

# RISE API v8

---

## CHANGE REQUEST

---

We experienced an issue that had not previously been accounted for in the development of the RISE API program. We happened upon an instance where a race field was not able to be drawn until after the running of 2 other races in the same meeting. This Heats/Final scenario meant that we had a meeting that had one race as a header only – no runners (totally fine – meetings are quite often released with one race’s field not yet available – but usually that missing field will be populated the prior day. In this instance where it was the same day it was after the running of the heats which were race 1 and race 2 of the meeting, that the fields for the final (which was Race 8 on the card) was created and then the file needed to be downloaded again.

Our system is set to ignore meetings that are in progress, so we could not access the data. This is a rare occurrence (might be in 2 meetings across a full year) but we do need to be able to handle these rare occasions.

Would like to add a subroutine “Get Current” that can be manually triggered – from the RISE API GUI - which will download only the current meetings that are in progress. The easiest method would be to create a duplication of the current code that has the ‘current meeting’ restriction removed, however this would negatively impact if any further changes are to be made to the data – resulting eventually in duplication of changes which is not ideal.

The preferred solution would be to add a subroutine to the existing code that allows a flag to be set to override the ‘current meeting’ restrictions. The GetCurrent subroutine should then activate that flag and get only the files for today’s date : “0” and which have the meeting status = partial results

Whilst I’m not familiar with the code I thought I’d include the snippets in the development documents that are related to this. Note this extract from the initial documentation which calls for any meeting that is partially resulted to be ignored:

automatically populated by the script using today’s date and the value of DATE\_RANGE e.g. if “5” and today is 07/09/2021 [day0 in range] the latest would be 12/09/2021 as shown in the call below:

<https://api.rise-digital.com.au/v2/meetings?earliest=2021-09-07&latest=2021-09-12>

IF meetingStage=PARTIAL RESULTS

IGNORE THAT MEETING

ELSE

Save all details for each meeting THEN

FOR EACH meetingCode

Obtain all details for all races *\*\* note that the races information is repeated in the output for the raceAndForm call that is executed next, so this step could simply ascertain the total number of races in a meeting and each of it’s raceCodes and save the rest of the “race data” from the next call along with the runner and form history details.*

<https://api.rise-digital.com.au/v2/meetings/PE090921/races>

FOR EACH RACE in a MEETING using raceCode

Also changes made in v8 to check for completeness of meeting may also need to be circumvented via a flag when the new Get Current routine is triggered. I'm not sure if anything was added but the request (Feedbackv1.07) required the program to check if the meeting downloaded was complete, I think the only change was to the entry in the currMeets.json file but thought I'd mention it just in case.

11/07/2022

We need to "Catch" the Gateway Timeout error and ensure that the meeting that is being created at that point in time is checked for completeness – and if found to be incomplete the entry to the currMeets.json file for the truncated meeting should be removed so that we can restart the program and it will download the data again, this time in full.

Unfortunately, for ease of testing there is no way of determining when this error will occur. It is not necessarily more prevalent at any particular time of day or week that it occurs, but it definitely happens at least once every day or two. I can only suggest that it might be best to choose the maximum number of days to download and reset the currMeets entries each time you attempt a download to make sure that the maximum number of files are being downloaded at each pass, and that towards the end of the week (From Wednesday onwards) there are more meetings scheduled so you statistically might find more occurrences of the issue then.

#### NOTE

When the meeting is called and is partially resulted there will be some fields that have data that didn't previously, a quick list follows. I don't think they should make any difference to the creation of the XML file, but I have inspected and list them here just in case:

raceStatus: "Planned" -> "ResultsDone"

quarter1: null -> 31.7

quarter2: null -> 32.4

quarter3: null -> 30.3

quarter4: null -> 30.1

overallTime: null -> 2:56.9

leadTime: null -> 12.8

mileRate: null -> "2:05.7"

marginFirstToSecond: " "

marginSecondToThird: " " ->

overallTime: null -> "2:11.2"

We have encountered an instance where there is a server-side error that can halt the progress of the API calls. It has brought to light an issue with the program that needs enhanced error handling to overcome.

This error was specific to a single meeting. The program reports the error as fatal, halting any progress and stopping the download for any meetings that are in the queue to be downloaded after the problematic meeting, essentially meaning that we cannot access them. We need to enhance the error handling of the program to present the user with the option to SKIP the meeting that has caused the error and continue with the next meeting in the queue.

For background, the details of the error that we encountered were as follows:

Attempting to download meetings for the 23<sup>rd</sup> December.

Meeting list/queue is (in order)

1. AY231222 - Albany
2. AP231222 – Albion Park
3. GP231222 – Gloucester Park
4. HD231222 - Horsham
5. MD231222 - Maitland
6. MX231222 - Melton
7. EY231222 – Wagga

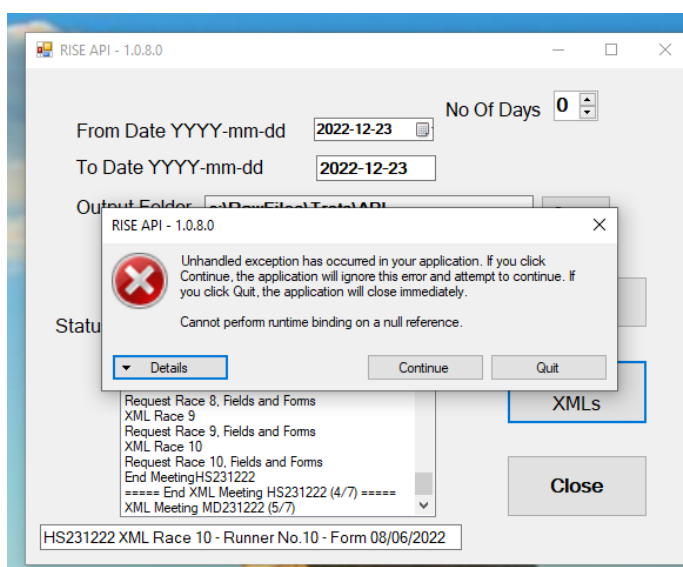
Meetings 1-4 are downloaded ok, however there is an issue when the raceAndForm call is processed for meeting No5. Maitland – the server returns the following message (taken from Manual postman call)

```
"message": "Not Found"
```

This presents the RISE API user with a pop-up error

“The remote server returned an error: (404) Not Found”

And the only option is to quit the program. This means that we are never able to download the meetings 6 & 7 – Melton & Wagga despite there being no problem with these meetings. The error is caught and user notified as shown below, but enhance the program to allow the CONTINUE button to then skip this problem meeting and go to the next meeting in the download list.



### ERROR DETAILS

See the end of this message for details on invoking

just-in-time (JIT) debugging instead of this dialog box.

\*\*\*\*\* Exception Text \*\*\*\*\*

```
Microsoft.CSharp.RuntimeBinder.RuntimeBinderException: Cannot perform runtime binding on a null reference
  at CallSite.Target(Closure , CallSite , Object , String )
  at System.Dynamic.UpdateDelegates.UpdateAndExecute2[T0,T1,TRet](CallSite site, T0 arg0, T1 arg1)
  at CallSite.Target(Closure , CallSite , Object , String )
  at RiseApi.fMain.xmlMeet(Object meet)
  at CallSite.Target(Closure , CallSite , fMain , Object )
  at RiseApi.fMain.processMeetings(Object dMeetings)
  at CallSite.Target(Closure , CallSite , fMain , Object )
  at System.Dynamic.UpdateDelegates.UpdateAndExecuteVoid2[T0,T1](CallSite site, T0 arg0, T1 arg1)
  at RiseApi.fMain.getMeets()
  at RiseApi.fMain.btnGetMeets_Click(Object sender, EventArgs e)
  at System.Windows.Forms.Control.OnClick(EventArgs e)
  at System.Windows.Forms.Button.OnClick(EventArgs e)
  at System.Windows.Forms.Button.OnMouseUp(MouseEventArgs mevent)
  at System.Windows.Forms.Control.WmMouseUp(Message& m, MouseButtons button, Int32 clicks)
  at System.Windows.Forms.Control.WndProc(Message& m)
  at System.Windows.Forms.ButtonBase.WndProc(Message& m)
  at System.Windows.Forms.Button.WndProc(Message& m)
  at System.Windows.Forms.Control.ControlNativeWindow.OnMessage(Message& m)
  at System.Windows.Forms.Control.ControlNativeWindow.WndProc(Message& m)
  at System.Windows.Forms.NativeWindow.Callback(IntPtr hWnd, Int32 msg, IntPtr wparam, IntPtr lparam)
```

\*\*\*\*\* Loaded Assemblies \*\*\*\*\*

mscorlib

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4515.0 built by: NET48REL1LAST\_C  
CodeBase: file:///C:/Windows/Microsoft.NET/Framework/v4.0.30319/mscorlib.dll

RiseApi

Assembly Version: 1.0.8.0  
Win32 Version: 1.0.8.0  
CodeBase: file:///C:/RiseApi/RiseApi.exe

System.Windows.Forms

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4550.0 built by: NET48REL1LAST\_C  
CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Windows.Forms/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.Windows.Forms.dll

System

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4536.0 built by: NET48REL1LAST\_C  
CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.dll

System.Drawing

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4390.0 built by: NET48REL1LAST\_C  
CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Drawing/v4.0\_4.0.0.0\_\_b03f5f7f11d50a3a/System.Drawing.dll

System.Configuration

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4190.0 built by: NET48REL1LAST\_B  
CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Configuration/v4.0\_4.0.0.0\_\_b03f5f7f11d50a3a/System.Configuration.dll

System.Core

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4566.0 built by: NET48REL1LAST\_C  
CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Core/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.Core.dll

System.Xml

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4084.0 built by: NET48REL1  
CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Xml/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.Xml.dll

Newtonsoft.Json

Assembly Version: 13.0.0.0  
Win32 Version: 13.0.1.25517  
CodeBase: file:///C:/RiseApi/Newtonsoft.Json.DLL

System.Numerics

Assembly Version: 4.0.0.0  
Win32 Version: 4.8.4084.0 built by: NET48REL1

CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Numerics/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.Numerics.dll

---

System.Runtime.Serialization

Assembly Version: 4.0.0.0

Win32 Version: 4.8.4536.0 built by: NET48REL1LAST\_C

CodeBase:

file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Runtime.Serialization/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.Runtime.Serialization.dll

---

System.Data

Assembly Version: 4.0.0.0

Win32 Version: 4.8.4536.0 built by: NET48REL1LAST\_C

CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_32/System.Data/v4.0\_4.0.0.0\_\_b77a5c561934e089/System.Data.dll

---

Microsoft.CSharp

Assembly Version: 4.0.0.0

Win32 Version: 4.8.4084.0

CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/Microsoft.CSharp/v4.0\_4.0.0.0\_\_b03f5f7f11d50a3a/Microsoft.CSharp.dll

---

System.Dynamic

Assembly Version: 4.0.0.0

Win32 Version: 4.8.4084.0

CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/System.Dynamic/v4.0\_4.0.0.0\_\_b03f5f7f11d50a3a/System.Dynamic.dll

---

Accessibility

Assembly Version: 4.0.0.0

Win32 Version: 4.8.4084.0 built by: NET48REL1

CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_MSIL/Accessibility/v4.0\_4.0.0.0\_\_b03f5f7f11d50a3a/Accessibility.dll

---

Anonymously Hosted DynamicMethods Assembly

Assembly Version: 0.0.0.0

Win32 Version: 4.8.4515.0 built by: NET48REL1LAST\_C

CodeBase: file:///C:/WINDOWS/Microsoft.Net/assembly/GAC\_32/mscorlib/v4.0\_4.0.0.0\_\_b77a5c561934e089/mscorlib.dll

---

\*\*\*\*\* JIT Debugging \*\*\*\*\*

To enable just-in-time (JIT) debugging, the .config file for this application or computer (machine.config) must have the jitDebugging value set in the system.windows.forms section. The application must also be compiled with debugging enabled.

For example:

```
<configuration>
  <system.windows.forms jitDebugging="true" />
</configuration>
```

When JIT debugging is enabled, any unhandled exception will be sent to the JIT debugger registered on the computer rather than be handled by this dialog box.

## POSTMAN MANUAL INVESTIGATION:

A call for the race list for Maitland via

<https://api.rise-digital.com.au/v2/meetings/MD231222/races>

is successful,

However, the race by race call for the 'raceAndForm' e.g.

<https://api.rise-digital.com.au/v2/races/MD23122203/raceAndForm>

returns the following:

```
{
  "message": "Not Found"
}
```

From Date YYYY-mm-dd No Of Days To Date YYYY-mm-dd Output Folder  ☐ Auto RunStatus 

×

The remote server returned an error: (500) Internal Server Error.