## Парсинг html-кода в Power Query

В связи с увлечением Fantasy Premier League меня заинтересовал рейтинг игроков на сайте [livefpl](https://www.livefpl.net/elite).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рис. 1. Web-страница с рейтингом игроков

Чтобы скачать таблицу, я открыл Excel, прошел по меню *Данные* –> *Из Интернета*, и в открывшемся окне ввел url-адрес https://www.livefpl.net/elite. После несложных преобразований я получил изящную таблицу:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 2. Таблица в Excel с рейтингом игроков

С этой таблицей лишь одна проблема – в ней потеряны ссылки на официальные аккаунты игроков на сайте [fantasy.premierleague.com](https://fantasy.premierleague.com/entry/4305040/history).

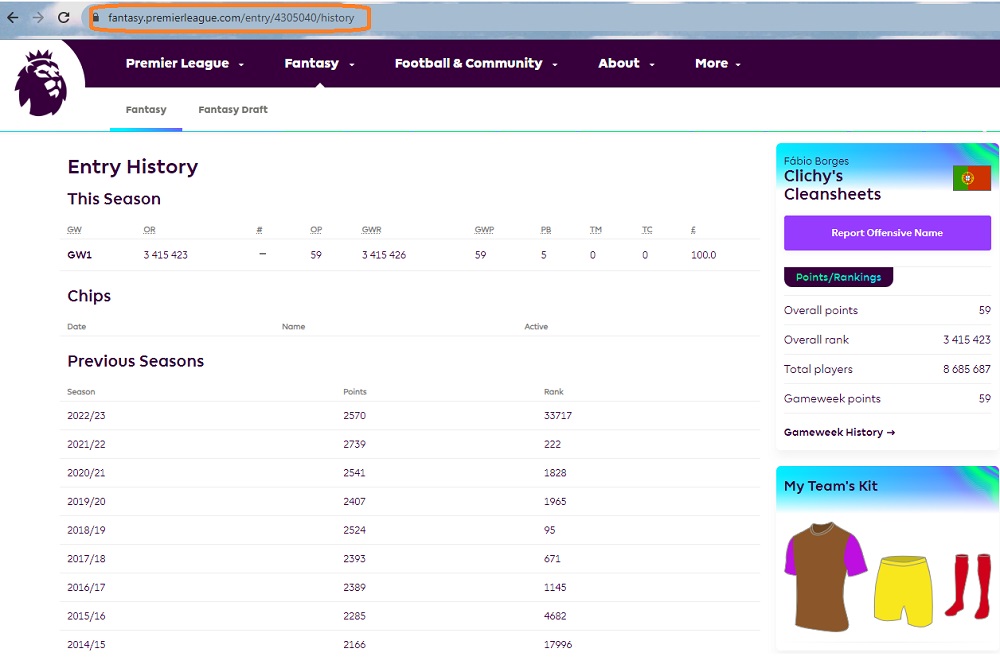


Рис. 3. Аккаунт #1 рейтинга livefpl Fábio Borges на сайте [fpl](https://fantasy.premierleague.com/entry/4305040/history)

Всё дело в том, что номер аккаунта указан не в самой таблице, а в виде web-ссылки, приклеенной к имени игрока:

Изображение выглядит как текст, программное обеспечение, веб-страница, Значок на компьютере

Автоматически созданное описание

Рис. 4. Ссылка на аккаунт fpl есть в html-коде страницы

При просмотре html-кода страницы эту ссылку можно найти (см. правую часть рис. 4). Как добраться до этой ссылки? Задал вопрос на форуме [planetaexcel](https://www.planetaexcel.ru/forum/index.php?PAGE_NAME=read&FID=1&TID=159325&TITLE_SEO=159325-zapros-v-pq-obrabatyvaet-tablitsu-iz-ineta_-no-teryaet-ssylku_-svyazannuyu-s-odnoy-iz-yacheek-tablitsy.-est-li-sposob-izvlech-etu-ssylku). Но ответа не получил.

#### Решение с помощью интерфейса редактора Power Query

Решение нашлось в [статье](https://datachant.com/2017/05/08/web-scraping-power-bi-part-2/) эксперта в области Power Query Гила Равива. На русском языке вышла книга Гила [Power Query в Excel и Power BI: сбор, объединение и преобразование данных](https://baguzin.ru/wp/?p=24451).

При импорте рейтинга с сайта livefpl Power Query по умолчанию использует функцию Web.Page(Web.Contents("https://www.livefpl.net/elite"))

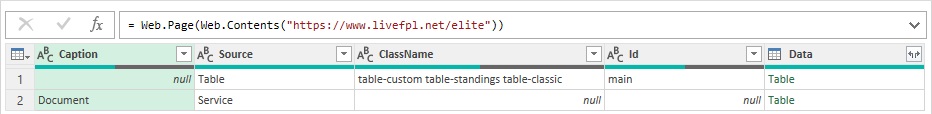


Рис. 5. Первый шаг запроса Power Query

Вот что сказано в [справке](https://learn.microsoft.com/ru-ru/powerquery-m/web-page) Microsoft о функции…

Web.Page(html as any) as table

… Возвращает содержимое документа HTML, разбитого на составные структуры, а также представление полного документа и его текста ***после удаления тегов***.

А нам то нужны теги! Поэтому Гил предлагает убрать функцию Web.Page и оставить Web.Contents().

Эта [функция](https://learn.microsoft.com/ru-ru/powerquery-m/web-contents)…

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

… возвращает содержимое, скачанное с адреса url ***в двоичном виде***.

Однако в редакторе PQ в строке кода не получится заменить…

Web.Page(Web.Contents("https://www.livefpl.net/elite"))

… на…

Web.Contents("https://www.livefpl.net/elite")

Редактор автоматически вернет первоначальное значение. Не беда. Откройте расширенный редактор, и выполните замену:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 6. Удаление функции Web.Page() в расширенном редакторе

Нажмите *Готово*. Вы увидите, что редактор PQ вернул двоичный файл:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис. 7. Функция Web.Contents() возвращает двоичный файл

Щелкните правой кнопкой мыши на файле и выберите *Текст*:

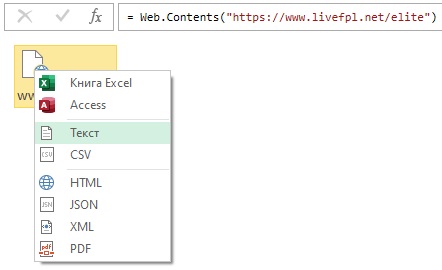


Рис. 8. Извлечение текста из двоичного файла

Эта команда интерфейса в редакторе PQ создаст строку кода…

= Table.FromColumns({Lines.FromBinary(Web.Contents("https://www.livefpl.net/elite"), null, null, 65001)})

Здесь [функция](https://learn.microsoft.com/ru-ru/powerquery-m/lines-frombinary) Lines.FromBinary() преобразует двоичное значение в список текстовых значений, разделенных разрывами строк:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 9. Содержимое страницы livefpl.net/elite в виде html-кода

С помощью фильтра оставляем строки, содержащие текст *history*:

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 10. Отфильтрованные строки

И, наконец, с помощью инструмента *Столбец из примеров* вкладки *Добавление столбца* сначала в отдельный столбец выделяем номер аккаунта, а затем имя игрока:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рис. 11. Выделение номера аккаунта с помощью инструмента *Столбец из примеров*

Вуаля!

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рис. 12. Результат извлечения номеров аккаунтов из html-кода

#### Решение с использованием кода на языке М

В своих попытках найти решение я обратился в частной переписке к [Михаилу Музыкину](https://muzykin.com/), эксперту по функциям языка М. Михаил предложил элегантное решение, которое позволяет в одном запросе получить и рейтинг и ссылки на аккаунты:

let

from = Binary.Buffer(Web.Contents("https://www.livefpl.net/elite")),

table = Web.Page(from){0}[Data],

links = List.Transform(

List.Select(

Text.Split(

Text.FromBinary(from),

"<a href="""

),

(x)=>Text.Contains(x,"https://fantasy.premierleague.com")

),

(x)=>Text.Split(x,""""){0}

),

to=Table.FromColumns(Table.ToColumns(table)&{links})

in

to

Несколько слов о том, как работает код. Точнее, как я это понимаю))

Web.Contents() извлекает бинарное содержимое страницы https://www.livefpl.net/elite.

Binary.Buffer() – помещает это содержимое в буфер. Во-первых, это ускоряет обработку, так как делается один запрос к странице https://www.livefpl.net/elite, а не два. Во-вторых, за время между обращениями за данными и web-ссылками содержимое страницы в Инете может измениться.

Web.Page(from) – возвращает содержимое страницы, разбитое на составные структуры в виде таблицы. Одна строка таблицы – одна структура. Как было сказано выше, теги удалены.

Web.Page(from){0} – возвращает первую строку таблицы.

Web.Page(from){0}[Data] – возвращает столбец *Data* первой строки таблицы, фактически ячейку. Поскольку в этой ячейке находится таблица, она возвращается в раскрытом виде.

Следующий фрагмент кода будем раскручивать из глубины.

Text.FromBinary(from) – возвращает текст из бинарного буфера, фактически html-код.

Text.Split(Text.FromBinary(from),"<a href=""") – разделяет html-код на элементы списка по разделителю

<a href="

List.Select() – фильтрует список, оставляя только элементы, удовлетворяющие условию:

(x)=>Text.Contains(x,"https://fantasy.premierleague.com")

Суть этого условия: оставить элементы, которые содержат текст:

https://fantasy.premierleague.com

List.Transform(list as list, transform as function) as list – в общем случае возвращает новый список, применяя функцию преобразования *transform* к списку *list*. В нашем случае в качестве функции используется

(x)=>Text.Split(x,"""")

Text.Split() разбивает каждый элемент списка, используя в качестве разделителя кавычки ". Почему используется четыре кавычки? Первые и четвертые определяют, текстовую строку и внутри этих кавычек находится сам разделитель. Третьи кавычки – собственно разделитель. Вторые кавычки экранируют разделитель, поскольку оказалось, что он представлен специальным символом.

Функция…

(x)=>Text.Split(x,""""){0}

… возвращает первый элемент списка.

Последний фрагмент сначала разбивает таблицу *table* на столбцы, затем добавляет столбец *links*, а затем объединяет все столбцы в таблицу *to*.