first runner from next horse racing market using placeOrders

Common

```

Function GetPlaceOrdersRequestString(ByVal MarketId As String, ByVal SelectionId As String, ByVal Price As
String) As String
    GetPlaceOrdersRequestString = "{" & "marketId":" & MarketId & "," & "instructions":[{"selectionId":" &
SelectionId & ","handicap":"0","side":"BACK","orderType":"LIMIT","limitOrder":{"size":"
0.01","price":" & Price & ","persistenceType":"LAPSE"}]}]"
End Function

```

JSON-RPC

```

Dim Price: Price = AvailableToBack.Item(1).Item("price")

Dim Request: Request = MakeJsonRpcRequestString(PlaceOrdersMethod, GetPlaceOrdersRequestString(MarketId,
SelectionId, Dim PlaceOrdersResponse As String: PlaceOrdersResponse = SendRequest(GetJsonRpcUrl(), GetAppKey(),
GetSession(), Request)

Dim PlaceExecutionReport: Set PlaceExecutionReport = ParseJsonRpcResponseToCollection(PlaceOrdersResponse)
Dim BetPlacementResult: BetPlacementResult = PlaceExecutionReport.Item("status")

```

RESCRIPT

```

Public Const PlaceOrdersMethod As String = "placeOrders"

Dim Price: Price = AvailableToBack.Item(1).Item("price")

Dim Request: Request = GetPlaceOrdersRequestString(MarketId, SelectionId, Price)
Dim PlaceOrdersResponse As String: PlaceOrdersResponse = SendRequest(GetRestUrl() + PlaceOrdersMethod,
GetAppKey(), GetSession(), Request)

Dim PlaceExecutionReport: Set PlaceExecutionReport = ParseRestResponseToCollection(PlaceOrdersResponse)
Dim BetPlacementResult: BetPlacementResult = PlaceExecutionReport.Item("status")

```

Other Common Code

```

Function ParseJsonRpcResponseToCollection(ByVal Response As String) As Object
    On Error GoTo ErrorHandler:
    Dim Lib As New JsonLib
    Set ParseJsonRpcResponseToCollection = Lib.parse(Response).Item("result")
    Exit Function

ErrorHandler:
    HandleError
End Function

Function ParseRestResponseToCollection(ByVal Response As String) As Object
    On Error GoTo ErrorHandler:
    Dim Lib As New JsonLib
    Set ParseRestResponseToCollection = Lib.parse(Response)
    Exit Function

ErrorHandler:
    HandleError
End Function

Sub HandleError()

    If Err.Number <> 0 Then
        AppendToLogFile "Error occurred: " & Err.Number & " - " & Err.Description
    End If

    End ' Exit the macro entirely

End Sub

Function MakeJsonRpcRequestString(ByVal Method As String, ByVal RequestString As String) As String
    MakeJsonRpcRequestString = "{" & ""jsonrpc"": ""2.0"", ""method"": ""SportsAPING/v1.0/" & Method & "",
""params"": " & RequestString & ", ""id"": 1}"
End Function

```

- [API-NG Excel VBA Sample Code](#)
 - [Prerequisites](#)
 - [Installation](#)
 - [How to run](#)
 - [JSON-RPC](#)
 - [RESCRIPT \(JSON\)](#)
 - [Code Snippets](#)
 - [Calling API-NG](#)
 - [Calling API-NG via JSON-RPC](#)
 - [Calling API-NG via RESCRIPT](#)
 - [Ascertain the event type id for Horse Racing using listEventTypes](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Get next available horse racing market and runner information using listMarketCatalogue](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Get available back prices for the next horse racing Market using listMarketBook](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Place a bet on first runner from next horse racing market using placeOrders](#)
 - [Common](#)
 - [JSON-RPC](#)
 - [RESCRIPT](#)
 - [Other Common Code](#)