Парсинг html-кода в Power Query

В связи с увлечением Fantasy Premier League меня заинтересовал рейтинг игроков на сайте livefpl.

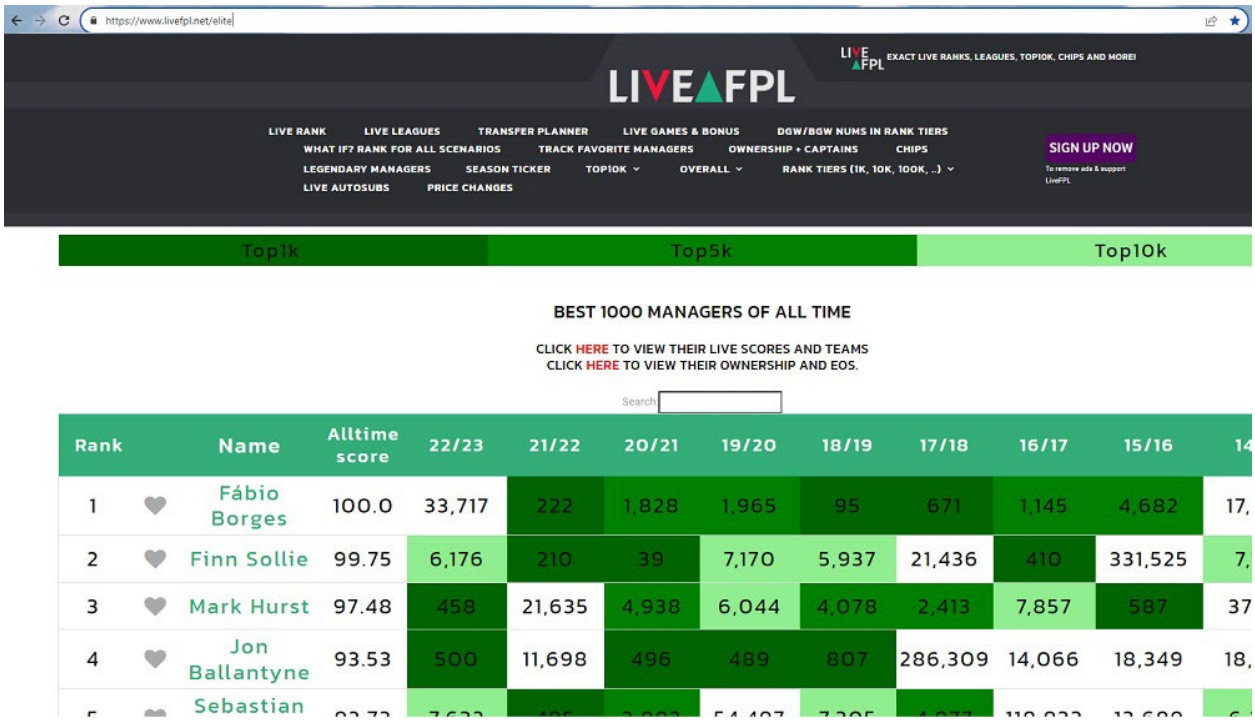

Рис. 1. Web-страница с рейтингом игроков

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:

	A	B	C	D	E	F	G	H	I	J
1	Rank	Name	Alltime score	22/23	21/22	20/21	19/20	18/19	17/18	16/17
2	1	Fábio Borges	100,00	33 717	222	1 828	1 965	95	671	1 14
3	2	Finn Sollie	99,75	6 176	210	39	7 170	5 937	21 436	41
4	3	Mark Hurst	97,48	458	21 635	4 938	6 044	4 078	2 413	7 85
5	4	Jon Ballantyne	93,53	500	11 698	496	489	807	286 309	14 06
6	5	Sebastian Jönsson	92,73	7 632	495	3 893	54 497	7 305	4 977	118 93
7	6	Calm _	90,67	456	186	292 969	60 291	1 471	4 796	73 91
8	7	Jørgen Stenseth	90,55	6 974	5 119	1 907	12 193	532	974	39 08
9	8	Mario Sulenta	90,34	348 615	726	4 005	95 073	1 363	4 925	6 35
10	9	Mark Lynch	89,48	379	1 050	493 797	9 179	8 429	2 942	37 04
11	10	Rob Mayes	89,11	6 769	82	782	3 685	29 530	16 937	18 92
12	11	Ben Alexander	89,06	3 906	67 802	222	206 690	10 363	2 751	2 01
13	12	Sean Connors	88,39	17 199	278	48 113	4 575	9 262	26 883	56
14	13	James Cooper	88,31	6 420	22 667	7 619	2 997	1 000	4 404	81 77
15	14	Stephen Devlin	88,08	34	52 120	3 004	6 171	1 383	99 820	31 05

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.

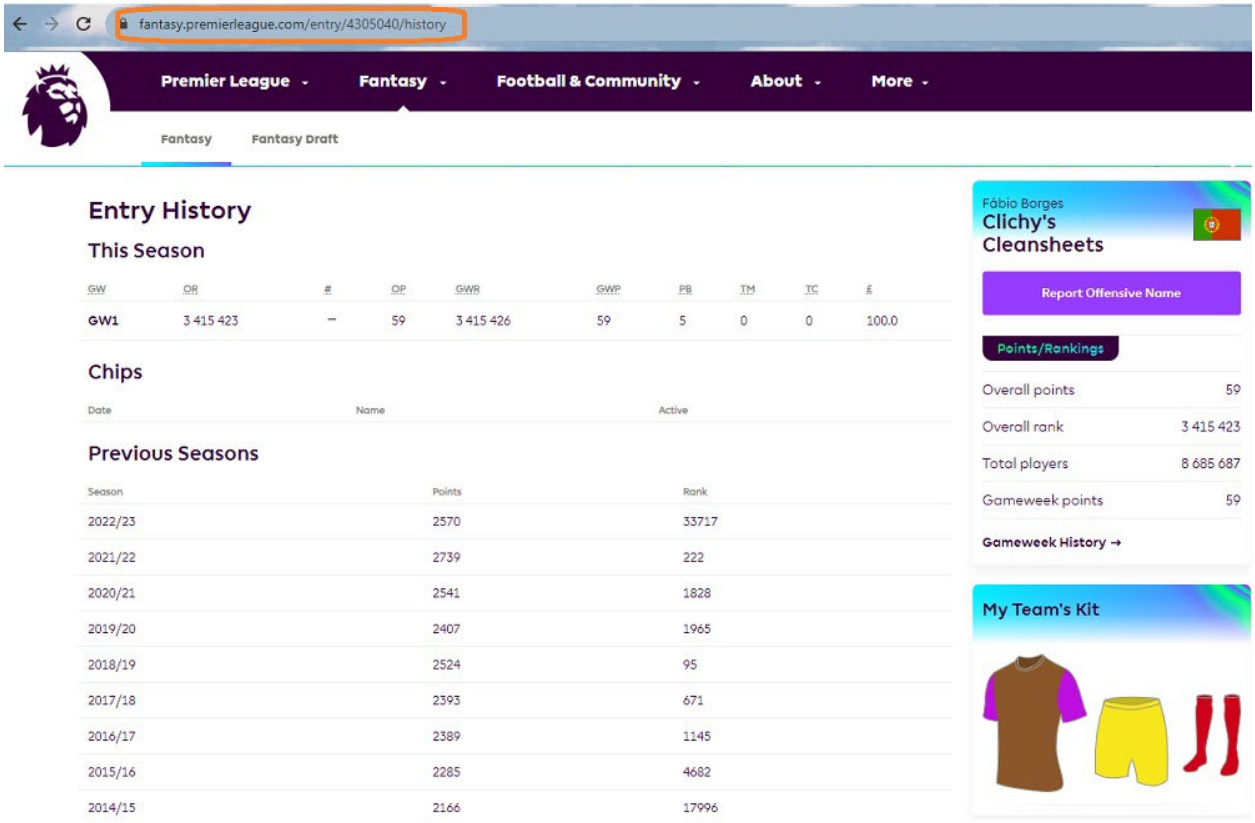

Рис. 3. Аккаунт #1 рейтинга livefpl Fábio Borges на сайте fpl

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

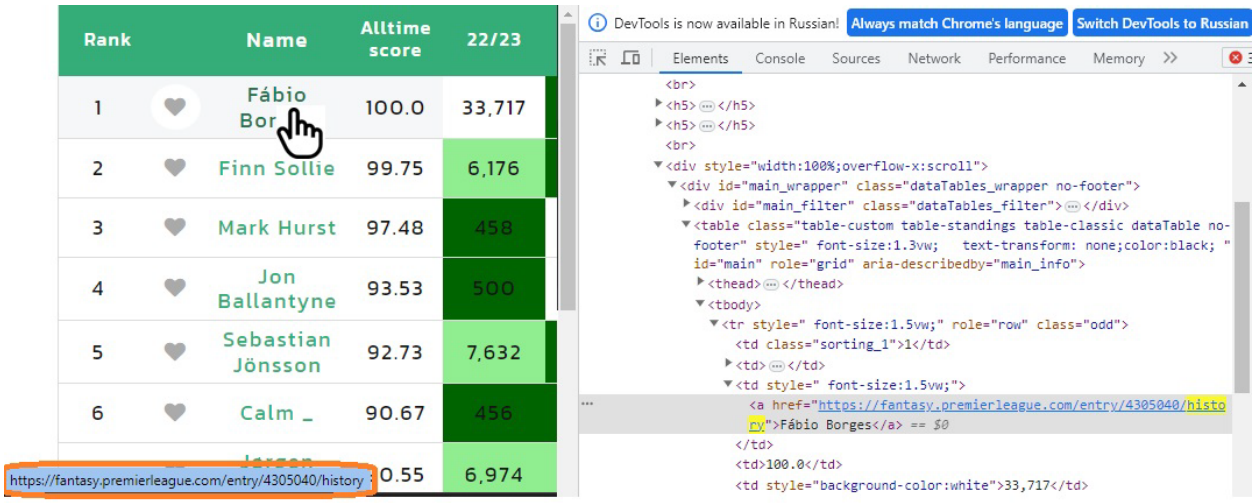

Рис. 4. Ссылка на аккаунт fpl есть в html-коде страницы

При просмотре html-кода страницы эту ссылку можно найти (см. правую часть рис. 4). Как добраться до этой ссылки? Задал вопрос на форуме planetaexcel. Но ответа не получил.

Решение с помощью интерфейса редактора Power Query

Решение нашлось в статье 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.

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

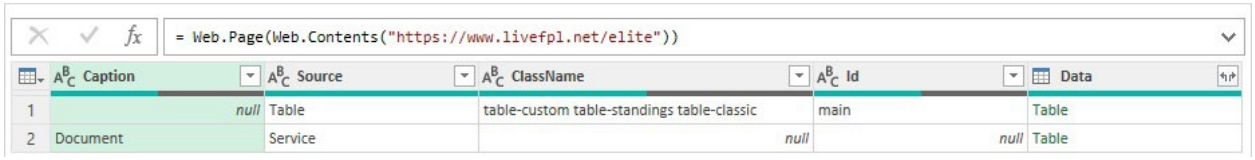

Рис. 5. Первый шаг запроса Power Query

Вот что сказано в справке 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…

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:

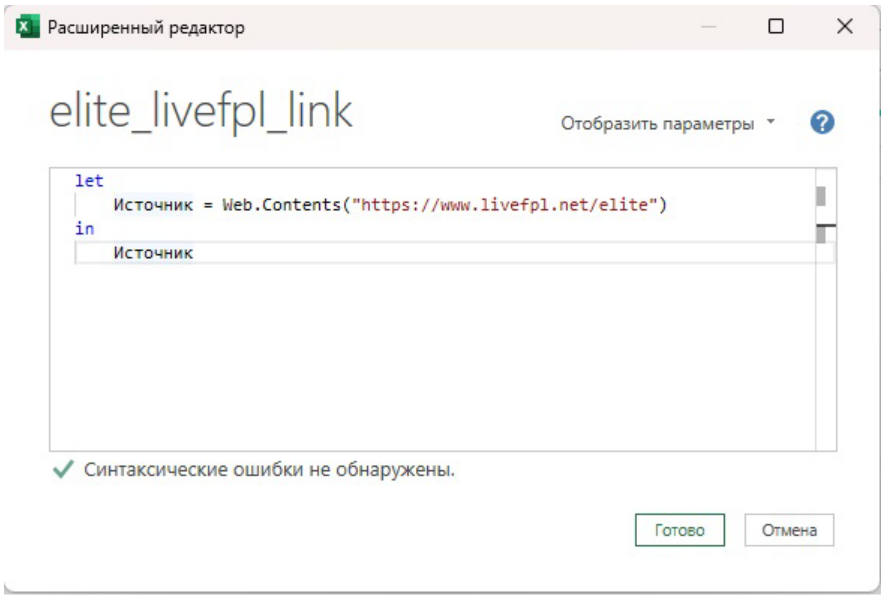

Рис. 6. Удаление функции Web.Page() в расширенном редакторе

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

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

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

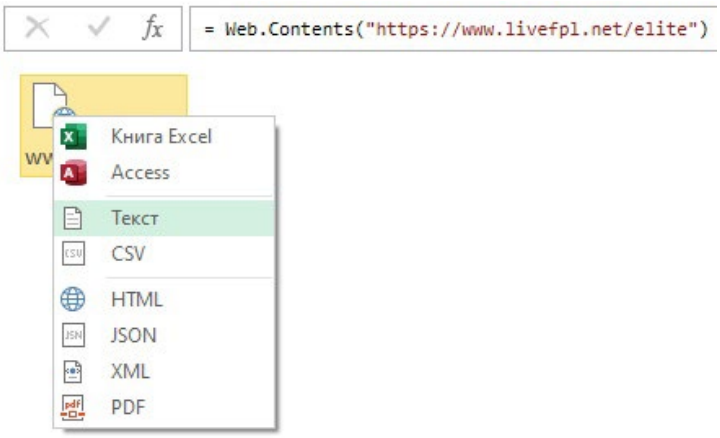

Рис. 8. Извлечение текста из двоичного файла

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 Lines.FromBinary() преобразует двоичное значение в список текстовых значений, разделенных разрывами строк:

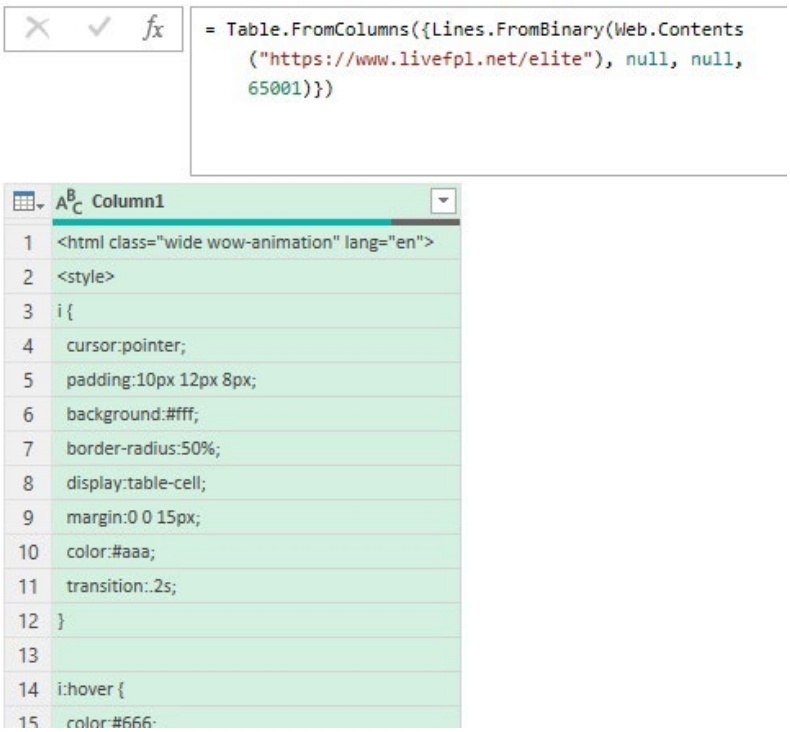

Рис. 9. Содержимое страницы livefpl.net/elite в виде html-кода

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

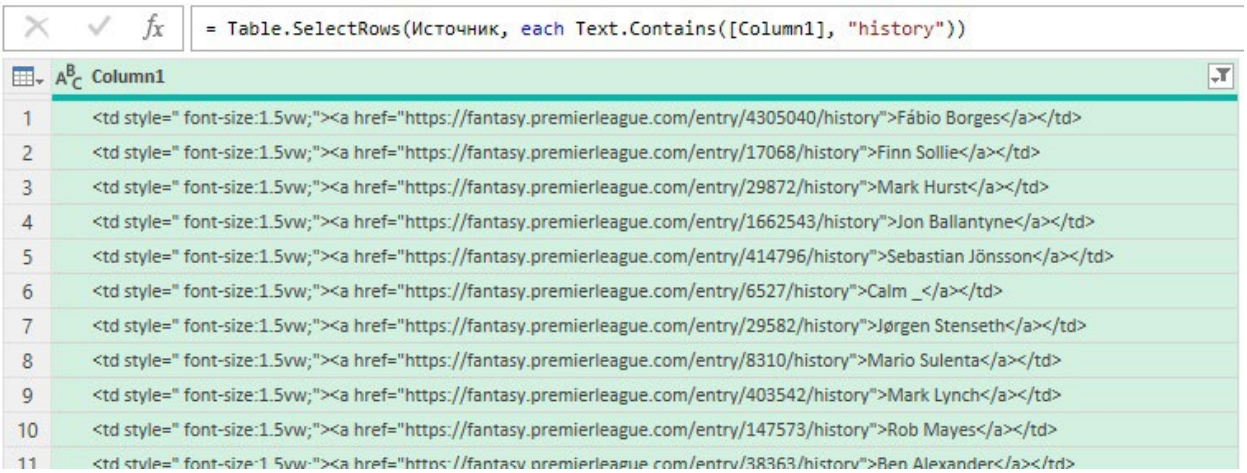

Рис. 10. Отфильтрованные строки

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

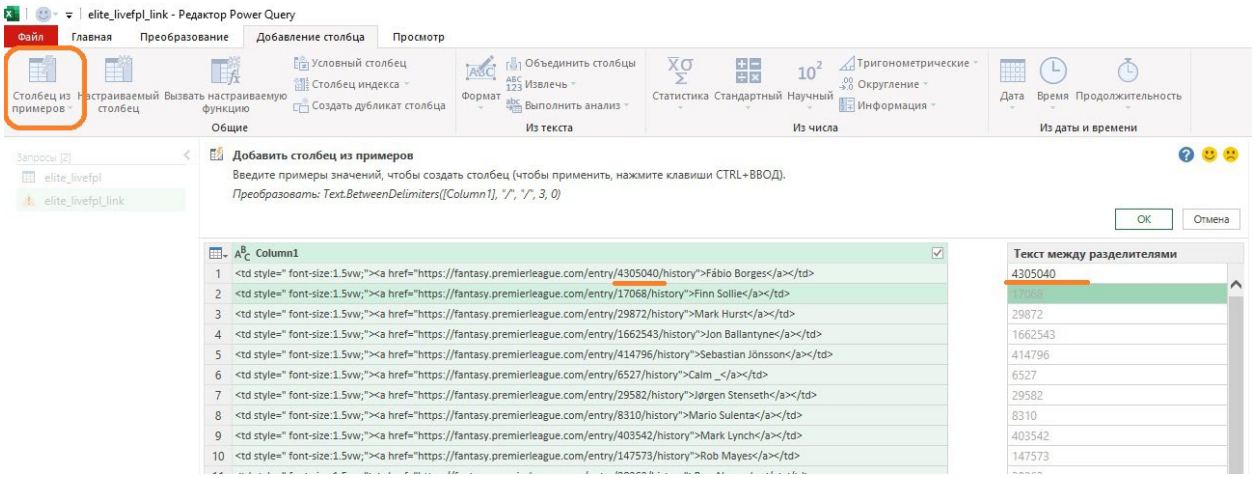

Рис. 11. Выделение номера аккаунта с помощью инструмента *Столбец из примеров*

Вуаля!

	A	B
1	entry	name
2	4 305 040	Fábio Borges
3	17 068	Finn Sollie
4	29 872	Mark Hurst
5	1 662 543	Jon Ballantyne
6	414 796	Sebastian Jönsson
7	6 527	Calm _
8	29 582	Jørgen Stenseth
9	8 310	Mario Sulenta
10	403 542	Mark Lynch
11	147 573	Rob Mayes
12	38 363	Ben Alexander
13	168 456	Sean Connors
14	2 863	James Cooper

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

Решение с использованием кода на языке M

В своих попытках найти решение я обратился в частной переписке к Михаилу Музыкину, эксперту по функциям языка M. 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*.