**一、部署类(针对linux服务器)**

1、怎么搭建pxe安装环境？(有一个独立的测试网络)

2、怎么使用U盘进行环境安装？

3、如何进行硬raid的配置和清除？

4、linux上如何配置nfs服务器，以及客户端上进行挂载？

5、已经装好的环境如何进行磁盘扩容？

**二、脚本实现**

1. 远程调用实现EXSI服务器中的虚拟机：安装、还原、关机、开启、添加网卡、断开网络连接

**三、高可用和负载均衡的测试从哪些方面进行考虑？**

**四、API测试用例设计**

1. 针对以下API进行用例设计（API见下一页）
2. 使用python实现以下API的调用

**创建主机资产/网络资产**

**1.0.0**

创建一个新的资产。

**POST**

/api/dev

**Header Parameter**

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| Content-Type | String | api请求参数。 |

* [Header 参数](https://10.10.19.216/shterm/resources/docs/rest/index.html#header-examples-Dev-PostHost-1_0_0-0)

{

"Content-Type": "application/json;charset=UTF-8"

}

**参数**

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| name | String | 资产名。  取值范围: 1..200 |
| ip | String | 资产IP。  取值范围: 1..200 |
| type | Number | 资产类别，可选值：0 - 主机，1 - 网络资产 |
| sysType | Object | 资产类型，传参为对象资产类型ID,例如："sysType":{"id":1}。 预定义的资产类型id：主机有Linux,Windows,IBM AIX, HP UX, IBM AS/400的ID分别为"1"，"2"，"4"，"18"。网络资产有Cisco IOS, Huawei Quidway, Juniper NetScreen, H3C Comware,General Network的ID分别为"5"，"6"，"7"，"8"，"9"。 |
| resGroups**可选** | Array | 资产组列表，列表内容为资产组的对象列表，例如："resGroups":[{"id":1},{"id":2}]。 |
| department | Object | 部门，例如："department":{"id":1}。 |
| extra**可选** | Object | 附加信息，例如："extra" : {"extraKey":"extraValue"}。 |
| extInfo**可选** | Object | 资产属性属性值,例如"extInfo":{"1":"bb"}。此处的"1"在系统设置中的资产资产属性 DevExt实体中的"id"必须有，并且"bb"数据类型必须符合DevExt中的"type",是否必须要填写要很据DevExt中"required"决定， DevExt中"extra"字段定义了"aa"取值范围等。 |
| state**可选** | Number | 默认值为0。资产状态：0-活动；1-禁用，例如："state": 0。  取值范围: 0-1 |
| charset | String | 系统编码 ，资产编码类型，缺省从资产类型中获取。编码类型（ISO-8859-1/GB18030/US-ASCII/UTF-8）。 |
| owner**可选** | Object | 责任人，用户实体，例如：{"id":1}。 |
| description**可选** | String | 简要说明, 最大字符串长度512。 |

* [成功回应 201 示例](https://10.10.19.216/shterm/resources/docs/rest/index.html#success-examples-Dev-PostHost-1_0_0-0)

HTTP/1.1 201 Created

{

"id":65

}

**Error 400**

| **名称** | **描述** |
| --- | --- |
| code | 错误码。目前支持错误码包括：   * Max：数字超过允许的最大值； * Min：数字超过允许的最小值； * Size：字符串超过允许的最大长度； * Pattern：提交的值不符合特定正则表达式的匹配规则； * Null：提交的值必须为 null； * NotNull：提交的值必须不能为 null； * Duplicate：数据已经存在，例如新增加的名字已经被使用了； * NotExist：依赖的数据不存在或已经被删除，例如给用户设定某角色，在提交时该角色已经被其他用户删除了； * InUse：由于数据正在被使用或被引用，不允许删除或修改； * Conflict：提交的请求数据，在系统中存在冲突； * TypeMismatch：提交的值不符合要求的类型。此时回应信息中包含的 "requiredType" 字段给出要求的类型。 |
| objectName | 出错的实体名称。 |
| field | 出错的属性或参数名称。 |
| requiredType | 要求的数值类型。 |
| rejectedValue | 错误值。 |
| defaultMessage | 错误信息。 |

* [无效请求回应 400 示例](https://10.10.19.216/shterm/resources/docs/rest/index.html#error-examples-Dev-PostHost-1_0_0-0)

HTTP/1.1 400 Bad Request

{

"code": "Duplicate",

"objectName": "MyObject",

"field": "myField",

"requiredType": "java.lang.Lang",

"rejectedValue": "sampleValue",

"defaultMessage": "errorMsg"

}