

CrabServer TaskTrackingDatabase

Objective

A new *API* has to be defined to handle task related information. Its proposed name is *TaskStateAPI* and it should follow as much as possible the same 'look and feel' of the *JobStateAPI*.

Methods in *TaskStateAPI*

The following methods should be implemented in order to deal with task creation and completion:

- `create(taskName)`: insert information about a new task. Called from *CrabServerWorker*.
- `createJob(taskName, jobName, cacheArea)`: insert information about a new job associated to a task. Called from *CrabServerWorker*.
- `finished(taskName)`: a task has finished. Called from *TaskTracking* when completed percentage is 100%
- `submitFailure(taskName)`: a task cannot be submitted by *BOSS*. Called from *CrabServerWorker* if submission fails.
- `failed(taskName)`: Called from *TaskTracking* when all jobs in the task have failed.

The following methods should be implemented in order to deal with task monitoring:

- `percentageDone(taskName)`: the percentage of jobs finished (successfully or not).
- `finishedJobs(taskName)`: the number of jobs that have finished.
- `failedJobs(taskName)`: the number of jobs that have failed.
- `numberOfJobs(taskName)`: the number of jobs in a task.

Database schema

Two tables has to be added: `js__taskInstance` that will contain one row for each tasks, and `js__task` that will contain one row for each job in a task. A proposed schema definition is as follows:

```
CREATE TABLE js_taskInstance (
  id int NOT NULL auto_increment,
  taskName varchar(255) NOT NULL default '',
  primary key(id),
  unique(taskName),
  key(taskName)
) TYPE = InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE js_task (
  id int NOT NULL auto_increment,
  taskId int NOT NULL default '0',
  jobSpecId varchar(255) NOT NULL default '',
  PRIMARY KEY(id),
  FOREIGN KEY(taskId) references js_taskInstance(id) ON DELETE CASCADE,
  FOREIGN KEY(jobSpecId) references js_JobInstance(JobSpecID) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

TaskTrackingDatabase < Main < TWiki

As an example, if the task 'test1' is defined in terms of three jobs 'job1', 'job2' and 'job3', and the task 'test2' with jobs 'otherjob1' and 'otherjob2', the representation can be as follows:

id	taskName
...	...
4	test1
5	test2
...	...

id	taskId	jobSpecId
...
7	4	job1
8	4	job2
9	4	job3
10	5	otherjob1
11	5	otherjob2
...

-- CarlosKavka - 17 Jan 2007

This topic: [Main > TaskTrackingDatabase](#)

Topic revision: [r1 - 2007-01-17](#) - [CarlosKavka](#)



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? [Send feedback](#)