

# AutoAnalyst API

From ICPC-Contest Control Standard

Jump to navigationJump to search



## Contents

- 1 Augmented event stream (icat/api/EventFeed)
- 2 Query interfaces
  - 2.1 HTTP GET /icat/api/scoreboard
  - 2.2 HTTP GET /icat/api/TeamNotifications/<teamid>
  - 2.3 HTTP GET /icat/api/EditActivity?from=FROM&to=TO
  - 2.4 HTTP GET /icat/api/LastEditActivity?min=MIN
  - 2.5 HTTP GET /icat/api/CodeActivity?team\_id=<tid1,tid2,...>&problem\_id=<pid1,pid2,...>&granularity=MIN
  - 2.6 HTTP GET /icat/api/LastEditedNotSolvedProblem?team\_id=<teamid>

## Augmented event stream (icat/api/EventFeed)

The stream produced by the AutoAnalyst contains all tags produced by Kattis, but adds analystmsg tags to the stream. The stream will initially be provided as a file that is appended to as the competition proceeds, that can be read by other processes. Other means of interfacing can also be provided, for instance a TCP connection similar to the one provided by the Contest Control System.

Examples:

```
┌-----┐
|<analystmsg>
|<id>24</id>
|<team>39</team>
|67
|<priority>0</priority>
|<problem>7</problem>
|<run_id>122</run_id>
|<submission>122</submission>
|<category>auto</category>
|<message>St. Petersburg State University of IT, Mechanics and Optics now leads the competition after
|solving G - Minimum Cost Flow. It has now solved 3 problems.</message>
|</analystmsg>
└-----┘
```

```
┌-----┐
|<analystmsg>
|<id>15</id>
|30
|<priority>1</priority>
|<submission>-1</submission>
└-----┘
```

```
<category>human</category>
<message>K and D are promising to be the most popular problems on the first hour. Many teams are
working on them at the moment.</message>
</analystmsg>
```

id: Sequential identifier of this message. All notifications get a unique and strictly increasing id.  
team: Identifier of the team that triggered the event (typically by submitting a solution)  
time: Time in contest when this event occurred  
priority: Number indicating the priority of the message where 0 is the highest priority. For WF 2015 we will have:

```
0 All very important messages
1 Default priority human analyst messages
2 High priority katalyzer messages
3 Default priority katalyzer messages
```

problem: Problem that the message is relevant to  
run\_id: Identifier of the submission that the message is related to, if any  
submission: Identifier of the submission that the message is related to, if any. If no submission then the value is -1.  
category: String describing the category of message such as 'human' or 'auto' depending on who produced the message  
message: Message from analyzer, in plain text

## Query interfaces

These interfaces provide current information through http access. Content is sent using gzip compression, and the Content-Encoding http header will therefore always be set to gzip. Repetitive json documents compress very well, and all modern browsers handle the decompression transparently.

### HTTP GET /icat/api/scoreboard

Returns a JSON-document describing the current state of the scoreboard as specified in the Contest API scoreboard endpoint. Moreover, it contains additional data on what the teams are working on and what rank teams would get when the next problem is solved. It is intended to be queried by ajax directly from the browser, but can of course also be fetched by a server process.

The following additional elements are defined on top of the base specification:

- Each scoreboard row can contain an element **main\_lang**, a string identifying the main programming language used by this team.
- Each problem element in problems array in each scoreboard row can the following extra elements:
  - A string **label**, containing the label of the problem on the scoreboard, typically a single letter.
  - If **solved** is false, then an object **potential** containing an integer element **rank** describes to what rank the team would rise if it would solve this problem now; if additionally the integer element **before** is present, the problem must be solved within that many minutes from now to attain **rank**.
  - If the team has submissions for this problem, then the string **lang** describes the

language used for their last submission.

A sample team scoreboard row could look like this:

```
{
  "rank":7,
  "team_id":56,
  "score":{
    "num_solved":4,
    "total_time":302
  },
  "main_lang":"Java",
  "problems":[
    {"problem_id":10,"label":"A","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}},
    {"problem_id":11,"label":"B","solved":true, "num_judged":1,"time":20, "lang":"C++"},
    {"problem_id":12,"label":"C","solved":true, "num_judged":1,"time":104,"lang":"Java"},
    {"problem_id":13,"label":"D","solved":true, "num_judged":1,"time":66, "lang":"Java"},
    {"problem_id":14,"label":"E","solved":true, "num_judged":1,"time":112,"lang":"Java"},
    {"problem_id":15,"label":"F","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}},
    {"problem_id":16,"label":"G","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}},
    {"problem_id":17,"label":"H","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}},
    {"problem_id":18,"label":"I","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}},
    {"problem_id":19,"label":"J","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}},
    {"problem_id":20,"label":"K","solved":false,"num_judged":3,"time":0,"potential":{"rank":6,"lang":"Java"},
    {"problem_id":21,"label":"L","solved":false,"num_judged":0,"time":0,"potential":{"rank":4,"before":5}}
  ]
}
```

### HTTP GET /icat/api/TeamNotifications/<teamid>

Returns a json document containing all AutoAnalyst notifications relevant to this team, ordered chronologically. These are exactly the same notifications that have been published in the event stream, but made available in a query API and in json format.

```
{
  "team": {
    "id":56,
    "tag":"#t56",
    "name":"Shanghai Jiao Tong University"
  },
  "messages": [
    {
      "id": 79,
      "time": 20,
      "problem": 2,
      "message": "Team Shanghai Jiao Tong University submitted solution for B - Curvy Little Bottles. If correct, they will get rank 5 (7)"
    },
    {
      "id": 80,
      "time": 20,
      "problem": 2,
      "message": "Shanghai Jiao Tong University solves its first problem: B - Curvy Little Bottles"
    }
  ]
}
```

### HTTP GET /icat/api/EditActivity?from=FROM&to=TO

Returns a json document containing all edit activities detected in the open interval *FROM-TO*

(contest time in minutes) ordered chronologically in an array.

```
|EditActivity?from=80&to=85 could for example return
```

```
|[
| { "id": "213", "time": "80",
|   "team_id": "56", "problem_id": "A",
|   "language": "C++", "line_count": "49", "diff_line_count": "10", "file_size_bytes": "4277"
| },
| { "id": "214", "time": "84",
|   "team_id": "84", "problem_id": "B",
|   "language": "C++", "line_count": "23", "diff_line_count": "15", "file_size_bytes": "1547"
| }
| { "id": "215", "time": "85",
|   "team_id": "56", "problem_id": "A",
|   "language": "C++", "line_count": "55", "diff_line_count": "6", "file_size_bytes": "4444"
| }
| ]
```

### HTTP GET /icat/api/LastEditActivity?min=MIN

Returns a json document containing all edit activities detected in the last *MIN* minutes ordered chronologically in an array.

```
|LastEditActivity?min=5 could for example return
```

```
|[
| { "id": "213", "time": "80",
|   "team_id": "56", "problem_id": "A",
|   "language": "C++", "line_count": "49", "diff_line_count": "10", "file_size_bytes": "4277"
| },
| { "id": "214", "time": "84",
|   "team_id": "84", "problem_id": "B",
|   "language": "C++", "line_count": "23", "diff_line_count": "15", "file_size_bytes": "1547"
| }
| { "id": "215", "time": "85",
|   "team_id": "56", "problem_id": "A",
|   "language": "C++", "line_count": "55", "diff_line_count": "6", "file_size_bytes": "4444"
| }
| ]
```

### HTTP GET

**/icat/api/CodeActivity?team\_id=<tid1,tid2,...>&problem\_id=<pid1,pid2,...>&granularity=MIN**

Returns a JSON document with a summary of edit and submission activity, intended for generating a graph showing which problems the teams are spending their time on. A few minutes into the contest, a possible response could be

```
|{ "granularity": 5,
|  "problems": [
|    { "problem_id": "F",
|      "edit_summary": [
|        { "contest_time": 0, "num_edited_lines": 2 },
|        { "contest_time": 5, "num_edited_lines": 4 }
|      ],
|      "submissions": {
|        "AC": [
|          { "contest_time": 8, "team_id": 47, "lang_id": "C++", "submission_id": "6" },
|          { "contest_time": 9, "team_id": 71, "lang_id": "C++", "submission_id": "12" }
|        ]
|      }
|    }
|  ]
| }
```

```

|         ],
|         "WA": [
|           { "contest_time": 9, "team_id": 71, "lang_id": "C++", "submission_id": "7" }
|         ]
|       }
|     }
|   ]
| }

```

All arguments are optional. "granularity" is important to note when interpreting the edit summary, since it describes the length of the "bins" in the edit summary. In the example, there are two edits between minutes 0 and 5, and four edits between minutes 5 and 10. A granularity of 1 would have meant two edits between 0 and 1, and four edits between 5 and 6.

### HTTP GET /icat/api/LastEditedNotSolvedProblem?team\_id=<teamid>

Returns a JSON document with the problem a team worked on last, along with the contest time when the problem was last edited. The time of submission is not regarded as an edit in this context, and as soon as a submission has been accepted that problem is excluded altogether. This might mean that a problem that was edited prior to the last one might be returned, and hence it's important to take how "last\_edit\_time" compares to the current contest time into account when deciding on the relevance of the information.

```

{ "team_id": 29,
  "problem_id": [
    "E"
  ],
  "last_edit_time": 155
}

```

A `team_id` query parameter needs to be provided, and only one integer is accepted. If no *unsolved* problem has been edited yet, the "problem\_id" array is empty and "last\_edit\_time" set to null. In the case where more than one problem was edited during the same minute, all of these problems will be listed in alphabetical order in the "problem\_id" array.

Note that only submissions accepted up until scoreboard freeze will be removed from the result. For example, if a team (regardless of the contest time) last edited problem A, then this endpoint will return A as the last edited problem if and only if the team did not solve problem A before scoreboard freeze. After freeze, the endpoint will keep saying that they edited problem A until the teams starts to work on another problem that was not already solved before the freeze.

Retrieved from "[https://clics.ecs.baylor.edu/index.php?title=AutoAnalyst\\_API&oldid=2952](https://clics.ecs.baylor.edu/index.php?title=AutoAnalyst_API&oldid=2952)"

- 
- This page was last edited on 16 April 2018, at 05:29.
  - Content is available under Creative Commons Attribution-ShareAlike unless otherwise noted.