

API грид-сервисов, создаваемых на основе технологий REST/JSON

Грид-сервис распределения многошаговых заданий

API RESTful грид-сервисов приводится на примере сервиса выполнения многошаговых заданий. Этот сервис, который мы назовем *pilot*, работает с заданиями, которые представляют собой объекты сложной структуры, состоящие из элементарных задач. Каждая элементарная задача является запуском какой-либо программы на выполнение на грид-ресурсе по протоколу Globus WS-GRAM. Интерфейс прикладного программирования этого сервиса должен предоставлять возможность создания ресурсов заданий и задач, изменения их свойств, выполнения операций над заданиями (запуск, отмена), получение информации о состоянии заданий и задач, а так же получения учетной информации о выполнении заданий и задач. Таким образом, основными ресурсами сервиса являются задания, задачи, и учетная информация.

Спецификация API

Систему предлагается реализовать в виде двух компонентов: системы управления выполнением заданий и системы регистрации и контроля выполнения задач. Основное назначение системы — выполнение заданий.

Задание — это направленный ациклический граф, узлами в котором являются задачи, а ребра задают порядок их выполнения. Задачи, из которых состоит задание, выполняются в порядке обхода графа. При этом в местах ветвления, если это возможно, разные ветки графа выполняются одновременно. Задачи для выполнения передаются в систему регистрации задач.

На данном этапе ребра графа задают только порядок выполнения задач. Однако в будущем, возможно, ребра будут определять так же передачу выходных данных одних задач на вход другим.

Задача является базовым элементом, который может быть выполнен системой регистрации задач. Запуском задач на выполнение, а так же слежением за их выполнением занимается система регистрации и контроля выполнения задач. Систему распределения заданий предлагается реализовать как веб-сервис, следующий концепции REST, с представлением документов в виде JSON-документов.

Далее будем считать, что элементу *pilot/* соответствует корневой URI всех сервисов, например, <https://pilot.grid-net.ru/>. Все данные передаются в формате JSON. Формальные описания форматов приводятся в виде JSON Schema

Ресурсы сервиса *pilot*:

- *pilot/jobs/* — все задания данного пользователя;
- *pilot/jobs/<jobid>/* — задание;

- `pilot/jobs/<jobid>/<taskid>/` — задача задания;

При отправке всех запросов для установления `https`-соединения необходимо использовать прокси-сертификат пользователя, со всеми необходимыми расширениями.

Все запросы, отправляемые на сервер, имеющие не пустое тело, должны содержать заголовок `Content-MD5`. Все ответы сервера, имеющие не пустое тело, всегда содержат заголовок `Content-MD5`.

Управление заданиями и задачами

Создание нового задания

Для создания новых заданий используется запрос `POST` к `pilot/jobs/`, запрос является объектом, имеющим следующие атрибуты:

- `definition`, объект - описание задания, и, возможно, задач, согласно спецификации описания заданий, задач и ресурсных требований.

Ответ на запрос соответствует ответу на запрос списка заданий. Возможные коды HTTP ответа: 201, 400, 401, 412.

Получение информации о задании

Для получения информации о состоянии задания используется запрос `GET` к `pilot/jobs/<jobid>/`, где `<jobid>` — идентификатор задания. Ответ на запрос является объектом, имеющим следующие атрибуты:

- `created`, `modified`, строка - дата/время создания задания и дата/время последнего изменения задания или его состояния.
- `expires`, строка - дата/время автоматического удаления задания с сервера.
- `server_time`, строка - дата/время на сервере на момент генерации данного ответа сервера.
- `server_policy_url`, строка - URL документа, описывающего политику сервера, применяемую к данному заданию; политика сервера описывает используемые сервером умолчания, и ограничения.
- `owner`, строка - DN владельца задания
- `vo`, строка - виртуальная организация, от имени которой запущено задание. `null` если неизвестно.
- `state`, список объектов - история изменений состояния задания. Каждый объект имеет атрибуты:
- `s`, строка - состояние задания; возможные значения:
 - `new`: только что созданное задание;
 - `pending`: задание ожидает начала выполнения;
 - `running`: задание выполняется;
 - `paused`: выполнение задания приостановлено;
 - `finished`: выполнение задания завершено;
 - `aborted`: выполнение задания отменено.

- ts, строка - дата/время, когда наступило соответствующее состояние.

Объект состояния может так же иметь другие атрибуты, не определенные в данном списке.

Текущее состояние задания соответствует элементу с наиболее свежей датой/временем. Порядок, в котором объекты присутствуют в списке, может быть произвольным.

- operation, список объектов - текущие операции и история выполнения операций над заданием. Каждый объект имеет атрибуты:
 - op, строка - операция; возможные значения:
 - start: запуск задания на выполнение;
 - pause: приостановка выполнения задания;
 - abort: отмена выполнения задания.
 - id, строка - уникальный в рамках одного задания идентификатор операции.
 - created, строка - дата и время помещения данной операции в очередь.
 - completed, строка - дата и время завершения данной операции.
 - success, boolean - было ли завершение операции успешным.
 - result, объект - расширенный результат завершения операции, произвольный объект.

Объект операция может так же иметь другие атрибуты, не определенные в данном списке.

Операции обрабатываются системой в хронологическом порядке. Порядок, в котором объекты присутствуют в списке, может быть произвольным.

- definition, объект - описание задания, без описания задач, согласно спецификации описания заданий, задач и ресурсных требований.
- tasks, объект - URI задач задания; атрибуты объекта соответствуют идентификаторам задач из описания задания, значения этих атрибутов — URI соответствующих задач (возможно, относительные)
- deleted, boolean - значение True означает, что данное задание помечено как удаленное; удаленные задания доступны только для чтения.

Возможные коды HTTP ответа: 201, 400, 401, 412.

Пример ответа с информацией о задании

```
{ "created": "2010-07-03T10:27:00Z",
  "modified": "2010-07-03T10:27:14Z",
  "expires": "2010-07-04T10:28:30Z",
  "server_time": "2010-07-03T10:28:30Z",
  "server_policy_uri": "https://pilot.grid-net.ru/policy/job/",
  "definition": { "version": 2,
                  "description": "тестовое задание",
                  "default_storage_base":
"gsiftp://tb01.grid-net.ru/home/shamardin/jt/",
                  "tasks": [ { "id": "a",
                              "description": "задача #1",
                              "definition": { "version": 2,
                                              "executable":
"/usr/bin/whoami",
```

```
        "stdout": "test.txt"
      }
    ]
  }
}

"state": [ { "s": "new",
             "ts": "2010-07-03T10:27:00Z" },
           { "s": "pending",
             "ts": "2010-07-03T10:27:14Z" },
           { "s": "running",
             "ts": "2010-07-03T10:27:22Z" } ],
"owner": "/C=RU/O=NanoGrid/OU=users/OU=sinp.msu.ru/CN=Lev Shamardin",
"operation": [ { "op": "start",
                 "id": "c9deca6c-3208-4146-848b-2b65b0943127",
                 "created": "2010-07-03T10:27:03Z",
                 "completed": "2010-07-03T10:27:14Z",
                 "success": true
               } ],
"tasks": {
  "a": "https://pilot.grid-net.ru/jobs/912832/a/"
},
"deleted": false
}
```

Получение частичной информации о задании

Для получения информации о состоянии задания используется запрос GET к `pilot/jobs/<jobid>/?parts=a;b;c...`, где `<jobid>` — идентификатор задания, а значение параметра `parts` содержит перечисление частей информации о задании, которую необходимо получить. Части информации, которые можно получить:

- `state`: информация о состоянии задания
- `operations`: информация об операциях над заданием

Изменение задания и выполнение операций над заданием

Для изменения задания, а так же для выполнения операций над заданием (например, запуск задания), используются запросы методом PUT по URI задания, который обычно имеет вид `pilot/jobs/<jobid>/`.

Запрос является объектом, имеющим следующие атрибуты

- `definition`, объект, опциональный параметр

Новое описание задания и, возможно, задач, согласно спецификации описания заданий, задач и ресурсных требований. При наличии данного атрибута в запросе произойдет изменение описания задания и, возможно, задач задания. Изменение возможно только для заданий, находящихся в состоянии `new`. Попытка изменения описания задания, находящегося в другом состоянии вызовет ошибку 403. Все задачи, отсутствующие в новом описании задания, будут

сразу же удалены с сервера. Задачи, описания которых отсутствуют в новом описании задания, изменены не будут.

- operation, объект, опциональный параметр - при наличии этого параметра будет создана новая операция над заданием, если ее не существовало ранее\$ объект имеет следующие атрибуты:
 - op, строка - выполняемая операция; возможные значения:
 - start: запустить задание на выполнение;
 - pause: приостановить выполнение задания;
 - abort: отменить выполнение задания.
 - id, строка

Уникальный в рамках одного задания идентификатор операции. Генерация уникального id является ответственностью клиента, в качестве id рекомендуется использовать, например, uuid4.

Возможные коды HTTP ответа: 204, 401, 404.

Пример запроса на запуск задания

```
{ "operation": { "op": "start",  
                "id": "c9deca6c-3208-4146-848b-2b65b0943127" } }
```

Получение списка заданий

Для получения списка заданий пользователя (в любом состоянии) используется запрос методом GET к pilot/jobs/. Ответом сервера является список объектов, каждый из которых имеет следующие атрибуты:

- uri, строка - URI задания
- job_id, строка - идентификатор задания

Возможные коды HTTP ответа: 200, 401. Пример ответа на запрос получения списка заданий

```
[ { "uri": "https://pilot.grid-net.ru/jobs/199231/" },  
  { "uri": "https://pilot.grid-net.ru/jobs/912832/" } ]
```

Получение списка заданий других пользователей

Для получения списка заданий любых пользователей (в любом состоянии) используется запрос методом GET к pilot/jobs/?owner=X с дополнительным параметром owner в QUERY_STRING.

Значения параметра owner интерпретируется как shell-like glob pattern, т.е. все символы кроме * и ? соответствуют самим себе, символ ? соответствует значению «один любой символ», символ * соответствует значению «ноль или более любых символов». Ответом сервера является список объектов, каждый из которых имеет следующие атрибуты:

- uri, строка - URI задания

- owner, строка - DN владельца задания.

Ответ сервера содержит только те задания, доступ к которым разрешен согласно политике доступа к заданиям данного сервера. Например, администратор виртуальной организации может получать доступ ко всем заданиям пользователей данной виртуальной организации и т.п.

Удаление задания

Для удаления задания используется запрос методом DELETE по URI задания. После удаления задания вся информация о нем и его задачах становится недоступной. При удалении задания, которые выполняются в момент удаления, производится принудительная остановка задания.

Возможные коды HTTP ответа: 204, 401, 404.

Получение информации о задаче

Для получения информации о задаче, включающей состояние задачи, используется запрос методом GET по URI задачи, который обычно имеет вид `pilot/jobs/<jobid>/<taskid>/`.

Ответом сервера является объект, имеющий следующие атрибуты:

- created, modified, строка дата/время создания задачи и дата/время последнего изменения задачи или ее состояния.
- job, строка - URI задания, к которому относится данная задача.
- state, список объектов -список состояний задания за все время его существования. Каждое состояние является объектом, имеющим атрибуты:
 - s, строка - состояние; возможные значения:
 - new: новая необработанная задача;
 - pending: задача ожидает начала выполнения;
 - running: задача выполняется;
 - paused: выполнение задачи приостановлено;
 - finished: выполнение задачи завершено;
 - aborted: выполнение задачи отменено.
- ts, строка - дата и время перехода задачи в данное состояние; возможно так же наличие дополнительных атрибутов, специфичных для данного состояния.
- definition, объект - описание задачи согласно спецификации описания заданий, задач и ресурсных требований.
- exit_code, число - код завершения задачи, если он известен.
- deleted, boolean - значение True означает, что данная задача помечена как удаленная; удаленные задачи доступны только для чтения.

Возможные коды HTTP ответа: 200, 401, 404.

Пример ответа с состоянием задачи

```
{ "created": "2010-07-03T10:27:00Z",  
  "modified": "2010-07-03T10:27:02Z",  
  "job": "https://pilot.grid-net.ru/jobs/912832/",
```

```
"definition": { "version": 2,
                "executable": "/usr/bin/whoami",
                "stdout": "test.txt"
              },
"deleted": false,
"state": [ { "s": "new",
             "ts": "2010-07-03T10:27:00Z" },
           { "s": "pending",
             "ts": "2010-07-03T10:27:14Z" },
           { "s": "running",
             "ts": "2010-07-03T10:27:22Z" },
           { "s": "finished",
             "ts": "2010-07-03T10:27:23Z" } ],
"exit_code": 0
}
```

Изменение задачи

Для изменения описания задачи задачи используются запросы методом PUT по URI задачи, который обычно имеет вид `pilot/jobs/<jobid>/<taskid>`. Запрос является объектом, имеющим следующие атрибуты:

- `definition`, объект - уовое описания задачи согласно спецификации описания заданий, задач и ресурсных требований. Изменение задачи возможно только в том случае, если задача находится в состоянии `new`. Попытка изменения описания задачи, находящейся в другом состоянии вызовет ошибку 403.

Пример запроса на изменение задачи

```
{
  "definition": { "version": 2,
                  "executable": "/usr/bin/whoami",
                  "stdout": "test.txt"
                }
}
```

Получение информации о доступных ресурсах

Для получения информации о доступности ресурсов для выполнения задания, включая всего его задачи с учетом требований, используется запрос методом GET по URI вида `pilot/v2/jobs/<jobid>/resources`. Ответом является объект, имеющий следующие атрибуты:

- `runnable`, boolean - можно ли выполнить все задачи задания на существующих в системе ресурсах. Данный атрибут имеет значение истина, если возможно выполнить все задачи задания на существующих ресурсах, и ложь в противном случае.
- `resources`, список объектов - список объектов, каждый из которых представляет информацию о ресурсах, подходящих для одной из задач задания. Каждый объект имеет следующие атрибуты:
 - `task_id`, строка - идентификатор задачи

- resources, список объектов - список совместимых ресурсов (GRAM'ов). Каждый элемент списка является объектом, имеющим следующие атрибуты:
 - host, строка - имя хоста
 - port, число - порт
 - lrms_type, строка - тип lrms
 - queue, строка - название очереди

Поскольку выполнение matchmaking может занимать определенное время, ответ на такой запрос к серверу может занять существенное время.

Запросы аккаунтинговой информации

Для получения аккаунтинговой информации необходимо использовать запросы методом GET по в иерархии pilot/v2/accounting/. Поскольку аккаунтинговая информация может иметь значительные объемы дублирующейся информации, клиентам рекомендуется поддерживать Content-Encoding типа gzip, и сообщать об этом серверу в заголовках запроса через заголовок Accept-Encoding. Сервер по умолчанию возвращает ответы в формате JSON. Возможно возвращение ответов в формате CSV, для этого клиент должен добавить в запрос заголовок:

```
Accept: text/csv
```

Обращения за аккаунтинговой информацией могут производиться как с авторизацией сертификатом или прокси-сертификатом пользователя, так и с авторизацией сертификатом хоста. Вопрос об объеме доступа к аккаунтинговой информации по тому или иному сертификату решается со стороны сервера.

Запрос аккаунтинговой информации за заданный период времени

Запрос производится по URI вида pilot/v2/accounting/period/<ts1>-<ts2>/, где <ts1> и <ts2> могут иметь одно из следующих значений:

- current: текущее дата/время на сервере. Дата и время во временной зоне UTC, в формате YYYYmmddHHMMSS[.FFFFFF], где YYYY — год, mm — месяц, dd — день, HH — час, MM — минуты, SS — секунды, [.FFFFFF] — доли секунды. Например: 20091124124337 соответствует 24 ноября 2009 года, 15:43:34 по Московскому зимнему времени, 20051211182733.832922 соответствует 11 декабря 2005 года, 21:27:33 и 0.832922 секунды по Московскому зимнему времени.

Невыполнение следующих требований приведет к ответу с кодом 400:

- <ts1> не может иметь значение current;
- <ts2> должен быть больше <ts1>.

Примеры URI запросов:

- pilot/v2/accounting/period/20091124000000-current/ — аккаунтинговая информация с 00:00:00 UTC 24 ноября 2009 года по текущий момент
- pilot/v2/accounting/period/20091124000000-20091124124337.291323/ — аккаунтинговая информация с 00:00:00 UTC 24 ноября 2009 года по 12:43:37 и 0.291323 секунды UTC 24

ноября 2009 года.

Запрос последних записей аккаунтинговой информации

Запрос производится по URI вида `pilot/v2/accounting/last/N/`, где N — количество возвращаемых записей. Возвращаются последние N записей аккаунтинга.

Ответы в формате JSON

Ответы сервера на запросы аккаунтинга в формате JSON представляют собой списки объектов, имеющих следующие атрибуты:

- `ts`, строка - дата и время соответствующей записи аккаунтинга, в формате ISO 8601, во временной зоне UTC с долями секунды.
- `user_dn`, строка - Subject сертификата пользователя, являющегося владельцем задания.
- `job_id`, строка - идентификатор задания на данном сервере
- `task_id`, строка, опциональный параметр - идентификатор задачи на данном сервере, из соответствующего задания. `null`, если не имеет отношения к событию.
- `vo`, строка - виртуальная организация, к которой относится данная операция (если известна); `null`, если виртуальная организация не может быть определена.
- `event`, строка - учитываемое событие. Возможные значения:
 - `job_started`: началась обработка задания. Для событий этого типа обязательно присутствуют поля `job_id` и `user_dn`.
 - `job_finished`: обработка задания завершилась успешно (все задания успешно завершились). Для событий этого типа обязательно присутствуют поля `job_id` и `user_dn`.
 - `job_aborted`: обработка задачи завершилась не успешно. В `detail` может содержаться `task_id` задачи, которая привела к не успешному завершению обработки задания. Для события этого типа обязательно присутствуют поля `job_id` и `user_dn`, может присутствовать поле `detail`.
 - `task_started`: задача была запущена. В `detail` содержится информация о GRAM, на котором запущена задача, в формате `host[:port]/type-queue`. Для событий этого типа обязательно присутствуют поля `job_id`, `user_dn`, `task_id` и `detail`.

`task_finished`: задача завершилась успешно. Для событий этого типа обязательно присутствуют поля `job_id`, `user_dn`, `task_id`. Может присутствовать поле `detail`, содержащее код завершения задачи.

- `task_aborted`: задача завершилась не успешно. Для событий этого типа обязательно присутствуют поля `job_id`, `user_dn`, `task_id`. Может присутствовать поле `detail`, содержащее код завершения задачи.
- `detail`, строка, опциональный параметр - дополнительная информация о событии. См. описание значений в описании параметра `event.null`, если дополнительной информации нет.
- `info`, объект - дополнительная информация о событии, если известна; `null`, если дополнительной информации нет. Имеет различные значения для разных типов событий.
- `job_aborted` содержит следующие атрибуты:
 - `task_uri`: строка, URI задачи.
- `task_started` содержит следующие атрибуты:

- hostname: строка, имя хоста, на которую запустилась задача.
- port: число, опциональный параметр, порт GRAM.
- lrms_type: строка, тип lrms.
- queue: строка, опциональный параметр, название очереди.
- submission_id: строка, Globus Submission ID.

Ответы в формате CSV

Ответы сервера на запросы аккаунтинга в формате CSV соответствуют спецификации RFC 4180 с заголовками. Ответ содержит колонки ts, user_dn, job_id, task_id, event, detail, полностью соответствующие соответствующим полям ответа в формате JSON.

Запуск задания

Последовательность действий, которые выполняет клиент для того, чтобы запустить задание, следить за его состоянием и получить результаты:

- Создать задание (POST <base>/jobs/). В описании задания можно сразу передать и описания всех задач.
- Добавить операцию start (PUT <base>/jobs/<jobid>/).

Через некоторое время после этого начнется выполнение задания. За состоянием можно следить при помощи обращений методом GET по URI задания и/или отдельных задач.

Коды состояния и ошибок

200 OK

Запрос обработан без ошибок, тело ответа содержит ответ на запрос в указанном выше формате.

201 Created

Запрос обработан без ошибок, ответ имеет пустое тело и заголовок Location, содержащий URI созданного ресурса.

204 No Content

Запрос обработан без ошибок, ответ имеет пустое тело, содержательного ответа на запрос не предполагается по спецификации протокола. Такой ответ обычно является подтверждением удаления какого-либо ресурса.

400 Bad Request

Ошибка обработки запроса. Запрос содержал данные в неправильном формате, либо пытался изменить параметры, которые нельзя изменить.

401 Unauthorized

Ошибка обработки запроса. Доступ к запрашиваемому объекту запрещен.

403 Forbidden

Попытка выполнения операции, которая запрещена. Например, попытка изменить описание задания в состоянии, отличного от new.

404 Not Found

Ошибка обработки запроса. Запрашиваемый ресурс не существует.

412 Precondition Failed

Ошибка обработки запроса. Контрольная сумма тела запроса не соответствует заголовку Content-MD5 запроса. Тело ответа пустое.

From:

<https://theory.sinp.msu.ru/dokuwiki/> - **THEORY**

Permanent link:

<https://theory.sinp.msu.ru/dokuwiki/doku.php/dcomp/fcp/rest/api>

Last update: **26/01/2017 12:07**

