**Cloudspokes REST Leaderboard service**Daniel Llewellyn (Kenji776)

**Abstract:**

A simple interface to generate a JSON payload containing the leaders for an optional given technology for a given date range. It supports pagination with custom page sizes as well. Results are returned in order so all you have to do is iterate over the response to build your scoreboard. Also additional information is included to help you build a more robust app.

**Arguments:**

The class is capable of taking several filtering arguments that can alter the returned data to match your needs. The follow parameters may be passed as either GET or POST data to the class. If the parameter is not passed, the specified default value is used.

**technology** (string) (default=null): limit learderboard results to a given technology type

**maxrows** (integer) (default = 10): maximum number of leaderboard results to query for

**pagesize** (integer) (default = 10): number of leaderboard results on a page

**pagenum** (integer) (default = 1): the number of the page of data to get

**challenge\_date** (date literal keyword): a date keyword that will only find challenges matching that keyword.

For more information on salesforce date literals visit <http://www.salesforce.com/us/developer/docs/api/Content/sforce_api_calls_soql_select_dateformats.htm>

**Sample Return Data**

The class returns data in a consistent reliable JSON structure format. It will echo back to you all the parameters you passed it, so you can verify they were passed as expected. It contains a simple Boolean success flag that will inform you if your request completed successfully. There is a message variable that will contain further details if an error is encountered. The data attribute is an array of people, in order of their total winnings. Each array element contains basic facts about the winner such as name and email, and within that the participations key contains all the participation information that contributed to this leaderboard entry.

{

"total\_results": "1",

"success": "true",

"page\_start\_rank": "1",

"page\_end\_rank": "1",

"page\_number": "1",

"technology": null,

"total\_pages\_available": "1",

"data": “[

{

"total\_money": "15000.00",

"rank": 1,

"profile\_pic\_\_c": "null",

"email\_\_c": "kenji776@gmail.com",

"first\_name\_\_c": "Daniel",

"last\_name\_\_c": "Llewellyn",

"username\_\_c": "Kenji776",

"total\_points": "2500.0",

"participations": [

{

"contest\_id": "a0AU0000000XbZ9MAK",

"contest\_name": "TakeOverTheWorld",

"place\_\_c": "1",

"score\_\_c": "300",

"money\_awarded\_\_c": "10000.00",

"points\_awarded\_\_c": "1500",

"place": "1",

"participation\_id": "a04U0000000THkOIAW"

},

{

"contest\_id": "a0AU00000006MbaMAE",

"contest\_name": "BeAwesome",

"place\_\_c": "1",

"score\_\_c": "200",

"money\_awarded\_\_c": "5000.00",

"points\_awarded\_\_c": "1000",

"place": "1",

"participation\_id": "a04U0000000SQ0UIAW"

}

],

"member": "a0CU00000001Cu0MAE"

},

]”,

"date\_keyword": null,

"page\_size": "2",

"results\_on\_page": "2",

"message": "Query Ran"

}

**Example Calls**

Below are some example calls you can make that showcase the easy of data filtering to match your needs.

<https://na12.salesforce.com/services/apexrest/v.9/leaderboard>

-Get the overall winners for all time (top 10 based on defaults).

<https://na12.salesforce.com/services/apexrest/v.9/leaderboard?dateRange=last_year>

-Get the overall winners for last year

<https://na12.salesforce.com/services/apexrest/v.9/leaderboard?dateRange=this_year&technology=javascript>

-Get the javascript category winners for this year

<https://na12.salesforce.com/services/apexrest/v.9/leaderboard?dateRange=this_month&technology=salesforce&maxrows=5>

-Get the top 5 salesforce winners for this month

<https://na12.salesforce.com/services/apexrest/v.9/leaderboard?pagesize=5&pagenum=2>

-Get the second page of winners for all time. Only display 5 people per page.

**Gotchas**

Due to the way the return data structure had to be constructed (an array of strings) the data array within the main JSON return is treated as string instead of a JSON array. What this means is that when Salesforce returns this array it comes wrapped in a set of quotes. Since it is actually a JSON array, and not a string, this causes it to become invalid JSON. To make it valid once you have received the data payload simply strip the quotes from around the data key value. You can also pass the contents of the data key into a javascript eval statement and it will return a JSON array. So while the extra quotes may cause problems for parsers, simply removing them should fix it.

I recognize that this is kind of a lame bug, but Salesforce is extremely restrictive in the data types you can return and since I wanted a constant, predictable return format this was the sacrifice I had to make. If I tried to return a simple sObject there wouldn’t be a good way to report errors, etc.