

# Обработка данных с reddit

Обработка данных с reddit  
Задание 1 - Сортировка статей  
Задание 2 - Агрегация статей  
Примечание к заданиям 1 и 2  
Задание 3 (дополнительно)  
Технологии

## Задание 1 - Сортировка статей

На входе:

1) json вида

```
{
  "kind": "Listing",
  "data": {
    "modhash": "",
    "children": [{
      "kind": "t3",
      "data": {
        "domain": "rathercurio.us",
        "banned_by": null,
        "media_embed": {},
        "subreddit": "javascript",
        "selftext_html": null,
        "selftext": "",
        "likes": null,
        "secure_media": null,
        "saved": false,
        "id": "1n4m6p",
        "secure_media_embed": {},
        "clicked": false,
        "stickied": false,
        "author": "masterJ",
        "media": null,
        "score": 16,
        "approved_by": null,
        "over_18": false,
        "hidden": false,
        "thumbnail": "",
        "subreddit_id": "t5_2qh30",
        "edited": false,
        "link_flair_css_class": null,
        "author_flair_css_class": null,
        "downs": 2,
        "is_self": false,
        "permalink": "/r/javascript/comments/1n4m6p/functional_mixins_and_advice_in_javascript/",
        "name": "t3_1n4m6p",
        "created": 1380169452.0,
        "url": "http://rathercurio.us/functional-mixins-and-advice-in-javascript",
        "author_flair_text": null,
        "title": "Functional Mixins and Advice in JavaScript",
        "created_utc": 1380140652.0,
        "link_flair_text": null,
        "ups": 18,
        "num_comments": 8,
        "num_reports": null,
        "distinguished": null
      }
    }, {
      "kind": "t3",
      "data": {
        "domain": "bunselmeyer.net",
        "banned_by": null,
        "media_embed": {},
        "subreddit": "javascript",
        "selftext_html": null,
        "selftext": "",
        "likes": null,
        "secure_media": null,
        "saved": false,
        "id": "1n46by",
        "secure_media_embed": {},
        "clicked": false,
        "stickied": false,
        "author": "tpk1024",
        "media": null,
        "score": 14,
        "approved_by": null,
        "over_18": false,
        "hidden": false,
        "thumbnail": "",
        "subreddit_id": "t5_2qh30",
        "edited": false,
        "link_flair_css_class": null,
        "author_flair_css_class": null,
        "downs": 0,
        "is_self": false,
        "permalink": "/r/javascript/comments/1n46by/template_inheritance_for_angular_js/",
        "name": "t3_1n46by",
        "created": 1380158376.0,
        "url": "http://www.bunselmeyer.net/#!/thoughts/angular-blocks",
        "author_flair_text": null,
        "title": "Template inheritance for Angular JS",
        "created_utc": 1380129576.0,
        "link_flair_text": null,
        "ups": 14,
        "num_comments": 0,
        "num_reports": null,
        "distinguished": null
      }
    }
  ]
}
```

целиком он тут <http://www.reddit.com/r/javascript/json> (это отображение страницы <http://www.reddit.com/r/javascript>)

- 2) направление и поле (дата создания, баллы) для сортировки выходных данных
- 3) выходной формат (csv, sql) и доп. параметры для форматтера (разделитель для csv, имя таблицы и полей для sql и пр.)

на выходе отображаем следующие поля:

- id
- title
- utc creation date
- score

например для статей отсортированных по баллам (в порядке убывания) в формате csv должно получиться, что-то вроде:

```
"1n3jj3", "Must.js – An assertion library with BDD syntax (awesome.must.be.true())". Ships with many expressive matchers. Honors RFC 2119 by using MUST. Good stuff and well tested.", "25.09.2013 16:09:00", 26
"1n46by", "Template inheritance for Angular JS", "26.09.2013 05:09:00", 14
"1n3bpg", "Why Lavaca is the only sane HTML5 mobile development framework out there (and why Sencha Touch sucks)", "25.09.2013 12:09:00", 8
```

## Задание 2 - Агрегация статей

Имя на входе тот же json (что и в задании 1) и выходной формат мы должны получить информацию по количеству статей для каждого домена отсортированную по убыванию, т.е. поля:

- domain
- articles count (количество статей на домене)
- score summ (сумма баллов всех статей домена)

например в csv:

"github.com", 13, 123  
"rathercurio.us", 10, 55  
"bunselmeyer.net", 5, 40

## Примечание к заданиям 1 и 2

Оба задания стоит рассматривать как часть целого - т.е. нужно организовать расширяемую систему для обработки json данных с reddit.com.  
Нужно предусмотреть (но не делать):

- изменение полей по которым осуществляется текущая обработка данных (сортировка и агрегация)
- возможность расширения набора стратегий обработки json'a (сейчас это сортировка и агрегация)
- расширение набора выходных форматов

Вариант реализации: веб-приложение, которому можно скормить url с reddit.com, указать операцию (сортировка, агрегация), формат + доп. параметры (например разделитель для csv) в результате получить данные в желаемом формате.

## Задание 3 (дополнительно)

Дан массив объектов с полями id (уникальный идентификатор) и parentId (идентификатор родителя).  
Для входных данных верно следующее: индекс ребенка всегда больше индекса родителя в исходном массиве.

Задача: превратить массив "плоских" объектов в массив объектов с вложенными детьми (поле children).

Исходные данные:

```
[
  {id: 1, parentId: 0},
  {id: 2, parentId: 0},
  {id: 3, parentId: 1},
  {id: 4, parentId: 1},
  {id: 5, parentId: 2},
  {id: 6, parentId: 4},
  {id: 7, parentId: 5}
];
```

должны превратиться в:

```
[
  {
    "id": 1,
    "parentId": 0,
    "children": [
      {
        "id": 3,
        "parentId": 1
      },
      {
        "id": 4,
        "parentId": 1,
        "children": [
          {
            "id": 6,
            "parentId": 4
          }
        ]
      }
    ]
  },
  {
    "id": 2,
    "parentId": 0,
    "children": [
      {
        "id": 5,
        "parentId": 2,
        "children": [
          {
            "id": 7,
            "parentId": 5
          }
        ]
      }
    ]
  }
]
```

## Технологии

[Node.js](#) поддерживаемой **LTS** версии (4.x, 6.x и т.д.).

Предпочтительные npm пакеты:

- express 4.x
- twostep
- jade/pug
- conform
- underscore
- mongodb
- mongodbext

Code style - <https://github.com/2do2go/code-style/blob/master/js.md>