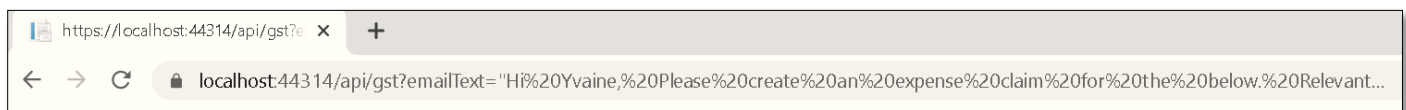


Listed below are the expected output from the RESTful Service:

1. REST Service **accepts a block of text**:

The REST service accepts a block of text as a **query string**. The URL with the sample text provided is as seen below:

```
https://localhost:44314/api/gst?emailText="Hi Yvaine, Please create an expense claim for the below. Relevant details are marked up as requested...
<expense><cost_centre>DEV002</cost_centre><total>1024.01</total><payment_method>personal card</payment_method></expense> From: Ivan Castle Sent: Friday, 16 February 2018 10:32 AM
To: Antoine Lloyd <Antoine.Lloyd@example.com> Subject: test Hi Antoine, Please create a reservation at the <vendor>Viaduct Steakhouse</vendor> our <description>development team's project end celebration dinner</description> on <date>Tuesday 27 April 2017</date>. We expect to arrive around 7.15pm. Approximately 12 people but I'll confirm exact numbers closer to the day. Regards, Ivan"
```



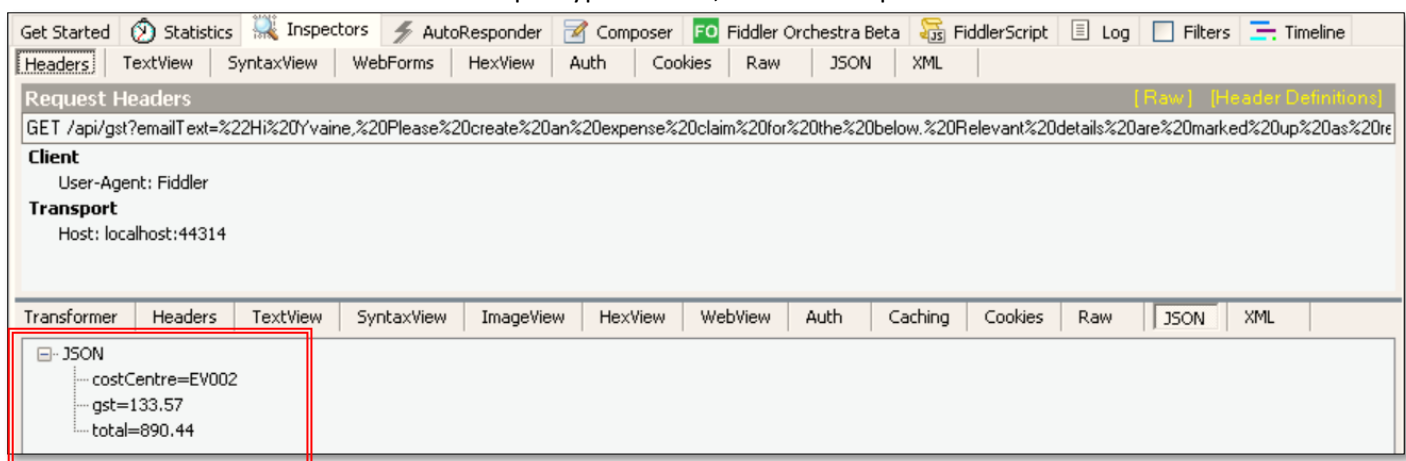
2. The REST Service **extracts the relevant data** from the XML tags and displays the “**cost centre**”, “**total** amount without GST” and the “**GST**” amount.

Below is a screenshot of the output when the sample text is passed to the RESTful Service (URL as seen in point #1 above)

```
<GstInfo xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/EmailExtractRESTfulAPI.Models">
  <costCentre>EV002</costCentre>
  <gst>133.57</gst>
  <total>890.44</total>
</GstInfo>
```

Please note the xml output is the default output generated by the Visual Studio template. I have not added any of the XML codes.

The same URL when run in Fiddler with output type as JSON, the JSON response is as shown in the screenshot below:



The code also handles 3 failure conditions, as listed below:

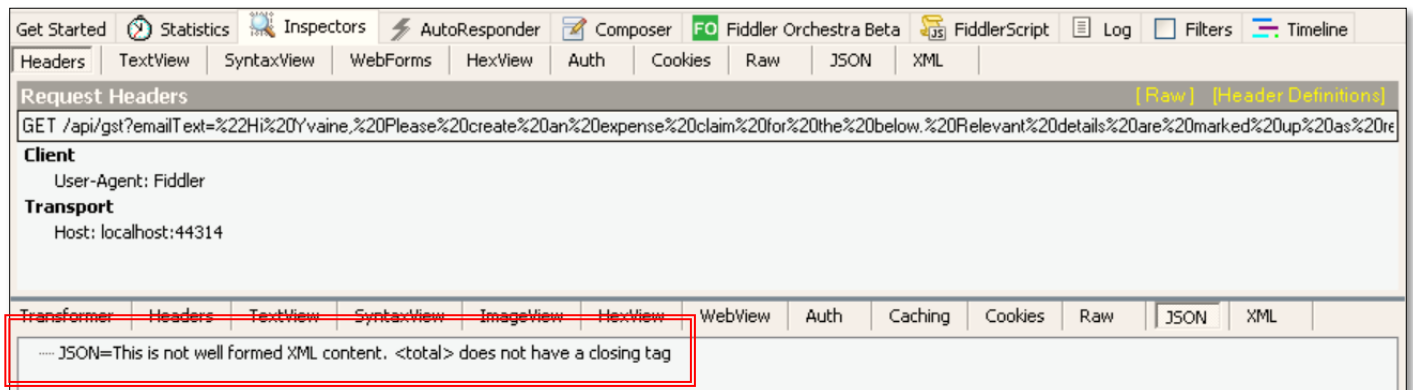
1. The first failure condition is to detect **opening tags that have no corresponding closing tag**. In this case the whole message should be rejected. Below is the input provided, where the <total> closing tag is missing.

```
https://localhost:44314/api/gst?emailText="Hi Yvaine, Please create an expense claim for the below. Relevant details are marked up as requested...
<expense><cost_centre>DEV002</cost_centre><total>1024.01<payment_method>personal
card</payment_method></expense> From: Ivan Castle Sent: Friday, 16 February 2018 10:32 AM
To: Antoine Lloyd <Antoine.Lloyd@example.com> Subject: test Hi Antoine, Please create a
reservation at the <vendor>Viaduct Steakhouse</vendor> our <description>development team's
project end celebration dinner</description> on <date>Tuesday 27 April 2017</date>. We
expect to arrive around 7.15pm. Approximately 12 people but I'll confirm exact numbers
closer to the day. Regards, Ivan"
```

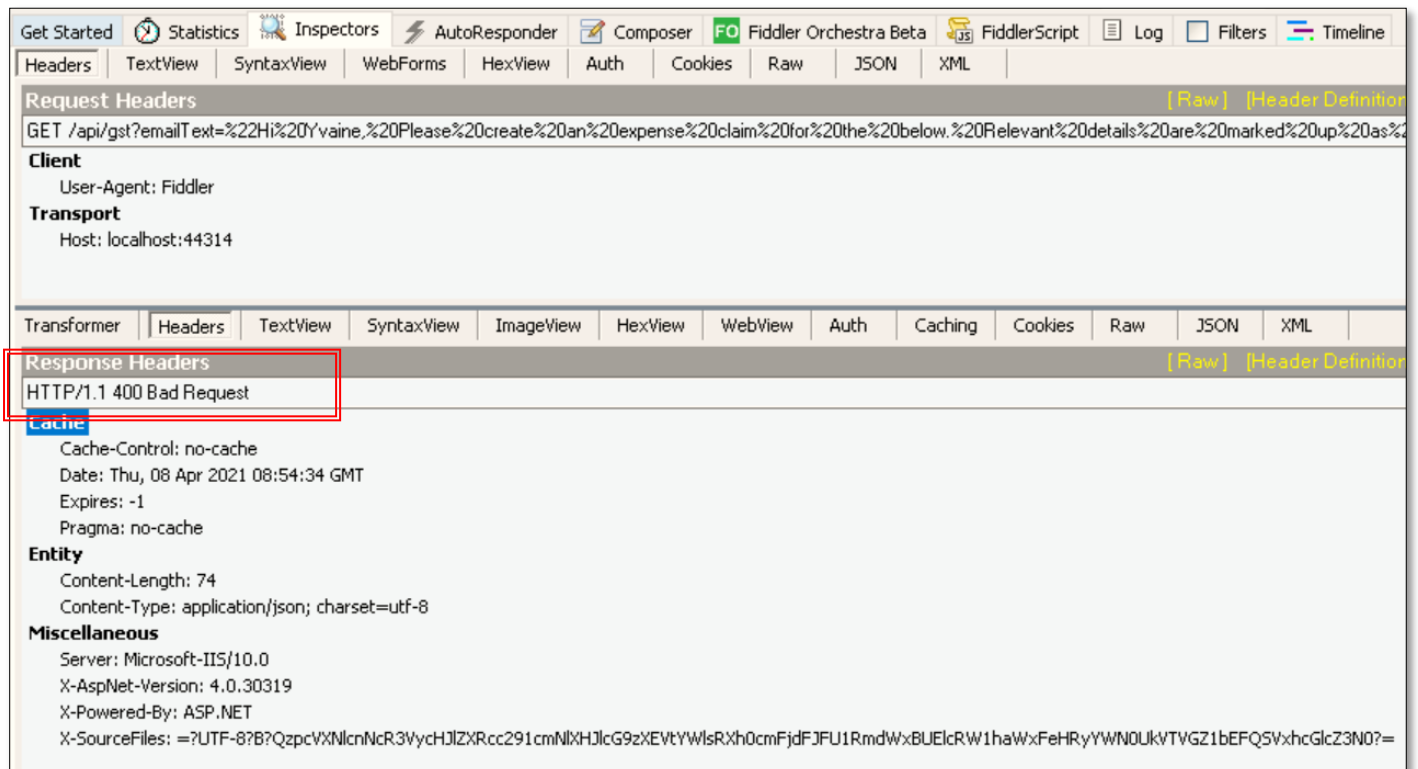
The output gives a response status code of **400, Bad Request**, with a meaningful response mentioning that it is not a well-formed XML and that <total> does not have a closing tag. The default output screenshot is as shown below:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">This is not well formed XML content. <total> does not have a closing tag.</string>
```

The same response in JSON format would look below.



The header with error status code is as seen below:



2. The second failure condition handled is for **missing <total> tag**. In this case the whole message should be rejected. Below is the input provided, where the <total> tag is missing.

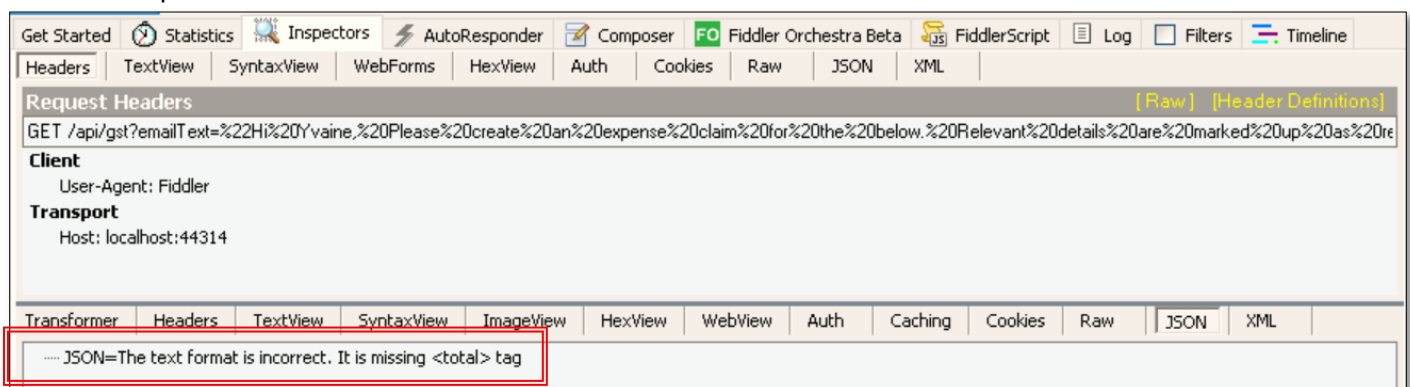
```
https://localhost:44314/api/gst?emailText="Hi Yvaine, Please create an expense claim for the below. Relevant details are marked up as requested...
<expense><cost_centre>DEV002</cost_centre><payment_method>personal
card</payment_method></expense> From: Ivan Castle Sent: Friday, 16 February 2018 10:32 AM
To: Antoine Lloyd <Antoine.Lloyd@example.com> Subject: test Hi Antoine, Please create a
reservation at the <vendor>Viaduct Steakhouse</vendor> our <description>development team's
project end celebration dinner</description> on <date>Tuesday 27 April 2017</date>. We
expect to arrive around 7.15pm. Approximately 12 people but I'll confirm exact numbers
closer to the day. Regards, Ivan"
```

The output gives a response status code of **400, Bad Request**, with a meaningful response mentioning that the text format is incorrect. It is missing the <total> tag.

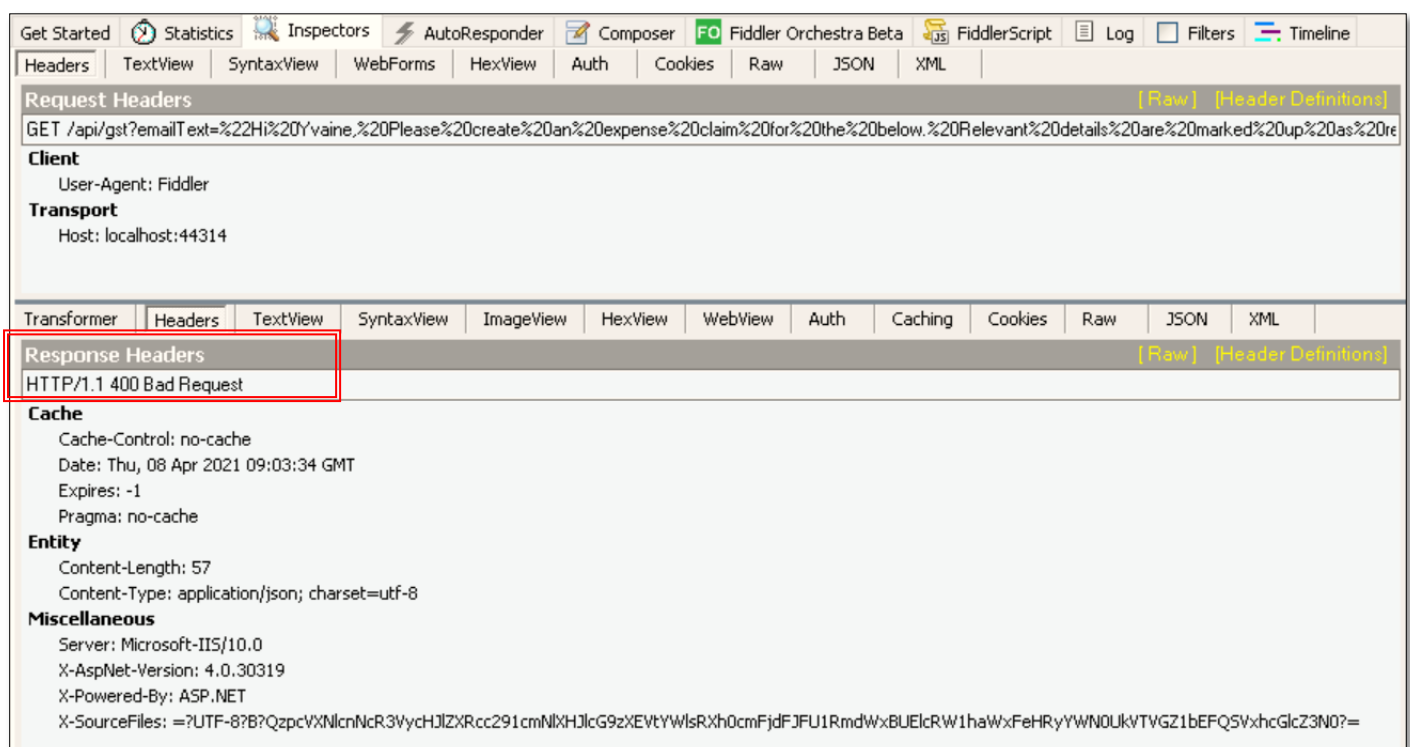
The default output screenshot is as shown below:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">The text format is incorrect. It is missing <total> tag</string>
```

The same response in JSON format would look below.:



The header with error status code is as seen below:



3. The third failure condition is to handle **missing <cost\_centre>**. In this case the 'cost centre' field in the output should be defaulted to **'UNKNOWN'**. Below is the input provided, where the <cost\_centre> tag is missing.

The output gives a response status code of 200, OK, with the cost\_centre shown as UNKNOWN.

```
https://localhost:44314/api/gst?emailText="Hi Yvaine, Please create an expense claim for the below. Relevant details are marked up as requested...
<expense><total>1024.01</total><payment_method>personal card</payment_method></expense>
From: Ivan Castle Sent: Friday, 16 February 2018 10:32 AM To: Antoine Lloyd
<Antoine.Lloyd@example.com> Subject: test Hi Antoine, Please create a reservation at the
<vendor>Viaduct Steakhouse</vendor> our <description>development team's project end
celebration dinner</description> on <date>Tuesday 27 April 2017</date>. We expect to arrive
around 7.15pm. Approximately 12 people but I'll confirm exact numbers closer to the day.
Regards, Ivan"
```

The default output screenshot is as shown below:

```
<GstInfo xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/EmailExtractRESTfulAPI.Models">
  <costCentre>UNKNOWN</costCentre>
  <gst>133.57</gst>
  <total>890.44</total>
</GstInfo>
```

The same response in JSON format would look below.

The screenshot shows the Fiddler interface with the 'JSON' tab selected in the bottom pane. The response body is displayed as a JSON object with three fields: 'costCentre' with the value 'UNKNOWN', 'gst' with the value '133.57', and 'total' with the value '890.44'. The top pane shows the request headers and client information.

The header with a status code 200 OK is as seen below:

The screenshot shows the Fiddler interface with the 'Response Headers' tab selected in the bottom pane. The response status is 'HTTP/1.1 200 OK'. The headers include 'Cache-Control: no-cache', 'Date: Thu, 08 Apr 2021 09:08:58 GMT', 'Expires: -1', 'Pragma: no-cache', 'Content-Length: 52', 'Content-Type: application/json; charset=utf-8', 'Server: Microsoft-IIS/10.0', 'X-AspNet-Version: 4.0.30319', 'X-Powered-By: ASP.NET', and 'X-SourceFiles: =?UTF-8?B?QzpcVXNlcnNrc3VycHJlZXRcc291cmNlXHJlcG9zXEVtYWlsRXh0cmFjdFJFU1RmdWwxBUElcRW1haWwxFehRyYWN0UkVtVGZ1bEFQSVxhcGlcZ3N0?='.