

Open Service Broker API(v2.13)

<https://github.com/openservicebrokerapi/servicebroker/blob/v2.13/spec.md>

1. API Overview

Service Broker API: 定义services marketplace平台和服务 brokers的HTTP接口

- **service broker**: 实现API的服务组件。平台marketplace为它的client。`Servicebroker`负责广播catalog(包括service offerings和服务 plans)到marketplace, 根据marketplace请求, ServiceBroker负责: provisioning、binding、unbinding和deprovisioning
 - **provisioning**: 保留service的一个resource。保留资源称为**ServiceInstance**。ServiceInstance代表内容跟service相关。如多租户server的单个database, 独立集群或web应用的一个账户
 - **binding**: 每个service可能不同。binding通常会创建生成必要credentials访问资源, 或提供ServiceInstance的相关配置变更信息
- 应用市场可从多个Servicebrokers暴露services: Servicebroker可使用不同的URL前缀和credentials, 支持多个平台市场

2. 术语

规范定义以下terms

- **Platform**: 管理cloud环境软件, 在这里provisioned引用和服务 Broker。用户不会直接从ServiceBroker创建Service, 而是请求平台(如cloud provider)去管理服务, 并由平台负责与服务Broker交互
- **Service**: 一个应用可使用managed software offering。通常Services会暴露可被执行部分动作的API。也有一些非交互Services, 可执行期望动作, 不需在应用中有直接交互提示
- **ServiceBroker**: ServiceBroker管理服务生命周期。Platforms与服务Broker交互。provision和管理 `ServiceInstance` 和 `ServiceBinding`
- **ServiceInstance**: Service offering实例化
- **ServiceBinding**: 关联引用到 `ServiceInstance` 标识。通常ServiceBindings会包含credentials, 用于与服务Instance通信
- **Application**: 使用或绑定到 `ServiceInstance` 的软件

3. API Version Header

平台对ServiceBroker请求需包括一个header, 指定marketplace使用的API版本号

- `X-Broker-API-Version: 2.13`
- 版本号格式: `MAJOR.MINOR`

该header允许Broker拒绝不支持版本的marketplace请求。虽然API revisions总是additive, brokers可能依赖平台所支持的更新API版本功能。这时broker可能会拒绝请求, 返回 `412 Precondition Failed` 并提供消息, 告诉使用者会使用的API版本

4. 验证

虽然platform和broker通信可为unsecure，建议所有通信为secured，通过TLS和authenticated

- 除非平台和service broker之间有部分out of band通信和agreement。marketplace需要每个请求使用HTTP basic验证service broker (`Authorization:` header)。规范没有指定平台和broker之间的验证方法
- 如果验证失败，broker需要使用predetermined验证机制验证请求；若验证失败，需返回 `401 Unauthorized` 的response
- 注意：通过out of band通信使用agreed的验证机制，会导致与其它平台的互操作性问题

5. URL属性

规范定义以下属性可在URLs出现

- `service_id`、`plan_id`、`instance_id`、`binding_id`、`operation`

6. Originating Identity

通常ServiceBroker需知道从平台初始化请求的用户标识。比如可用于审计或授权。为了协助这个，平台需要在每个请求提供identification信息到broker。Platforms可以支持该功能，若支持，需要遵循：

- 对于任何由平台用户执行的OSB API请求，每个HTTP请求需要有相关 `OriginatingIdentity` header
- 任何与平台用户动作没有关系的OSB API request，如平台查询marketplace，可在HTTP请求中排除该header
- 若在请求中存在，`OriginatingIdentity` header需要包含identify信息，代表执行动作导致请求被发送的平台用户

header格式需为：

```
X-Broker-API-Originating-Identity: platform value
```

`platform`：非空字符串，代表请求发送的平台。特定value需要匹配[profile](#)文档定义values(`context.platform` 属性)。当 `context` 作为消息部分发送，该value需要与 `context.platform` value相同

`value`：Base64编码字符串。字符串需为序列化JSON对象。特定属性与平台相关。查看[profile](#)文档

For example:

```
X-Broker-API-Originating-Identity: cloudfoundry
eyJANCiAgInVzZXJfaWQiOiAiInJgZzZWE3NDgtMzA5Mi00ZmY0LWI2NTYtMzljYWNjNGQ1MzYwIiw
NCiAgInVzZXJfbmFtZSI6ICJqb2VAZXhhbXBsZS5jb20iDQp9
```

`value` 解码时，为

```
{  
  "user_id": "683ea748-3092-4ff4-b656-39cacc4d5360"  
}
```

注意并非所有发送到Broker的消息被平台最终用户初始化。例如在orphan mitigation或查询broker的catalog, platform可能没有对应请求的最终用户, 从而originating identity header不会包含

7. Catalog管理(/v2/catalog)(Get)

/v2/catalog: platform与broker第一个交互endpoint。返回Broker可用services列表。Platforms从所有Brokers查询该endpoint, 提供汇总catalog给用户

- Platform定期重新查询broker的 `/v2/catalog`: 查看services列表是否有变更。Brokers可增加、删除修改 (metadata、plans等)上次查询的services列表
- 当判断broker有内容变更, Platform会使用资源(services或plans)的 `id`, 作为immutable属性, 使用其定位上次查询相同资源。同样broker在查询间, 不能更改资源id, 否则Platform会将其看成不同资源
- 当Platform收到相同类型资源不同 `id`, 就算是资源所有metadat相同, 也需将其看成该资源不同实例
- Broker作者需小心: 从catalog中删除services和plans时, 由于platforms可能已提供这些plans的实例。例如若相关service plan被删除时, platforms可限制用户在已有 `ServiceInstance` 的可执行actions。考虑deprecation策略

以下描述API的catalog请求和响应

7.2 请求

- `GET /v2/catalog`: Headers定义如下
 - `X-Broker-API-Version*`: string。查看[API Version Header](#)
- 例子:

```
$ curl -H "X-Broker-API-Version: 2.13" http://username:password@broker-url/v2/catalog
```

- 实现首个endpoint `GET /v2/catalog` 后, service broker需注册到平台, 让services和plans对最终用户可用

7.3 响应

- 200 OK: 成功处理请求时返回。返回body为
 - `services*`: service-objects数组。service的Schema定义如下。可为空
- CLI和web clients对service和plan names有不同需求。CLI-friendly字符串为全部小写, 忽略空格。尽量保持短小。web友好显示名字为驼峰, 支持空格和标点符号

7.3.1. Service对象

响应字段	类型	描述
name*	string	service的CLI友好name。只能小写，数字和-(没空格)。response返回中唯一。非空字符串
id*	string	标识符。后续请求在broker关联service。全局唯一。非空字符串。建议GUID
description*	string	service的short描述。不为空
tags	strings 数组	Tags提供灵活机制暴露分类、属性或使用基本技术，enabling equivalent services to be swapped out without changes to dependent logic in applications, buildpacks, or other services。如mysql, relational, redis, key-value, caching, messaging, amqp.
requires	strings 数组	permissions列表，用户需要在provision时，提供给service。当前仅支持 <code>syslog_drain</code> ， <code>route_forwarding</code> 和 <code>volume_mount</code>
bindable*	boolean	指定service的service instance是否可绑定到应用。service中所有plans默认值，Plans对象也可覆盖该字段
metadata	JSON 对象	service offering的metadata的opaque对象。Controller将其看成blob。Note that there are conventions in existing brokers and controllers for fields that aid in the display of catalog data.
dashboard_client	object	包含必要数据，激活service的Dashboard SSO feature功能
plan_updateable	boolean	services是否支持plan的upgrade/downgrade。注意misspelling of the attribute <code>plan_updatable</code> to <code>plan_updateable</code> was done by mistake。我们选择保留而非修复，怕破坏兼容以前。默认false
plans*	objects 数组	se'rvice的多个plans列表。schema定义如下。需包括至少一个plan

注意：Platforms通常会用service name作为输入参数，指定需实例化service。注意这些value在平台市场中需唯一。为实现该目标，service providers通常设置service的names前缀为某唯一值(如公司名)。另外有些Platform可能提供给用户前修改service names。该规范对如何暴露values到用户没有需求

7.3.2. DashboardClient对象

响应字段	Type	描述
id	string	dashboard使用Oauth client。若存在，需非空字符串
secret	string	dashboard client的secret。若存在，需非空字符串
redirect_uri	string	service dashboard的URI。dashboard请求token时，OAuth token server负责验证

7.3.3. Plan对象

响应字段	Type	描述
id*	string	标识符。后续请求broker时关联plan。全局唯一。非空字符串，建议GUID
name*	string	CLI友好name。只能小写字母和hyphens(没空格)。service中需唯一。非空
description*	string	plan的short描述。非空字符串
metadata	JSON对象	service plan中metadata的opaque对象。Controller将其看做blob。注意有惯例字段，帮助显示catalog数据的信息
free	boolean	false时，该plan的service instances免费。默认true
bindable	boolean	指定service plan的service instances是否可绑定到应用。可选字段。若指定，优先级比service的 <code>bindable</code> 高。没指定，默认值来源于service
schemas	object	service instances的Schema定义和plan的bindings

7.3.4. Schema对象

响应字段	Type	描述
service_instance	object	创建/修改service instance的schema定义
service_binding	object	创建service bindings的schema定义。仅在service plan为bindable时使用

7.3.5. ServiceInstances对象

响应字段	Type	描述
create	object	创建service instance的schema定义
update	object	更新service instance的schema定义

7.3.6. ServiceBindings对象

响应字段	Type	描述
create	object	创建service bindings的schema定义

7.3.7. InputParameters对象

响应字段	Type	描述
parameters	JSON schema对象	输入参数schma定义。每个输入参数使用JSON对象表示

若catalog中包含 `parameters`，使用以下rules

- Platforms需至少支持[JSON Schema draft v4](#).
- Platforms也需要准备支持以后版本的JSON schema
- `$schema` key需要在scheme中才，定义使用的JSON schema版本
- Schemas不能包含任何外部引用
- Schemas不能超过64kB

```
{
  "services": [{
    "name": "fake-service",
    "id": "acb56d7c-XXXX-XXXX-XXXX-feb140a59a66",
    "description": "fake service",
    "tags": ["no-sql", "relational"],
    "requires": ["route_forwarding"],
    "bindable": true,
    "metadata": {
      "provider": {
        "name": "The name"
      },
      "listing": {
        "imageUrl": "http://example.com/cat.gif",
        "blurb": "Add a blurb here",
        "longDescription": "A long time ago, in a galaxy far far away..."
      },
      "displayName": "The Fake Broker"
    },
  },
  "dashboard_client": {
    "id": "398e2f8e-XXXX-XXXX-XXXX-19a71ecbcf64",
    "secret": "277cabb0-XXXX-XXXX-XXXX-7822c0a90e5d",
    "redirect_uri": "http://localhost:1234"
  },
  "plan_updateable": true,
  "plans": [{
    "name": "fake-plan-1",
    "id": "d3031751-XXXX-XXXX-XXXX-a42377d3320e",
    "description": "Shared fake Server, 5tb persistent disk, 40 max concurrent connections",
    "free": false,
    "metadata": {
      "max_storage_tb": 5,
      "costs": [
        {
          "amount": {
```

```

        "usd":99.0
    },
    "unit":"MONTHLY"
},
{
    "amount":{
        "usd":0.99
    },
    "unit":"1GB of messages over 20GB"
}
],
"bullets": [
    "Shared fake server",
    "5 TB storage",
    "40 concurrent connections"
]
},
"schemas": {
    "service_instance": {
        "create": {
            "parameters": {
                "$schema": "http://json-schema.org/draft-04/schema#",
                "type": "object",
                "properties": {
                    "billing-account": {
                        "description": "Billing account number used to charge use
of shared fake server.",
                        "type": "string"
                    }
                }
            }
        },
        "update": {
            "parameters": {
                "$schema": "http://json-schema.org/draft-04/schema#",
                "type": "object",
                "properties": {
                    "billing-account": {
                        "description": "Billing account number used to charge use
of shared fake server.",
                        "type": "string"
                    }
                }
            }
        }
    }
},
"service_binding": {
    "create": {
        "parameters": {

```

```

        "$schema": "http://json-schema.org/draft-04/schema#",
        "type": "object",
        "properties": {
            "billing-account": {
                "description": "Billing account number used to charge use
of shared fake server.",
                "type": "string"
            }
        }
    }
}
}, {
    "name": "fake-plan-2",
    "id": "0f4008b5-XXXX-XXXX-XXXX-dace631cd648",
    "description": "Shared fake Server, 5tb persistent disk, 40 max
concurrent connections. 100 async",
    "free": false,
    "metadata": {
        "max_storage_tb": 5,
        "costs": [
            {
                "amount": {
                    "usd": 199.0
                },
                "unit": "MONTHLY"
            },
            {
                "amount": {
                    "usd": 0.99
                },
                "unit": "1GB of messages over 20GB"
            }
        ],
        "bullets": [
            "40 concurrent connections"
        ]
    }
}]
}]
}

```

8. 同步和异步操作

Platforms需要所有API请求提供响应，提供g诶用户快速响应。Service broker作者需让brokers正确响应所有请求，判断是否实现同步或异步响应

- 保证请求操作完成的Brokers，需返回同步响应

- 不能保证操作完成响应的Brokers，需使用异步response

为provision, update或bind操作提供同步response。实际完成后之前，用户可能会疑惑service可能不可用，没有方法知道什么时候。异步响应会让用户知道，操作正在进行中，也可提供操作的更新状态

不同service offering的同步或异步的支持，可能不同。service plan间也不支持

8.1 同步操作

为同步执行一个request，broker需返回不同状态码：

- provision和bind：201 Created
- update、unbind和deprovision：200 OK

支持provision, update, and delete的Brokers需要支持同步响应，若Client提供accepts_incomplete=true时，忽略查询参数

8.2 异步操作

注意：异步操作当前仅支持provision、update和deprovision

- 为让broker返回异步响应，请求需包括查询参数accepts_incomplete=true。若没包括参数或为false，且broker无法满足同步请求(保证操作响应时已完成)，broker需要拒绝请求，状态码422 Unprocessable Entity和以下body

```
{
  "error": "AsyncRequired",
  "description": "This service plan requires client support for
asynchronous service operations."
}
```

- 若包括以上查询参数，并且broker异步执行操作，broker需返回异步响应202 Accepted。response body 需要与broker服务同步请求的一样
- asynchronous响应触发平台市场，轮询GET /v2/service_instances/:instance_id/last_operation直到broker表明请求操作已经成功或失败。Brokers可包括一个status消息，对每个last_operation endpoint的response。提供最终用户的可视化操作过程

Blocking Operations

Brokers不需要并发支持对相同资源的请求。若broker收到请求，并且由于资源有其它活动无法处理。broker需拒绝请求，返回HTTP 422 Unprocessable Entity。响应HTTP body需包括属性description，指明失败原因

响应例子：

```
{
  "description": "Another operation for this service instance is in
progress"
}
```

9. Polling最后操作(last_operation)

当broker为[Provision](#), [Update](#)或[Deprovision], 返回状态码 202 Accepted, platform开始polling `/v2/service_instances/:instance_id/last_operation` endpoint, 获取最后操作状态。broker响应需包括字段 `state`, 也可包含字段 `description`

- `state` 合法值支持: `in progress`, `succeeded` 和 `failed`。只要broker返回 `"state": "in progress"`, platform一直轮询 `last_operation` endpoint。返回 `"state": "succeeded"` 或 `"state": "failed"` 使平台停止polling。 `description` 字段value会传递到 API client, 提供额外细关于操作的进度的节给用户

9.1 请求

- `/v2/service_instances/:instance_id/last_operation`: Get
 - `:instance_id`: 全局唯一非空字符串
- Parameters: 请求提供以下query字符串参数, 作为brokers的hints
- 轮询Interval和Duration: 不同平台client的polling频率和最大时间可能不同。若platform有 max polling duration并且limit达到, 平台需要停止polling, 操作状态需要看成 `failed`

查询字符串字段	Type	描述
service_id	string	catalog中service的ID。若存在, 需非空字符串
plan_id	string	catalog中plan的ID。若存在, 需非空字符串
operation	string	broke提供的操作标识符。When a value for <code>operation</code> is included with asynchronous responses for Provision , Update , and Deprovision requests, the platform MUST provide the same value using this query parameter as a percent-encoded string。若存在需非空字符串

注意: 虽然请求参数 `service_id` 和 `plan_id` 不是强制, Platform需包含所有发送到serice broker 的 `last_operation` 请求

- Headers: 操作定义以下HTTP Headers
 - X-Broker-API-Version*: string。See [API Version Header](#)
- cURL例子

```
$ curl http://username:password@broker-url/v2/service_instances/:instance_id/last_operation
```

9.2 响应

状态码	Description
200 OK	成功处理该请求时需返回。期望body如下
400 Bad Request	若请求被篡改或缺乏必须数据，需返回
410 Gone	用于异步删除操作。The platform MUST consider this response a success and remove the resource from its database。The expected response body is <code>{}</code> 。Returning this while the platform is polling for create or update operations SHOULD be interpreted as an invalid response and the platform SHOULD continue polling.

Responses with any other status code SHOULD be interpreted as an error or invalid response. The platform SHOULD continue polling until the broker returns a valid response or the [maximum polling duration](#) is reached. Brokers MAY use the `description` field to expose user-facing error messages about the operation state; for more info see [Broker Errors](#).

所有response对象需要为合法JSON对象。用于后续版本兼容

成功response的字段

- `state*`: string。合法value为: `progress`、`succeeded`和`failed`
 - `"state": "in progress"`: 平台需要继续polling
 - `"state": "succeeded"` 或 `"state": "failed"`: 导致平台cease polling.
- `description`: string。面向用户消息，显示到平台API client。可用于告诉用户关于操作状态的细节。若在，需要为非空字符串

```
{
  "state": "in progress",
  "description": "Creating service (10% complete)."
}
```

10. Provisioning

Broker收到平台provision请求需确保创建新资源。不同service/plan的provisioning代表内容不同，有一些通用

- 对MySQL service: provisioning可导致一个空的独立database server运行在自己VM或共享database server的一个空schema
- 对非数据Service: provisioning可仅仅为多租户SaaS应用的一个账户

10.1 请求

- `/v2/service_instances/:instance_id`: PUT
 - `:instance_id`: 全局唯一非空字符串。该ID用户在后续请求(bind和deprovision), broker用其关联资源
- 参数
 - `accepts_incomplete`: boolean。true表示marketplace和clients支持异步broker操作。若请求不带该参数, 并且broker仅支持异步provision。broker需拒绝该请求, 返回 `422 Unprocessable Entity`
- 操作定义以下HTTP Headers
 - `X-Broker-API-Version*`: string。See [API Version Header](#)
 - `X-Broker-API-Originating-Identity`: string。See [Originating Identity](#)
- 请求Body

Request字段	Type	描述
<code>service_id*</code>	string	service的ID(来源catalog)。需全局唯一非空字符串
<code>plan_id*</code>	string	plan的ID(来源catalog), for which the service instance has been requested。service中需唯一。非空字符串
<code>context</code>	object	Platform特定ServiceInstance被provisioned的上下文信息。Although most brokers will not use this field, it could be helpful in determining data placement or applying custom business rules。 <code>context</code> will replace <code>organization_guid</code> and <code>space_guid</code> in future versions of the specification - in the interim both SHOULD be used to ensure interoperability with old and new implementations.
<code>organization_guid*</code>	string	Deprecated。使用 <code>context</code> 。The platform GUID for the organization under which the service instance is to be provisioned. Although most brokers will not use this field, it might be helpful for executing operations on a user's behalf. MUST be a non-empty string.
<code>space_guid*</code>	string	Deprecated。使用 <code>context</code> 。The identifier for the project space within the platform organization. Although most brokers will not use this field, it might be helpful for executing operations on a user's behalf. MUST be a non-empty string.
<code>parameters</code>	JSON object	ServiceInstance的配置选项呢。Brokers SHOULD ensure that the client has provided valid configuration parameters and values for the operation.

```
{
  "service_id": "service-id-here",
  "plan_id": "plan-id-here",
  "context": {
    "platform": "cloudfoundry",
    "some_field": "some-contextual-data"
  },
  "organization_guid": "org-guid-here",
  "space_guid": "space-guid-here",
  "parameters": {
    "parameter1": 1,
    "parameter2": "foo"
  }
}
```

- cURL例子

```
$ curl http://username:password@broker-
url/v2/service_instances/:instance_id?accepts_incomplete=true -d '{
  "service_id": "service-id-here",
  "plan_id": "plan-id-here",
  "context": {
    "platform": "cloudfoundry",
    "some_field": "some-contextual-data"
  },
  "organization_guid": "org-guid-here",
  "space_guid": "space-guid-here",
  "parameters": {
    "parameter1": 1,
    "parameter2": "foo"
  }
}' -X PUT -H "X-Broker-API-Version: 2.13" -H "Content-Type:
application/json"
```

10.2 响应

状态码	描述
200 OK	若service instance已存在，并且请求参数等同于已有已有的service instance。期望响应body如下
201 Created	若service instance被provisioned为请求的结果返回。期望响应body如下
202 Accepted	若instance provisioning正在进行返回。触发平台市场去poll Service Instance Last Operation Endpoint 的操作状态。注意一个re-sent的 PUT request若没有完全provisioned，需要返回一个 202 Accepted，并非 200 OK
400 Bad Request	若请求为malformed或丢失强制数据，需要返回
409 Conflict	需要返回，若相同id的service instance已经存在，但属性不同。期望响应body为 <code>{}</code> ，though the description field MAY be used to return a user-facing error message, as described in Broker Errors .
422 Unprocessable Entity	MUST be returned if the broker only supports asynchronous provisioning for the requested plan and the request did not include <code>? accepts_incomplete=true</code> . The expected response body is: <code>{ "error": "AsyncRequired", "description": "This service plan requires client support for asynchronous service operations." }</code> , as described below.

- 任何其它status code被解析为失败。Brokers可包括用户消息到字段 `description`。细节查看[Broker Errors](#).
- 所有响应需要为合法JSON对象 (`{}`)。为了后续兼容
- 成功响应，broker需返回以下字段

Response字段	Type	描述
dashboard_url	string	service instance的Web管理URL。将其定义为service dashboard。URL须包括足够信息，让dashboard标识正在访问的资源(以下例子为 <code>9189kdfsk0vfnku</code>)。注意：需为service instance返回 <code>dashboard_url</code> 的broker，需要返回provision请求的初始化response，service是异步provisioned也一样。若存在，需为非空字符串
operation	string	For asynchronous responses, service brokers MAY return an identifier representing the operation. The value of this field MUST be provided by the platform with requests to the Last Operation endpoint in a percent-encoded query parameter. 若存在，需为非空字符串

```
{
  "dashboard_url": "http://example-dashboard.example.com/9189kdfsk0vfnku",
  "operation": "task_10"
}
```

11. 更新ServiceInstance

通过实现该endpoint, service broker作者可使用户修改已有service instance两个属性: `service plan` 和 `parameters`

- 通过更新ServicePlan, 用户可更新或downgrade他们的ServiceInstance到其它Plans。通过修改参数, 用户可更改特定Service或Plan的配置选项
- 为支持Plan修改, Broker需要为每个service声明支持, 通过包含 `plan_updateable: true` 到 catalog endpoint
- Not all permutations of plan changes are expected to be supported. For example, a service might support upgrading from plan "shared small" to "shared large" but not to plan "dedicated". It is up to the broker to validate whether a particular permutation of plan change is supported. If a particular plan change is not supported, the broker SHOULD return a meaningful error message in response.

11.1 请求

- `/v2/service_instances/:instance_id`: Patch
 - `:instance_id`: 全局唯一非空字符串。之前创建service instance的ID
- 参数
 - `accepts_incomplete`: boolean。true表示marketplace和clients支持异步broker操作。若请求不带该参数, 且broker仅支持异步provision service instance, broker需拒绝该请求, 返回 `422 Unprocessable Entity`
- 操作定义以下Headers
 - `X-Broker-API-Version*`: string。See API Version Header
 - `X-Broker-API-Originating-Identity`: string。See Originating Identity
- 请求Body

Request字段	Type	描述
context	object	service instance被创建的上下文data
service_id*	string	service的ID(来自catalog)。需全局唯一非空字符串
plan_id	string	plan的ID(来自catalog) for which the service instance has been requested。service内需唯一。 If present, MUST be a non-empty string.。 f this field is not present in the request message, then the broker MUST NOT change the plan of the instance as a result of this request.
parameters	JSON object	service instance的配置选项。Brokers SHOULD ensure that the client has provided valid configuration parameters and values for the operation。 If this field is not present in the request message, then the broker MUST NOT change the parameters of the instance as a result of this request.
previous_values	object	修改前的service instance信息
previous_values.service_id	string	Deprecated。 determined to be unnecessary as the value is immutable. ID of the service for the service instance. If present, MUST be a non-empty string.
previous_values.plan_id	string	修改前的plan的ID。 If present, MUST be a non-empty string.
previous_values.organization_id	string	Deprecated。 因为冗余。 Organization for the service instance MUST be provided by platforms in the top-level field <code>context</code> . ID of the organization specified for the service instance. If present, MUST be a non-empty string.
previous_values.space_id	string	Deprecated。 因为冗余。 Space for the service instance MUST be provided by platforms in the top-level field <code>context</code> . ID of the space specified for the service instance. If present, MUST be a non-empty string.

```
{
```



```

    "context": {
      "platform": "cloudfoundry",
      "some_field": "some-contextual-data"
    },
    "service_id": "service-id-here",
    "plan_id": "plan-id-here",
    "parameters": {
      "parameter1": 1,
      "parameter2": "foo"
    },
    "previous_values": {
      "plan_id": "old-plan-id-here",
      "service_id": "service-id-here",
      "organization_id": "org-guid-here",
      "space_id": "space-guid-here"
    }
  }
}

```

- cURL例子

```

$ curl http://username:password@broker-
url/v2/service_instances/:instance_id?accepts_incomplete=true -d '{
  "context": {
    "platform": "cloudfoundry",
    "some_field": "some-contextual-data"
  },
  "service_id": "service-id-here",
  "plan_id": "plan-id-here",
  "parameters": {
    "parameter1": 1,
    "parameter2": "foo"
  },
  "previous_values": {
    "plan_id": "old-plan-id-here",
    "service_id": "service-id-here",
    "organization_id": "org-guid-here",
    "space_id": "space-guid-here"
  }
}' -X PATCH -H "X-Broker-API-Version: 2.13" -H "Content-Type:
application/json"

```

11.2 响应

状态码	描述
200 OK	MUST be returned if the request's changes have been applied。期望想为 <code>{}</code> 。
202 Accepted	MUST be returned if the service instance update is in progress. This triggers the platform marketplace to poll the Last Operation for operation status. Note that a re-sent <code>PATCH</code> request MUST return a <code>202 Accepted</code> , not a <code>200 OK</code> , if the requested update has not yet completed.
400 Bad Request	MUST be returned if the request is malformed or missing mandatory data.
422 Unprocessable entity	MUST be returned if the requested change is not supported or if the request cannot currently be fulfilled due to the state of the service instance (e.g. service instance utilization is over the quota of the requested plan). Brokers SHOULD include a user-facing message in the body; for details see Broker Errors . Additionally, a <code>422 Unprocessable Entity</code> can also be returned if the broker only supports asynchronous update for the requested plan and the request did not include <code>?accepts_incomplete=true</code> ; in this case the expected response body is: <code>{ "error": "AsyncRequired", "description": "This service plan requires client support for asynchronous service operations." }</code> .

- 任何其它状态码的响应被解析为失败。Brokers可包括一个面对用户的信息到字段 `description` field
- 为了后续兼容，所有响应需要为合法JSON对象 (`{}`)
- 成功响应，broker需返回以下字段

Response 字段	Type	描述
operation	string	对异步响应，service brokers需要返回标识符代表操作。平台需提供该字段value，请求 Last Operation endpoint。若存在，需为非空字符串

```
{ "operation": "task_10" }
```

12. Binding

若Catalog endpoint中service或plan定义 `bindable:true`，Platform可请求生成一个Service Binding

- 不是所有services都需为bindable --- some deliver value just from being provisioned。Brokers提供bindable的services需要在Catalog中定义为 `bindable: true`。不提供任何bindable的services的Brokers 不需要实现bind请求的endpoint

12.1 Binding类型

12.1.1 Credentials

- Credentials:** 应用/用户使用service instance时需要的多个信息。若broker支持生成credentials, 需要在请求创建service binding的请求的response, 返回 `credentials`。Credentials需要尽量唯一, 从而可针对每个binding的访问, 在任何时候可被revoked。而不影响该service instance其它bindings的consumer

12.1.2 Log Drain(管道)

- 有个service offering的分类提供日志数据的aggregation、indexing和analysis。为了使用这些services, 产生日志的应用需要stream logs的位置。一个来自于service broker的create binding响应, 需要包括一个 `syslog_drain_url`。平台需要在发送logs到service时, 使用该 `syslog_drain_url` value
- Brokers不能够在创建binding response时, 包括一个 `syslog_drain_url`, 若service的对应的Catalog entry没有包含一个属性 `"requires":["syslog_drain"]`

12.1.3 Route Services

- 有一个service offerings分类, 作为请求到应用的intermediate, 执行功能如rate limiting或授权
- 若平台支持route services, 需要发送routable的地址或endpoint, 包括应用和创建service binding的请求, 使用 `"bind_resource":{"route":"some-address.com"}`。broker可支持使用参数支持配置的address; 暴露这些功能到用户, 需要一个平台支持binding多个routable addresses到先相同service instance
- 若配置service支持该表现, broker需要在创建binding的response返回一个 `route_service_url`, 从而平台知道在哪里proxy应用请求。If the service is deployed such that the network configuration to proxy application requests through instances of the service is managed out-of-band, broker需要在response不返回 `route_service_url`
- Brokers在创建binding response时, 不能包含一个 `route_service_url`。若service的关联Catalog entry, 不包含一个属性 `"requires":["route_forwarding"]`

12.1.4 Volume Services

- services分类: 提供网络存储给应用, 通过应用容器volume mounts。这些services的create binding response需包括 `volume_mounts`
- Brokers MUST NOT include `volume_mounts` in a create binding response if the associated Catalog entry for the service did not include a `"requires":["volume_mount"]` property.

12.2 请求

- `/v2/service_instances/:instance_id/service_bindings/:binding_id`
 - `:instance_id`: 全局唯一非空字符串。需要为之前provisioned的service instance
 - `:binding_id`: 全局唯一非空字符串。该ID用于后续unbind请求, 从而broker用于关联

创建的资源

- Headers: 操作定义以下Headers
 - X-Broker-API-Version*: string。See [API Version Header](#).
 - X-Broker-API-Originating-Identity: string。See [Originating Identity](#).
- 请求Body

请求字段	Type	描述
context	object	ServiceBinding被创建爱你的上下文数据
service_id*	string	catalog中service的ID。需要为非空字符串
plan_id*	string	catalog中plan的ID。需要为非空字符串
app_guid	string	Deprecated in favor of <code>bind_resource.app_guid</code> . GUID of an application associated with the binding to be created. If present, MUST be a non-empty string.
bind_resource	JSON object	A JSON object that contains data for platform resources associated with the binding to be created. See Bind Resource Object for more information.
parameters	JSON object	Configuration options for the service binding. Brokers SHOULD ensure that the client has provided valid configuration parameters and values for the operation.

- Bind Resource Object

The `bind_resource` object: contains platform specific information related to the context in which the service will be used。In some cases the platform might not be able to provide this information at the time of the binding request, therefore the `bind_resource` and its fields are OPTIONAL.

Below are some common fields that MAY be used. Platforms MAY choose to add additional ones as needed.

请求字段	Type	描述
app_guid	string	GUID of an application associated with the binding. For credentials bindings. MUST be unique within the scope of the platform.
route	string	URL of the application to be intermediated. For route services bindings.

`app_guid` represents the scope to which the binding will apply within the platform. For example, in Cloud Foundry it might map to an "application" while in Kubernetes it might map to a "namespace". The scope of what a platform maps the `app_guid` to is platform specific and MAY vary across binding requests.

```
{
  "context": {
    "platform": "cloudfoundry",
    "some_field": "some-contextual-data"
  },
  "service_id": "service-id-here",
  "plan_id": "plan-id-here",
  "bind_resource": {
    "app_guid": "app-guid-here"
  },
  "parameters": {
    "parameter1-name-here": 1,
    "parameter2-name-here": "parameter2-value-here"
  }
}
```

- cURL例子

```
$ curl http://username:password@broker-
url/v2/service_instances/:instance_id/service_bindings/:binding_id -d '{
  "context": {
    "platform": "cloudfoundry",
    "some_field": "some-contextual-data"
  },
  "service_id": "service-id-here",
  "plan_id": "plan-id-here",
  "bind_resource": {
    "app_guid": "app-guid-here"
  },
  "parameters": {
    "parameter1-name-here": 1,
    "parameter2-name-here": "parameter2-value-here"
  }
}' -X PUT
```

12.3 响应

状态码	描述
200 OK	MUST be returned if the binding already exists and the requested parameters are identical to the existing binding. The expected response body is below.
201 Created	MUST be returned if the binding was created as a result of this request. The expected response body is below.
400 Bad Request	MUST be returned if the request is malformed or missing mandatory data.
409 Conflict	MUST be returned if a service binding with the same id, for the same service instance, already exists but with different parameters. The expected response body is <code>{}</code> , though the description field MAY be used to return a user-facing error message, as described in Broker Errors .
422 Unprocessable Entity	MUST be returned if the broker requires that <code>app_guid</code> be included in the request body. The expected response body is: <code>{ "error": "RequiresApp", "description": "This service supports generation of credentials through binding an application only." }</code> .

- 任何其它status code被解析为失败。Brokers可包括用户消息到字段`description`。细节查看[Broker Errors](#).
- 所有响应需要为合法JSON对象 (`{}`)。为了后续兼容
- 成功响应, broker需返回以下字段

Response字段	Type	描述
credentials	object	A free-form hash of credentials that can be used by applications or users to access the service.
syslog_drain_url	string	A URL to which logs MUST be streamed. <code>"requires": ["syslog_drain"]</code> MUST be declared in the Catalog endpoint or the platform MUST consider the response invalid.
route_service_url	string	A URL to which the platform MUST proxy requests for the address sent with <code>bind_resource.route</code> in the request body. <code>"requires": ["route_forwarding"]</code> MUST be declared in the Catalog endpoint or the platform can consider the response invalid.
volume_mounts	array-of-objects	An array of configuration for remote storage devices to be mounted into an application container filesystem. <code>"requires": ["volume_mount"]</code> MUST be declared in the Catalog endpoint or the platform can consider the response invalid.

12.3.1 Volume Mounts对象

Response字段	Type	描述
driver*	string	Name of the volume driver plugin which manages the device.
container_dir*	string	The path in the application container onto which the volume will be mounted. This specification does not mandate what action the platform is to take if the path specified already exists in the container.
mode*	string	"r" to mount the volume read-only or "rw" to mount it read-write.
device_type*	string	A string specifying the type of device to mount. Currently the only supported value is "shared".
device*	device-object	Device object containing device_type specific details. Currently only shared devices are supported.

12.3.2 Device对象

- Currently only shared devices are supported; a distributed file system which can be

mounted on all app instances simultaneously.

Field	Type	描述
volume_id*	string	ID of the shared volume to mount on every app instance.
mount_config	object	Configuration object to be passed to the driver when the volume is mounted.

```
{
  "credentials": {
    "uri": "mysql://mysqluser:pass@mysqlhost:3306/dbname",
    "username": "mysqluser",
    "password": "pass",
    "host": "mysqlhost",
    "port": 3306,
    "database": "dbname"
  }
}
```

```
{
  "volume_mounts": [{
    "driver": "cephdriver",
    "container_dir": "/data/images",
    "mode": "r",
    "device_type": "shared",
    "device": {
      "volume_id": "bc2cleab-05b9-482d-b0cf-750ee07de311",
      "mount_config": {
        "key": "value"
      }
    }
  }]
}
```

13. Unbinding(DELETE)

注意：不提供bindable的services或plans的Brokers不需实现该endpoint

- 当broker收到市场非unbind请求，需删除关联该binding的任何资源。在使用生成credentials的场景，会导致请求该ServiceInstance无法验证通过

13.1 请求

- `/v2/service_instances/:instance_id/service_bindings/:binding_id`: DELETE
 - `:instance_id`: 全局唯一非空字符串。需要为之前provisioned的ServiceInstance的ID

- `:binding_id`: 全局唯一非空字符串。需要为之前provisioned的的ServiceInstance的binding
- Parameters: 请求提供以下查询字符串参数, 为brokers提供有用的hints
 - `service_id`: string. catalog中Service的ID。需要为非空字符串
 - `plan_id`: string. catalog中plan的ID。需要为非空字符串
- Headers: 操作定义以下Headers
 - X-Broker-API-Version*: string. See [API Version Header](#)
 - X-Broker-API-Originating-Identity: string. See [Originating Identity](#).
- cURL例子

```
$ curl 'http://username:password@broker-url/v2/service_instances/:instance_id/service_bindings/:binding_id?service_id=service-id-here&plan_id=plan-id-here' -X DELETE -H "X-Broker-API-Version: 2.13"
```

13.2 响应

状态码	描述
200 OK	若请求binding被删除返回。期望响应body为 <code>{}</code> 。
400 Bad Request	若请求为malformed或丢失强制数据, 返回
410 Gone	若binding不存在返回。期望输出body为 <code>{}</code>

- 任何其它status code被解析为失败。Brokers可包括用户消息到字段 `description`。细节查看[Broker Errors](#).
- 所有响应需要为合法JSON对象 (`{}`)。为了后续兼容
- 成功响应, broker需返回 `{}`

14. Deprovisioning

- 当broker收到市场deprovision求, 需删除provision所有资源。通常所有资源立即reclaimed, 用于后续provisions

14.1 请求

- `(DELETE)/v2/service_instances/:instance_id`: DELETE
 - `:instance_id`: 全局唯一非空字符串。需要为之前provisioned的ServiceInstance的ID
- Parameters: 请求提供以下查询参数, 作为brokers的useful hints

Query-String字段	Type	描述
service_id*	string	catalog中service的ID。需为非空字符串
plan_id*	string	catalog中plan的ID。需为非空字符串
accepts_incomplete	boolean	A value of true indicates that both the marketplace and the requesting client support asynchronous deprovisioning. If this parameter is not included in the request, and the broker can only deprovision a service instance of the requested plan asynchronously, the broker MUST reject the request with a <code>422 Unprocessable Entity</code> as described below.

- Headers: 操作定义以下HTTP Headers
 - X-Broker-API-Version*: string。See [API Version Header](#)
 - X-Broker-API-Originating-Identity: string。See [Originating Identity](#).
- cURL

```
$ curl 'http://username:password@broker-
url/v2/service_instances/:instance_id?accepts_incomplete=true
&service_id=service-id-here&plan_id=plan-id-here' -X DELETE -H "X-Broker-
API-Version: 2.13"
```

14.2 响应

状态码	描述
200 OK	MUST be returned if the service instance was deleted as a result of this request. The expected response body is <code>{}</code> .
202 Accepted	MUST be returned if the service instance deletion is in progress. This triggers the marketplace to poll the Service Instance Last Operation Endpoint for operation status. Note that a re-sent <code>DELETE</code> request MUST return a <code>202 Accepted</code> , not a <code>200 OK</code> , if the delete request has not completed yet.
400 Bad Request	MUST be returned if the request is malformed or missing mandatory data.
410 Gone	MUST be returned if the service instance does not exist. The expected response body is <code>{}</code> .
422 Unprocessable Entity	MUST be returned if the broker only supports asynchronous deprovisioning for the requested plan and the request did not include <code>?accepts_incomplete=true</code> . The expected response body is: <code>{ "error": "AsyncRequired", "description": "This service plan requires client support for asynchronous service operations." }</code> , as described below.

- 任何其它status code被解析为失败。Brokers可包括用户消息到字段 `description`。细节查看[Broker Errors](#).
- 所有响应需要为合法JSON对象 (`{}`)。为了后续兼容
- 成功响应, broker需返回以下字段

Response 字段	Type	Description
operation	string	For asynchronous responses, service brokers MAY return an identifier representing the operation. The value of this field MUST be provided by the platform with requests to the Last Operation endpoint in a percent-encoded query parameter. If present, MUST be a non-empty string.

```
{ "operation": "task_10" }
```

15. Broker Errors

- Broker失败超过以上已经定义的HTTP response codes(如 [Deprovisioning](#)的 `410 Gone`), 需要返回对应HTTP response code(被选择准确反映失败的特性), 和一个包含合法JSON对象的body(不是一个数字)

- 所有响应需要为合法JSON对象 (`{}`)。为后续兼容

对于出错响应，以下字段为valid。其它被忽略。若返回空JSON对象，body为 `{}`，， 会显示给 requester一个通用generic 消息， 包含broker返回HTTP response code

- description: string。有意义error消息， 表达请求失败的原因

```
{
  "description": "Your account has exceeded its quota for service
instances. Please contact support at http://support.example.com."
}
```

16. Orphans

- 平台市场是ServiceInstance和Bindings的**source of true**。市场成功provisioned的ServiceInstances和bindings， ServiceBrokers需要有内容。没有的不应该存在
- Orphans: 市场超时(通常60秒)前， Broker不返回响应。例如若Broker超时前， 没返回响应provision请求， broker会最终成功provisioning一个ServiceInstance， 然后市场认为其失败。导致在Broker端有Orphan service实例
- 为减少orphan的ServiceInstances和bindings， marketplace需尝试删除无法确认成功创建的资源， 且需要保持尽力删除， 直到Broker响应成功

Platforms需要在以下场景初始化orphan mitigation

Broker响应的状态码	响应的Platform翻译	Platform初始化orphan mitigation?
200	Success	No
200 with malformed response	Failure	No
201	Success	No
201 with malformed response	Failure	Yes
All other 2xx	Failure	Yes
408	Timeout failure	Yes
All other 4xx	Broker rejected request	No
5xx	Broker error	Yes
Timeout	Failure	Yes

若平台marketplace遇到提供service或binding内部错误(如保存数据库失败)， 它需要至少发送单个delete或unbind请求到service broker， 避免orphan的创建