

# SPCC Rest API, version 1.0

---

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME
1.0	September 2010		SAT

---

---

## Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Browsing the API	1
1.2	HTTP request	1
1.3	Authentication	1
1.4	Data format	1
1.5	Virtual machine tags	2
<b>2</b>	<b>Changelog</b>	<b>2</b>
<b>3</b>	<b>Summary of the URIs</b>	<b>2</b>
<b>4</b>	<b>API</b>	<b>2</b>
4.1	URL: /	2
4.1.1	GET /	2
4.2	URL: /account	3
4.2.1	GET /account	3
4.2.2	POST /account	3
4.3	URL: /hosts/	4
4.3.1	GET /hosts/	4
4.3.2	POST /hosts/	5
4.4	URL: /hosts/:id	6
4.4.1	GET /hosts/:id	6
4.4.2	POST /hosts/:id	7
4.4.3	DELETE /hosts/:id	9
4.5	URL: /hosts/:id/snapshots	9
4.5.1	GET /hosts/:id/snapshots	9
4.6	URL: /hosts/:id/control	9
4.6.1	POST /hosts/:id/control	9
4.7	URL: /hosts/:id/activity/cpu/:span	10
4.7.1	GET /hosts/:id/activity/cpu/:span	10
4.8	URL: /hosts/:id/activity/network/:span	10
4.8.1	GET /hosts/:id/activity/network/:span	10
4.9	URL: /library/	10
4.9.1	GET /library	10
4.10	URL: /library/:id	11
4.10.1	GET /library/:id	11
4.10.2	POST /library/:id	11
4.10.3	DELETE /library/:id	11

---

---

4.11 URL: /images/ . . . . .	12
4.11.1 GET /images/ . . . . .	12
4.12 URL: /images/:id . . . . .	12
4.12.1 GET /images/:id . . . . .	12
4.13 URL: /snapshots/ . . . . .	13
4.13.1 GET /snapshots/ . . . . .	13
4.13.2 POST /snapshots/ . . . . .	13
4.14 URL: /snapshots/:id . . . . .	14
4.14.1 GET /snapshots/:id . . . . .	14
4.14.2 POST /snapshots/:id . . . . .	14
4.14.3 DELETE /snapshots/:id . . . . .	15
4.15 URL: /snapshots/:id/control . . . . .	15
4.15.1 POST /snapshots/:id/control . . . . .	15
4.16 URL: /invoices/ . . . . .	15
4.16.1 GET /invoices/ . . . . .	15
4.17 URL: /invoices/:id . . . . .	16
4.17.1 GET /invoices/:id . . . . .	16
4.18 URL: /tickets/ . . . . .	16
4.18.1 GET /tickets/ . . . . .	16
4.18.2 POST /tickets/ . . . . .	16
4.19 URL: /tickets/:id . . . . .	17
4.19.1 GET /tickets/:id . . . . .	17
4.19.2 POST /tickets/:id . . . . .	17

## 5 Using the API in your software

17

## 1 Overview

The SPCC REST API provides a mean to interact with our infrastructure to manage and monitor one or several virtual machines and their associated services.

It is reachable for any user who own an API key, through a HTTPS connection. All you need is your API key and the entry point of the API which is:

```
https://api.soprive.net/
```

You can find your API key by looking at the "API" tab under the "Account" tab on your administration interface.

### 1.1 Browsing the API

The same URL can also be used to interactively browse the API with your browser. Enter you API key in the `Key:` field and hit `Set`.

To browse the API interactively, go to the following address with a web browser:

```
https://api.soprive.net/
```

### 1.2 HTTP request

You must be able to perform the following HTTP requests: `GET`, `POST`, `PUT` and `DELETE`. These verbs allow to manage resources. A resource is a generic concept which names anything which is manageable by the API. (A virtual machine, a snapshot, your account settings, the activity log, and so on.)

- The `GET` verb is used to retrieve information about some resource: either information about a virtual machine, the list of the snapshots you made, the settings about your account,...
- The `DELETE` verb delete the corresponding resource. It might be a virtual machine, a snapshot, a service attached to your account,...
- The `POST` verb is used to create or update something. For example, to create a new snapshot or to update your account information such as phone number. When something is created, the location of the new resource is returned in the `Location` header.
- The `PUT` verb is used to replace an existing resource. The method is not in use currently. You can ignore it.

### 1.3 Authentication

You must pass your API key in the "Authorization" field of each HTTP request using the "Basic" scheme. The API key is the login, and the password part is ignored.

Check with your HTTP library how to provides this field. Some examples for common languages are given later in this document.

### 1.4 Data format

Data are exchanged using the JSON encoding scheme.

For request with the verb `POST` or `PUT`, you must usually encode data using JSON. Response from the API server are also encoded using JSON.

---

## 1.5 Virtual machine tags

You can associate up to 128 tags to a virtual machines.

Tag names must composed of printable ASCII characters (except the space) and up to 64 characters.

The tags have no particular meaning for our platform. You're free to put any tag you want. They can be used to filter the virtual machine list and they are listed in each virtual machine details.

Some valid tags: "webserver", "foo:bar:4321".

Invalid tags: "π", "some tag", "café".

## 2 Changelog

- Version 1.0: Initial public release.

## 3 Summary of the URIs

- Get the list of VM: GET /hosts/
- Create a VM: POST /hosts/
- Destroy a VM: DELETE /hosts/:id
- Update or resize a VM: POST /hosts/:id
- View VM details: GET /hosts/:id
- Start or poweroff a VM: POST /hosts/:id/control
- Create a snapshot: POST /snapshots/
- Delete a snapshot: DELETE /snapshots/:id
- View snapshot details: GET /snapshots/:id
- Restore a snapshot: POST /snapshots/:id/control

## 4 API

### 4.1 URL: /

#### 4.1.1 GET /

Retrieve the root information of the API.

**Example:**

```
{
  "tickets": {"url": "/tickets/"},
  "images": {"url": "/images/"},
  "account": {"url": "/account/"},
  "hosts": {"url": "/hosts/"},
  "library": {"url": "/library/"},
  "invoices": {"url": "/invoices/"},
  "api-version": [0, 1, 0, 0],
  "snapshots": {"url": "/snapshots/"}
```

The `api-version` entry gives the current version of the API. In this example it should be interpreted as version 0.1.0.0.

## 4.2 URL: /account

### 4.2.1 GET /account

Retrieve details about your account.

**Example:**

```
{
  "credit": {
    "currency": "EUR",
    "amount": "193.65"
  },
  "email-address": "jd@example.org",
  "account-type": "individual",
  "firstname": "John",
  "title": "mr",
  "lastname": "Doe",
  "creation": 1267630795,
  "ftp-prefix": "112-916-328",
  "address": {
    "city": "Paris",
    "street": [
      "4 rue du petit pont"
    ],
    "zipcode": "75003"
  },
  "phone": {
    "mobile": "06xxxxxxxx",
    "landline": null
  }
}
```

The `ftp-prefix` is needed to upload ISO image to our FTP server. See the So Privé FAQ for more details about that.

### 4.2.2 POST /account

Update account details.

To update some of your account details, post the following data:

```
{
  "address": {
    "street": [<TEXT>, <TEXT>] # or [<TEXT>]
    "city": <TEXT>,
    "zipcode": <TEXT>
  }
}
```

All fields are optional.

You cannot update other fields.

**Status:**

- 200: Update successful.
- 400: Malformed request.

**Example:**

---

```
{
  "address": {
    "street": ["1, rue du Fossé-des-Tanneurs"],
    "city": "Strasbourg",
    "zipcode": "67200"
  }
}
```

## 4.3 URL: /hosts/

### 4.3.1 GET /hosts/

Retrieve the list of virtual machines.

Optional arguments:

- state={ SPARE | STARTING | RUNNING | STOPPING | DESTROYING | DESTROYED }

Filter the list of virtual machine by their state. By default, destroyed virtual machines doesn't appear in the list. If you want the list of previously destroyed virtual machines, use `/hosts/?state=DESTROYED`.

- tag=<TAGNAME>

Filter the list of virtual machine by tag. Only the virtual machines with the corresponding tag will be listed in the result. See `POST /host/:id` to learn how to set tags for a virtual machine.

You can use both filter at the same time.

#### Example with GET /hosts/:

```
[
  {
    "url": "/hosts/v-0cc96f1e",
    "state": "RUNNING",
    "id": "v-0cc91e6f",
    "label": "pgsql1"
  },
  {
    "url": "/hosts/v-4de0ec1a",
    "state": "RUNNING",
    "id": "v-4de0ec1a",
    "label": "fe-www1"
  },
  {
    "url": "/hosts/v-7c60e52e",
    "state": "SPARE",
    "id": "v-7c60e52e",
    "label": "fe-www2"
  },
  {
    "url": "/hosts/v-f52e1455",
    "state": "RUNNING",
    "id": "v-f52e1455",
    "label": "backoffice"
  },
  {
    "url": "/hosts/v-4e321f8d",
    "state": "RUNNING",
    "id": "v-4e321f8d",
```

```
"label": "mx1"
}
]
```

**Example with GET /hosts/?state=SPARE:**

```
[
{
  "url": "/hosts/v-7c60e52e",
  "state": "SPARE",
  "id": "v-7c60e52e",
  "label": "fe-www2"
}
]
```

**Example with GET /hosts/?tag=webserver:**

```
[
{
  "url": "/hosts/v-4de0ec1a",
  "state": "RUNNING",
  "id": "v-4de0ec1a",
  "label": "fe-www1"
},
{
  "url": "/hosts/v-7c60e52e",
  "state": "SPARE",
  "id": "v-7c60e52e",
  "label": "fe-www2"
}
]
```

(This last example assumes that both virtual machines have the `webserver` tag.)

**Example with GET /hosts/?tag=webserver&state=RUNNING:**

```
[
{
  "url": "/hosts/v-4de0ec1a",
  "state": "RUNNING",
  "id": "v-4de0ec1a",
  "label": "fe-www1"
}
]
```

### 4.3.2 POST /hosts/

Create a virtual machine.

To create a virtual machine, post the following data to this URI:

```
{
  "core": <CORE>,
  "memory": <SIZE>,          # in gibibyte[1]
  "image": <IMAGE>,          # note: conflict with "snapshot"
  "storage": <SIZE>,         # in gigabyte[2] (if "image" is set)
  "snapshot": <SNAPSHOT>,   # note: conflict with "image"
  "sla": "std"|"ind"|"pro",
  "billing": "hourly"|"monthly",
  "label": <TEXT>,           # optional
  "sshkey": <TEXT>,          # optional
}
```

```
"tags": [<TAG>..] # optional
}
```

You can either create a virtual machine from one of our images, or directly from one of your snapshots.

- To create a virtual machine from one of our images, you must set `storage` and `image`.
- To create a virtual machine from one of your snapshots, you must set `snapshot`.

The URI of the new virtual machine will be given in the `Location` field of the response.

<sup>1</sup> A gibibyte is  $1024^3$  bytes.

<sup>2</sup> A gigabyte is  $1000^3$  bytes.

#### Status:

- 201: Virtual machine created. The response also contains the details of the new VM. See `GET /hosts/:id`.
- 400: Malformed request.
- 403: Usually because you try to create a VM when you've reached your quota or because you're using an unknown (or busy) snapshot.

#### Example to create a virtual machine from an image:

```
{
  "core": 2,
  "memory": 4,
  "storage": 50,
  "image": "gentoo-2010-10",
  "sla": "pro",
  "label": "www2",
  "billing": "hourly",
  "tags": ["webserver", "custom-id:42"]
}
```

#### Example to create a virtual machine from a snapshot:

```
{
  "core": 2,
  "memory": 4,
  "snapshot": "d25a9a78-16ba-46a0-8ed8-8c883f34043a",
  "sla": "pro",
  "label": "www2",
  "billing": "hourly",
  "tags": ["render-node"]
}
```

## 4.4 URL: /hosts/:id

### 4.4.1 GET /hosts/:id

Retrieve details about a virtual machine.

#### Status:

- 200: OK
- 404: Unknown virtual machine.

**Example:**

```
{
  "id": "v-4a9d3a86",
  "label": null,
  "creation": 1285154493,
  "state": "SPARE",
  "locked": false,
  "core": 1,
  "memory": 1,
  "storage": [
    {"image": "redhat-el5-server", "lun": 1, "size": 10},
    {"image": "blank", "lun": 2, "size": 90}
  ],
  "extradisk": {
    "iso": {
      "url": "/library/d4e8abae-bc3a-4a6c-aeeb-75a7c23372ff",
      "id": "d4e8abae-bc3a-4a6c-aeeb-75a7c23372ff"
    },
    "boot-first": false
  },
  "network": [{"hwaddr": "5e:cc:12:34:56:78", "ip": ["127.167.78.106"]}],
  "sla": "std",
  "activity": {
    "cpu": {"url": "/hosts/v-4a9d3a86/activity/cpu/"},
    "network": {"url": "/hosts/v-4a9d3a86/activity/network/"}
  },
  "tags": ["webserver", "unix", "value:42"],
  "billing": {
    "next-period": 1285158096,
    "cost": {"amount": "0.12", "currency": "EUR"},
    "first-period": 1285154496,
    "mode": "hourly"
  },
  "snapshots": {"url": "/hosts/v-4a9d3a86/snapshots"},
  "control": {"url": "/hosts/v-4a9d3a86/control"}
}
```

**4.4.2 POST /hosts/:id**

Update the virtual machine.

You can change one or several of the following settings:

- the label
- the list of tags
- the dimension of the virtual machine
- the billing mode (you can only switch to "monthly" for now)
- the ISO extra disk
- the storage space.

To change the label, post the following data:

```
{"label": <TEXT>}
```

To set a list of tags, post the following data:

```
{"tags": [<TEXT>..]}
```

To detach the ISO disk, post the following data:

```
{"extradisk": null}
```

To attach an ISO disk, post the following data:

```
{"extradisk":  
  {"iso": <ID>,           # an ISO ID from your library  
    "boot-first": false|true # optional, default to false  
  }}
```

To set only the `boot-first` flag:

```
{"extradisk": {"boot-first": false|true}}
```

To resize the virtual machine, post the following data:

```
{"core": <CORE>, "memory": <SIZE>, "billing": "monthly"}
```

(See virtual machine creation for details about these values.)

Each fields are optional.

The virtual machine must be stopped if you attempt to resize the memory or the number of core.

The billing mode can only be changed to `monthly`. This change can occur even if the virtual machine is running.

#### Status:

- 200: Virtual machine updated successfully.
- 400: Malformed request.
- 404: Unknown virtual machine.

#### Example to change the label:

```
{"label": "www3"}
```

#### Example to change the tags list:

```
{"tags": ["webserver", "monitored", "unix"]}
```

#### Example to attach an ISO image:

```
{"extradisk": {"iso": "133e72d1-538b-4c6b-b6f3-70b70c3c944b", "boot-first": true}}
```

#### Example to change the boot-first flag:

```
{"extradisk": {"boot-first": false}}
```

#### Example to resize the virtual machine:

```
{"core": 4, "memory": 6}
```

### 4.4.3 DELETE /hosts/:id

Delete a virtual machine.

The virtual machine must be stopped before you can delete it.

All the virtual machine storage will be deleted too. If you want to keep a copy of the virtual machine storage, first make a snapshot of it before deleting the virtual machine.

**Status:**

- 200: Virtual machine deleted successfully.
- 404: Unknown virtual machine.
- 409: Virtual machine is busy.

## 4.5 URL: /hosts/:id/snapshots

### 4.5.1 GET /hosts/:id/snapshots

Retrieve the list of snapshots related to the virtual machine.

**Status:**

- 200: OK
- 404: Unknown virtual machine.

**Example:**

```
[
  {
    "url": "/snapshots/38b18ca1-d675-46b5-638a-c8aa9eeaaf9d",
    "creation": 1280475968,
    "title": "Backup from a clean install"
  },
  {
    "url": "/snapshots/b5680d8a-0104-4f46-8a94-58da5e355eaa",
    "creation": 1282645748,
    "title": "Backup before upgrade to BigSoftware 2.0"
  }
]
```

## 4.6 URL: /hosts/:id/control

### 4.6.1 POST /hosts/:id/control

Control the virtual machine.

You can start or poweroff the virtual machine using this URI by posting the following data:

```
{"action": "START"|"POWEROFF"}
```

**Status:**

- 200: Order sent to the virtual machine.
- 404: Unknown virtual machine.
- 409: The virtual machine is not in an appropriate state.

**Example:**

```
{"action": "START"}
```

## 4.7 URL: /hosts/:id/activity/cpu/:span

### 4.7.1 GET /hosts/:id/activity/cpu/:span

Response type: PNG

Retrieve a graph of the CPU activity.

The span can be either day, week, month or year.

**Status:**

- 200: OK
- 404: Unknown virtual machine.

## 4.8 URL: /hosts/:id/activity/network/:span

### 4.8.1 GET /hosts/:id/activity/network/:span

Response type: PNG

Retrieve a graph of the network activity.

The span can be either day, week, month or year.

**Status:**

- 200: OK
- 404: Unknown virtual machine.

## 4.9 URL: /library/

### 4.9.1 GET /library

Get the list of ISO image you have uploaded to our server.

**Example:**

```
[
  {
    "url": "/library/3658a3c2-419b-496d-a9d4-f9351d2c1d53",
    "bytesize": 689313008,
    "id": "3658a3c2-419b-496d-a9d4-f9351d2c1d53",
    "title": "Custom Debian Lenny Live CD"
  },
  {
    "url": "/library/30f5e0f5-aef9-4ea4-a0e1-5a5309468ff6",
    "bytesize": 3982503449,
    "id": "30f5e0f5-aef9-4ea4-a0e1-5a5309468ff6",
    "title": "Full data backup from Foobar"
  }
]
```

## 4.10 URL: /library/:id

### 4.10.1 GET /library/:id

Get details about as ISO image from your library.

**Status:**

- 200: OK
- 404: Unknown ISO image.

**Example:**

```
{
  "bytesize": 689313008,
  "creation": 1282820740,
  "id": "3658a3c2-419b-496d-a9d4-f9351d2c1d53",
  "title": "Debian Lenny Live CD"
}
```

### 4.10.2 POST /library/:id

Update details of the ISO image.

You can update the title by posting to this URI the following data:

```
{"title": <TEXT>}
```

**Status:**

- 200: ISO image updated.
- 400: Malformed request.
- 404: Unknown ISO image.

**Example:**

```
{"title": "Extra packages v1.2"}
```

### 4.10.3 DELETE /library/:id

Delete an ISO image.

The ISO image must not be currently attached to a virtual machine.

**Status:**

- 200: ISO image deleted successfully.
  - 404: Unknown ISO image.
-

## 4.11 URL: /images/

### 4.11.1 GET /images/

Retrieve the list of disk image that can be used to create a new virtual machine.

#### Example:

```
[
  {
    "category": "linux",
    "description": "Linux GNU/Debian Lenny x86-64 OS",
    "title": "Linux GNU/Debian Lenny 5.0.4 x86-64 OS",
    "url": "/images/debian-lenny-504-img",
    "id": "debian-lenny-504-img",
    "min-size": 9
  },
  {
    "category": "netbsd",
    "description": "NetBSD 5.0.2 AMD64",
    "title": "NetBSD 5.0.2 AMD64",
    "url": "/images/nbsd-502-amd64-master-3",
    "id": "nbsd-502-amd64-master-3",
    "min-size": 10
  },
  {
    "category": "ms-windows",
    "description": "Microsoft Windows 2008 Datacenter R2",
    "title": "Microsoft Windows 2008 Datacenter R2",
    "url": "/images/w2k8-master-1",
    "id": "w2k8-master-1",
    "min-size": 50
  }
]
```

The `min-size` gives the minimum storage (in gigabyte) that can be set when creating a virtual machine with the corresponding image.

## 4.12 URL: /images/:id

### 4.12.1 GET /images/:id

Get details about a disk image.

#### Status:

- 200: OK
- 404: Unknown disk image.

#### Example:

```
{
  "id": "nbsd-502-amd64-master-3",
  "title": "NetBSD 5.0.2 AMD64",
  "description": "NetBSD 5.0.2 AMD64",
  "min-size": 10
}
```

## 4.13 URL: /snapshots/

### 4.13.1 GET /snapshots/

Get the list of snapshots you created from your virtual machines.

**Example:**

```
[
  {
    "url": "/snapshots/38b18ca1-d675-46b5-8a63-c8aa9eeaaf9d",
    "state": "READY",
    "size": 10,
    "id": "38b18ca1-d675-46b5-8a63-c8aa9eeaaf9d",
    "title": null
  },
  {
    "url": "/snapshots/78850423-c766-49e0-ab64-6ff1372260e7",
    "state": "READY",
    "size": 10,
    "id": "78850423-c766-49e0-ab64-6ff1372260e7",
    "title": null
  },
  {
    "url": "/snapshots/0fb3523d-b5af-422b-a6b1-baf9c3e52b29",
    "state": "READY",
    "size": 10,
    "id": "0fb3523d-b5af-422b-a6b1-baf9c3e52b29",
    "title": "Snapshot before upgrading to BigApplication 2.1"
  }
]
```

### 4.13.2 POST /snapshots/

Create a new snapshot.

You can create a snapshot of the storage of a virtual machine by posting to this URI the following data:

```
{
  "vm": <ID>,
  "title": <TEXT>, # optional
  "comment": <TEXT> # optional
}
```

The URI of the new snapshot will be given in the `Location` field in the response.

**Status:**

- 201: Snapshot created. The response also contains the details of the new snapshot. See `GET /snapshots/:id`.
- 400: Malformed request.
- 403: Unknown virtual machine.
- 409: The virtual machine is not stopped.

**Example:**

```
{"vm": "v-00000000", "title": "Weekly snapshot - 2010-12-19"}
```

## 4.14 URL: /snapshots/:id

### 4.14.1 GET /snapshots/:id

Get details about a snapshot.

**Status:**

- 200: OK
- 404: Unknown snapshot.

**Example:**

```
{
  "id": "38b18ca1-d675-46b5-8a63-c8aa9eeaaaf9d",
  "state": "READY",
  "creation": 1280475968,
  "title": "Snapshot Gentoo #1",
  "comment": "Un snapshot de test.",
  "included-in-library": false,
  "storage": [
    { "image": "gentoo-2010-0", "lun": 1, "size": 10 }
  ],
  "vm": { "id": "v-e2da5e97", "url": "/hosts/v-e2da5e97" },
  "control": { "url": "/snapshots/38b18ca1-d675-46b5-8a63-c8aa9eeaaaf9d/control" }
}
```

### 4.14.2 POST /snapshots/:id

Update details about a snapshot.

You can update the title, the comment or change if the snapshot should be included in your library or not.

You can post the following data to update them:

```
{
  "title": <TEXT>,
  "comment": <TEXT>,
  "included-in-library": false|true
}
```

All fields are optional.

**Status:**

- 200: Snapshot updated.
- 400: Malformed request.
- 404: Unknown snapshot.

**Example:**

```
{"included-in-library": true}
```

### 4.14.3 DELETE /snapshots/:id

Delete a snapshot.

The snapshot must not be in use (being restored to a virtual machine) when deleting it.

**Status:**

- 200: Snapshot deleted.
- 404: Unknown snapshot.
- 409: Invalid snapshot state.

### 4.15 URL: /snapshots/:id/control

#### 4.15.1 POST /snapshots/:id/control

Control the snapshot.

You can restore the snapshot to an existing virtual machine by posting to this URI.

**Status:**

- 200: Order sent to the snapshot.
- 400: Malformed request.
- 404: Unknown snapshot.

**Example:**

```
{
  "action": "RESTORE",
  "target": "v-12345678"
}
```

### 4.16 URL: /invoices/

#### 4.16.1 GET /invoices/

Get the list of invoices.

**Status:**

- 200: OK

**Example:**

```
[
  {
    "id": "FA100600019",
    "url": "/invoices/FA100601911",
    "type": "application/pdf",
    "date": 1275346800,
    "total": {"amount": "94.39", "currency": "EUR"},
    "total-vat": {"amount": "112.89", "currency": "EUR"}
  },
  {
    "id": "FA100600019",
```

```
"url": "/invoices/FA100700398",
"type": "application/pdf",
"date": 1277938800,
"total": {"amount": "124.11", "currency": "EUR"},
"total-vat": {"amount": "103.77", "currency": "EUR"}
}
]
```

## 4.17 URL: /invoices/:id

### 4.17.1 GET /invoices/:id

Get the invoice as a PDF document.

**Status:**

- 200: OK
- 404: Unknown invoice.

## 4.18 URL: /tickets/

### 4.18.1 GET /tickets/

Get the list of tickets related to your account.

**Status:**

- 200: OK

### 4.18.2 POST /tickets/

Create a new ticket for the So Privé support.

To create a ticket you must post the following data:

```
{"subject": <TEXT> # optional
"body": <TEXT>}
```

The URI of the new ticket will be given in the `Location` field of the response.

**Status:**

- 201: Ticket created.
- 400: Malformed request.

**Example:**

```
{"body": "Body for a new ticket"}
```

## 4.19 URL: /tickets/:id

### 4.19.1 GET /tickets/:id

Get details about a ticket.

**Status:**

- 200: OK
- 404: Unknown ticket.

**Example:**

```
{
  "id": 2,
  "state": "OPEN",
  "closed": null,
  "subject": "Question regarding XYZ",
  "body": "Can you explain why [..]",
  "creation": 1281104989,
  "comments": [
    {
      "date": 1281106022,
      "text": "You observe this behavior because [..]",
      "from": "admin"
    },
    {
      "date": 1281110975,
      "text": "Thanks for your reply. However, [..]",
      "from": "user"
    }
  ]
}
```

### 4.19.2 POST /tickets/:id

Post a comment to the ticket.

To post a comment you must post the following data:

```
{"comment": <TEXT>}
```

**Status:**

- 200: Comment added.
- 400: Malformed request.
- 404: Unknown ticket.

**Example:**

```
{"comment": "A comment for the API example"}
```

## 5 Using the API in your software

(Various examples will be added later.)