

## Task: Bidding

Implement a simple web service with two endpoints.

Implement this in either Java, JavaScript, or Python. For this task it's enough to store the configuration in memory. Don't worry about persistence or anything complicated.

### `/configure`

This endpoint is used to configure the service. This endpoint should accept JSON via POST. The JSON will have the following structure:

```
[
  {
    "id": <string>,
    "rules": [
      {
        "key": <string>, "value": <string>,
        "behavior": "boost" or "match"
      },
      ...
    ]
  },
  ...
]
```

This represents a list of campaigns, where each member of the array is a campaign. Each campaign has an `id`, and an arbitrary number of rules. Each rule has a `key`, a `value` to match, and a `behavior`, which can be either `"boost"` or `"match"`.

### `/bid`

This endpoint is used to request bids. This endpoint should accept GET requests with an arbitrary number of query parameters.

Find all campaigns that match this request, and return as JSON their `ids`, and the number of `boost` rules that matched. The returned JSON should look like this:

```
[
  {
    "id": <string>,
    "boosted": <integer>
  },
  ...
]
```

For every campaign, evaluate all of the rules. For a rule that has `behavior` set to `"match"`, check if there's a query parameter named `key`, with the value of `value`. If all of those rules match, evaluate all of the boost rules similarly.

For every campaign that had all rules match, calculate the number of boost rules that matched.

## Test Cases

Rules match like this:

```
$ /configure [  
  {  
    "id": "campaign1",  
    "rules": [  
      {"key": "a", "value": "a1", "behavior": "match"},  
      {"key": "b", "value": "b1", "behavior": "boost"}  
    ]  
  },  
  {  
    "id": "campaign2",  
    "rules": [  
      {"key": "c", "value": "c1", "behavior": "match"}  
    ]  
  }  
]
```

```
$ /bid  
[]
```

```
$ /bid?a=a1  
[{"id": "campaign1", "boosted": 0}]
```

```
$ /bid?a=a2  
[]
```

```
$ /bid?a=a1&b=b1  
[{"id": "campaign1", "boosted": 1}]
```

```
$ /bid?c=c1&b=b1  
[{"id": "campaign2", "boosted": 0}]
```

```
$ /bid?a=a1&c=c1  
[  
  {"id": "campaign1", "boosted": 0},  
  {"id": "campaign2", "boosted": 0}  
]
```

Empty configuration leads to empty bid results:

```
$ /configure []
```

```
$ /bid  
[]
```

```
$ /bid?a=a1  
[]
```

Empty rules match for all requests:

```
$ /configure [  
  {
```

```
    "id": "campaign1",  
    "rules": []  
  }  
]
```

```
$ /bid  
[{"id": "campaign1", "boosted": 0}]
```

```
$ /bid?a=a1  
[{"id": "campaign1", "boosted": 0}]
```