

## Крис Уэбб. Ошибка в Power Query Длина блока не соответствует его дополнению

Это перевод статьи [Криса Уэбба](#), дополненный моими комментариями и использованием Chat GPT (набраны с отступом).

При использовании функции Web.Contents для вызова веб-службы в Power Query может возникнуть ошибка *Длина блока не соответствует его дополнению*. Например

### Запрос 1<sup>1</sup>

```
let
    Источник = Web.Contents("https://api.open-meteo.com/v1/forecast?latitude=52.52&
    longitude=13.41&hourly=temperature_2m")
in
    Источник
```

...вызывает веб-службу и на момент написания статьи выдает ошибку:<sup>2</sup>

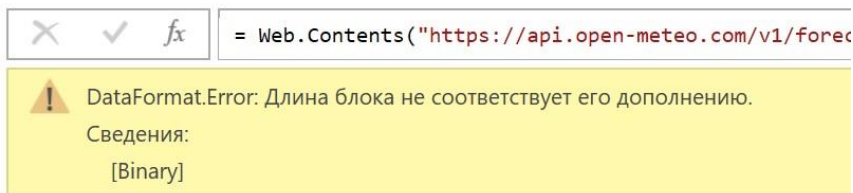


Рис. 1. Ошибка *Длина блока не соответствует его дополнению*

Проблема связана с тем, что некоторые веб-сервисы возвращают данные в сжатом виде *deflate*.

Сжатие *deflate* является одним из методов сжатия данных, который используется во многих веб-сервисах для уменьшения размера передаваемых данных и улучшения производительности. Он основан на алгоритме сжатия *Deflate*, который обычно применяется к HTTP-ответам сервера. Ошибка означает, что Power Query не может правильно обработать сжатые данные.

Есть два способа победить ошибку. Во-первых, можно *попросить* веб-сервис использовать сжатие *gzip*. Вы можете сделать это, воспользовавшись тем, что функция *Web.Contents* помимо url принимает и [довольно много](#) необязательных параметров.

```
Web.Contents(url as text, optional options as nullable record) as binary
```

Нас будет интересовать поле *Headers*. Указание его в виде записи предоставляет дополнительные заголовки для HTTP-запроса. Мы установим для заголовка *Accept-Encoding* значение *gzip*

```
Headers=[#"Accept-Encoding"="gzip"]
```

Полностью запрос преобразуется следующим образом:

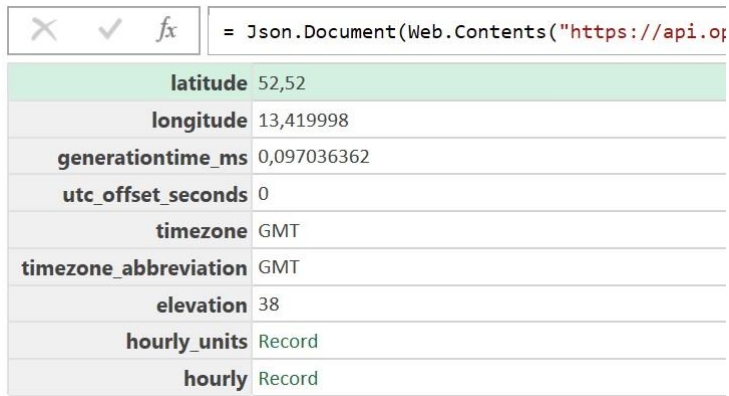
### Запрос2

```
let
    Источник = Json.Document(
        Web.Contents(
            "https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&
            hourly=temperature_2m",
            [Headers=[#"Accept-Encoding"="gzip"]]
        )
    )
in
    Источник
```

<sup>1</sup> Номер соответствует запросу в приложенном Excel-файле. – Прим. Багузина

<sup>2</sup> Разработчики Power Query постоянно выпускают обновления. На момент чтения заметки ошибка может исчезнуть.

Поскольку ошибка не возникла, запрос продолжил обработку данных, возвращенных веб-службой. Распознал, что данные являются документом JSON, и вернул запись:



latitude	52,52
longitude	13,419998
generationtime_ms	0,097036362
utc_offset_seconds	0
timezone	GMT
timezone_abbreviation	GMT
elevation	38
hourly_units	Record
hourly	Record

Рис. 2. Результат успешного запроса

Chat GPT также предложил передать в заголовке запрет на сжатие возвращаемых данных:

### Запрос3

```
let
    Источник = Json.Document(
        Web.Contents(
            "https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&
            hourly=temperature_2m",
            [Headers=[#"Accept-Encoding"="identity"]]
        )
    )
in
    Источник
```

Заголовок Accept-Encoding может принимать различные значения, указывающие на предпочитаемые методы сжатия данных. Некоторые из наиболее распространенных значений включают:

"gzip": Указывает на использование сжатия данных в формате gzip. Gzip является одним из наиболее популярных методов сжатия и широко используется веб-серверами и клиентами.

"deflate": Указывает на использование сжатия данных в формате deflate. Этот метод также широко используется и является альтернативой gzip. Однако некоторые веб-сервисы могут иметь проблемы с правильной обработкой сжатия deflate, как упоминалось в исходном вопросе.

"br": Указывает на использование сжатия данных с использованием алгоритма Brotli. Brotli является более новым методом сжатия, который предлагает лучшую степень сжатия данных по сравнению с gzip или deflate. Однако не все веб-сервисы поддерживают Brotli.

"compress": Указывает на использование сжатия данных в формате compress. Этот метод сжатия редко используется сегодня и редко поддерживается веб-серверами и клиентами.

"identity": Указывает на отсутствие сжатия данных. Если вы хотите получить данные без сжатия, вы можете указать значение "identity" в заголовке Accept-Encoding, как было описано в предыдущем ответе.

Важно отметить, что порядок указания значений в заголовке Accept-Encoding может иметь значение. Клиент указывает свои предпочтения по методам сжатия в порядке предпочтения, начиная с самого предпочитаемого метода. Веб-сервис может использовать эту информацию, чтобы выбрать подходящий метод сжатия или определить, может ли он удовлетворить предпочтения клиента.

Второй вариант обойти ошибку – использовать функцию Web.BrowserContents<sup>3</sup> вместо Web.Contents:

```
Web.BrowserContents(  
  "https://api.open-meteo.com/v1/forecast?latitude=52.52&longitude=13.41&hourly=temperature_2m"  
)
```

Это сложнее, так как функция Web.BrowserContents возвращает HTML-код веб-страницы, просмотренный веб-браузером. Помимо таких проблем, как обработка аутентификации, вам нужно будет проанализировать результат, чтобы получить необходимые данные.

Это выполнимо с помощью функции Html.Table. Ниже приведен пример обработки ответа для этого конкретного вызова веб-службы:

```
let  
  Source = Web.BrowserContents(  
    "https://api.open-meteo.com/v1/forecast?latitude=52.52&  
      longitude=13.41&hourly=temperature_2m"  
  ),  
  HTMLTable = Html.Table(  
    Source,  
    {"JSONPayload", "pre"}  
  ),  
  JSON = Json.Document(  
    HTMLTable{0}[JSONPayload]  
  )  
in  
  JSON
```

---

<sup>3</sup> Я использую Microsoft Office 365, но в моей версии Power Query на 06.05.2023 нет функции Web.BrowserContents. Возможно, Крис тестирует самые последние возможности программы.