Introduction

With Codeforces API you can get access to some of our data in machine-readable JSON format.

To access the data you just send a HTTP-request to address `http://codeforces.com/api/{methodName}` with method-specific parameters. Each method description has an example URL.

Each method call returns a JSON-object with three possible fields: status, comment and result.

- Status is either "OK" or "FAILED".
- If status is "FAILED" then comment contains the reason why the request failed. If status is "OK", then there is no comment.
- If status is "OK" then result contains method-dependent JSON-element which will be described for each method separately. If status is "FAILED", then there is no result.

API may be requested at most 5 times in one second. If you send more requests, you will receive a response with "FAILED" status and "Call limit exceeded" comment.

Language-depended fields like names or descriptions will be returned in the default language. Also, you can pass additional parameter `lang` with values `en` and `ru` to select the language of the result.

Authorization

All methods can be requested anonymously. This way only public data will be accessable via API. To access data, private for some user (e.g. hacks during the contest), API key must be generated on http://codeforces.com/settings/api page. Each API key has two parameters: `key` and `secret`. To use the key you must add following parameters to your request.

1. `apiKey` — it must be equal to `key`
2. `time` — current time in unix format (e.g., System.currentTimeMillis()/1000). If the difference between server time and time, specified in parameter, will be greater than 5 minutes, request will be denied.
3. `apiSig` — signature to ensure that you know both `key` and `secret`. First six characters of the `apiSig` parameter can be arbitrary. We recommend to choose them at random for each request. Let's denote them as `rand`. The rest of the parameter is hexadecimal representation of SHA-512 hash-code of the following string: `<rand>/<methodName>?param1=value1&param2=value2...&paramN=valueN#<secret>` where `(param_1, value_` are all the request parameters (including `apiKey`, `time`, but excluding `apiSig`) with corresponding values, sorted lexicographically first by `param_i`, then by `value_i`.

For example:

If your `key` is `xxx`, `secret` is `yyy`, chosen `rand` is `123456` and you want to access method `contest.hacks` for contest 374, you should compose request like this: `http://codeforces.com/api/contest.hacks?contestId=374&apiKey=xxx&time=1426636798&apiSig=123456<hash>`, where `<hash>` is `sha512Hex(123456/contest.hacks?apiKey=xxx&contestId=374&time=1426636798#yyy)`

JSONP

JSONP is supported. Add `jsonp` parameter to the query and the result will be returned as JavaScript function call.

For example, if `jsonp` parameter is `parseResponse` and method returned object `{"status":"OK","response":"..."}`, then final result is `parseResponse({"status":"OK","response":"..."});`.