

SpringCloud微服务系统API

简介:SpringCloud微服务系统API

HOST:http://localhost:8080

联系人:hyx

Version:1.0

接口路径:/v3/api-docs/zzq

[TOC]

产品管理

添加产品信息

接口地址:/products

请求方式:POST

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

添加新的产品信息到数据库，包括名称、描述和初始库存量。

请求示例:

```
{
  "productId": 0,
  "name": "",
  "description": "",
  "price": 0,
  "productionDate": "",
  "id": 0
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
product	Product	body	true	Product	Product
productId			false	integer(int32)	
name			false	string	
description			false	string	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
price			false	number	
productionDate			false	string(date)	
id			false	integer(int32)	

响应状态:

状态码	说明	schema
201	Created	EntityModelProduct

响应参数:

参数名称	参数说明	类型	schema
name		string	
description		string	
price		number	
productionDate		string(date)	string(date)
id		integer(int32)	integer(int32)
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "",
  "description": "",
  "price": 0,
  "productionDate": "",
  "id": 0,
  "_links": [
    {
      "href": "",
      "hreflang": "",

```

```
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

获取产品信息

接口地址:`/products/{productId}`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`application/hal+json`

接口描述:

根据产品ID获取产品详细信息。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
productId		path	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

删除产品信息

接口地址:/products/{productId}

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

从数据库中删除指定的产品信息。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
productId		path	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新产品信息

接口地址: /products/{productId}

请求方式: PATCH

请求数据类型: application/x-www-form-urlencoded, application/json

响应数据类型: application/hal+json

接口描述:

更新指定产品的信息，包括名称、描述等。

请求示例:

```
{
  "productId": 0,
  "name": "",
  "description": "",
  "price": 0,
  "productionDate": "",
  "id": 0
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
productId		path	true	integer(int32)	
product	Product	body	true	Product	Product
productId			false	integer(int32)	
name			false	string	
description			false	string	
price			false	number	
productionDate			false	string(date)	
id			false	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

出库任务管理

查询出库任务

接口地址:/outbound-tasks

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

按状态或日期范围查询出库任务。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
status		query	false	string	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
startDate		query	false	string	
endDate		query	false	string	
page		query	false	integer(int32)	
size		query	false	integer(int32)	
sort1		query	false	string	
sort2		query	false	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

创建出库任务

接口地址:/outbound-tasks

请求方式:POST

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

为订单创建出库任务。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		query	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新出库任务状态

接口地址:/outbound-tasks/{taskId}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

更新出库任务的状态。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
taskId		path	true	integer(int32)	
status		query	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

订单管理

查询订单

接口地址: /orders

请求方式: GET

请求数据类型: application/x-www-form-urlencoded

响应数据类型: application/hal+json

接口描述:

根据用户或状态查询订单。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
page	Zero-based page index (0..N)	query	false	integer	
size	The size of the page to be returned	query	false	integer	
sort	Sorting criteria in the format: property,(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	query	false	array	
userId		query	false	integer(int32)	
status		query	false	string	

响应状态:

状态码	说明	schema
200	OK	PagedModelEntityModelOrder

响应参数:

参数名称	参数说明	类型	schema
_embedded		PagedModelEntityModelOrder__embedded	PagedModelEntityModelOrder__embedded
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	
page		PageMetadata	PageMetadata

参数名称	参数说明	类型	schema
size		integer(int64)	
totalElements		integer(int64)	
totalPages		integer(int64)	
number		integer(int64)	

响应示例:

```
{
  "_embedded": {
    "orders": [
      {
        "status": "",
        "createdAt": "",
        "updatedAt": "",
        "id": 0,
        "_links": [
          {
            "href": "",
            "hreflang": "",
            "title": "",
            "type": "",
            "deprecation": "",
            "profile": "",
            "name": "",
            "templated": true
          }
        ]
      }
    ]
  },
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ],
  "page": {
    "size": 0,
    "totalElements": 0,
  }
}
```

```
        "totalPages": 0,
        "number": 0
    }
}
```

创建订单

接口地址: /orders

请求方式: POST

请求数据类型: application/x-www-form-urlencoded, application/json

响应数据类型: application/hal+json

接口描述:

创建一个新订单，同时进行库存检查和用户验证。

请求示例:

```
{
  "orderId": 0,
  "customer": {
    "id": 0,
    "name": "15",
    "password": "15",
    "email": "2334579301@qq.com",
    "phone": "133",
    "address": "北京市海淀区北京交通大学"
  },
  "status": "",
  "createdAt": "",
  "updatedAt": "",
  "id": 0
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
order	Order	body	true	Order	Order
orderId			false	integer(int32)	
customer	User		false	User	User
id			false	integer	
name			false	string	
password			false	string	
email			false	string	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
phone			false	string	
address			false	string	
status			false	string	
createdAt			false	string(date-time)	
updatedAt			false	string(date-time)	
id			false	integer(int32)	

响应状态:

状态码	说明	schema
201	Created	EntityModelOrder

响应参数:

参数名称	参数说明	类型	schema
status		string	
createdAt		string(date-time)	string(date-time)
updatedAt		string(date-time)	string(date-time)
id		integer(int32)	integer(int32)
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "status": "",
  "createdAt": "",
  "updatedAt": "",
  "id": 0,
  "_links": [
    {
```

```
        "href": "",
        "hreflang": "",
        "title": "",
        "type": "",
        "deprecation": "",
        "profile": "",
        "name": "",
        "templated": true
      }
    ]
  }
```

取消订单

接口地址: /orders/{orderId}

请求方式: DELETE

请求数据类型: application/x-www-form-urlencoded

响应数据类型: application/hal+json

接口描述:

取消订单，并回滚库存。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		path	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新订单状态

接口地址:/orders/{orderId}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

更新订单的状态。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		path	true	integer(int32)	
status		query	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

订单商品管理

添加商品到订单

接口地址: /order-items

请求方式: POST

请求数据类型: application/x-www-form-urlencoded, application/json

响应数据类型: application/hal+json

接口描述:

为特定订单添加商品。

请求示例:

```
{
  "orderItemId": 0,
  "order": {
    "orderId": 0,
    "customer": {
      "id": 0,
      "name": "15",
      "password": "15",
      "email": "2334579301@qq.com",
      "phone": "133",
      "address": "北京市海淀区北京交通大学"
    },
    "status": "",
    "createdAt": "",
    "updatedAt": "",
    "id": 0
  },
  "product": {
    "productId": 0,
    "name": "",
    "description": "",
    "price": 0,
    "productionDate": "",
    "id": 0
  },
}
```



```
"quantity": 0
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderItem	OrderItem	body	true	OrderItem	OrderItem
orderItemId			false	integer(int32)	
order			false	Order	Order
orderId			false	integer	
customer	User		false	User	User
id			false	integer	
name			false	string	
password			false	string	
email			false	string	
phone			false	string	
address			false	string	
status			false	string	
createdAt			false	string	
updatedAt			false	string	
id			false	integer	
product			false	Product	Product
productId			false	integer	
name			false	string	
description			false	string	
price			false	number	
productionDate			false	string	
id			false	integer	
quantity			false	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

删除订单商品

接口地址:/order-items/{orderId}

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

从订单中删除一个商品，并更新库存。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		path	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
------	------	----	--------

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新订单商品

接口地址:/order-items/{orderId}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

更新订单中某个商品的数量。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		path	true	integer(int32)	
newQuantity		query	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

订单异常管理

查询异常

接口地址:order-exceptions

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

按订单ID或日期查询异常记录。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		query	false	integer(int32)	
date		query	false	string(date)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

记录异常

接口地址:/order-exceptions

请求方式:POST

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

记录订单处理中的异常。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
orderId		query	true	integer(int32)	
description		query	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新异常记录

接口地址:/order-exceptions/{exceptionId}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

更新异常描述或状态。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
exceptionId		path	true	integer(int32)	
newDescription		query	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

获取工作人员

getCollectionResource-worker-get_1

接口地址:WorkerRepository

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

get-worker

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
page	Zero-based page index (0..N)	query	false	integer	
size	The size of the page to be returned	query	false	integer	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
sort	Sorting criteria in the format: property,(asc	desc). Default sort order is ascending. Multiple sort criteria are supported.	query	false	array

响应状态:

状态码	说明	schema
200	OK	PagedModelEntityModelWorker

响应参数:

参数名称	参数说明	类型	schema
_embedded		PagedModelEntityModelWorker__embedded	PagedModelEntityModelWorker__embedded
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	
page		PageMetadata	PageMetadata
size		integer(int64)	
totalElements		integer(int64)	
totalPages		integer(int64)	
number		integer(int64)	

响应示例:

```
{
  "_embedded": {
```



```
"workers": [
  {
    "name": "",
    "password": "",
    "role": "",
    "_links": [
      {
        "href": "",
        "hreflang": "",
        "title": "",
        "type": "",
        "deprecation": "",
        "profile": "",
        "name": "",
        "templated": true
      }
    ]
  }
],
"_links": [
  {
    "href": "",
    "hreflang": "",
    "title": "",
    "type": "",
    "deprecation": "",
    "profile": "",
    "name": "",
    "templated": true
  }
],
"page": {
  "size": 0,
  "totalElements": 0,
  "totalPages": 0,
  "number": 0
}
}
```

postCollectionResource-worker-post

接口地址: /WorkerRepository

请求方式: POST

请求数据类型: application/x-www-form-urlencoded, application/json

响应数据类型: application/hal+json

接口描述:

create-worker

请求示例:

```
{
  "id": 0,
  "name": "",
  "password": "",
  "role": ""
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
workerRequestBody	WorkerRequestBody	body	true	WorkerRequestBody	WorkerRequestBody
id			false	integer(int64)	
name			false	string	
password			false	string	
role			false	string	

响应状态:

状态码	说明	schema
201	Created	EntityModelWorker

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
role		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "",
  "password": "",
  "role": "",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

getItemResource-worker-get

接口地址:/WorkerRepository/{id}

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

get-worker

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelWorker
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	

参数名称	参数说明	类型	schema
role		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "",
  "password": "",
  "role": "",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

putItemResource-worker-put

接口地址:[/WorkerRepository/{id}](#)

请求方式:PUT

请求数据类型:[application/x-www-form-urlencoded](#),[application/json](#)

响应数据类型:[application/hal+json](#)

接口描述:

update-worker

请求示例:

```
{
  "id": 0,
  "name": "",
  "password": "",
  "role": ""
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	
workerRequestBody	WorkerRequestBody	body	true	WorkerRequestBody	WorkerRequestBody
id			false	integer(int64)	
name			false	string	
password			false	string	
role			false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelWorker
201	Created	EntityModelWorker
204	No Content	

响应状态码-200:

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
role		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	

参数名称	参数说明	类型	schema
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "",
  "password": "",
  "role": "",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

响应状态码-201:

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
role		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	

参数名称	参数说明	类型	schema
templated		boolean	

响应示例:

```
{
  "name": "",
  "password": "",
  "role": "",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

deleteItemResource-worker-delete

接口地址:/WorkerRepository/{id}

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:*/*

接口描述:

delete-worker

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	

响应状态:

状态码	说明	schema
204	No Content	
404	Not Found	

响应参数:

暂无

响应示例:

patchItemResource-worker-patch

接口地址:/WorkerRepository/{id}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

patch-worker

请求示例:

```
{
  "id": 0,
  "name": "",
  "password": "",
  "role": ""
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	
workerRequestBody	WorkerRequestBody	body	true	WorkerRequestBody	WorkerRequestBody
id			false	integer(int64)	
name			false	string	
password			false	string	
role			false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelWorker
204	No Content	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
role		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "",
  "password": "",
  "role": "",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

executeSearch-worker-get

接口地址: /WorkerRepository/search/existsByName

请求方式: GET

请求数据类型: application/x-www-form-urlencoded

响应数据类型: application/hal+json

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
username		query	false	string	

响应状态:

状态码	说明	schema
200	OK	
404	Not Found	

响应参数:

暂无

响应示例:

executeSearch-worker-get_1

接口地址:/WorkerRepository/search/existsByNameAndPassword

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
username		query	false	string	
password		query	false	string	

响应状态:

状态码	说明	schema
200	OK	
404	Not Found	

响应参数:

暂无

响应示例:

executeSearch-worker-get_2

接口地址:</WorkerRepository/search/findByName>

请求方式:GET

请求数据类型:[application/x-www-form-urlencoded](#)

响应数据类型:[application/hal+json](#)

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
username		query	false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelWorker
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
role		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "",
  "password": "",
  "role": "",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

库存管理

查询库存

接口地址:/inventory

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

查询特定产品的当前库存量。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
productId		query	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
------	------	----	--------

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新库存

接口地址:/inventory/{productId}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

更新指定产品的库存数量。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
productId		path	true	integer(int32)	
changeInQuantity		query	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

库存警告

接口地址:/inventory/warnings

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

查询库存量低于安全水平的所有产品。

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	

参数名称	参数说明	类型	schema
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

权限管理

登录

接口地址:/auth/login

请求方式:POST

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

登录

请求示例:

```
{
  "username": "15",
  "password": "15"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
loginDomain	LoginDomain	body	true	LoginDomain	LoginDomain

参数名称	参数说明	请求类型	是否必须	数据类型	schema
username	username		true	string	
password	password		true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:POST

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:PUT

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
------	------	----	--------

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	

参数名称	参数说明	类型	schema
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:OPTIONS

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	

参数名称	参数说明	类型	schema
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

登出

接口地址:/auth/logout

请求方式:HEAD

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

登出

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	

参数名称	参数说明	类型	schema
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

注册

接口地址:/auth/register

请求方式:POST

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

注册

请求示例:

```
{
  "username": "Test",
  "password1": "z1111",
  "password2": "z1111",
  "email": "test@example.com",
  "phone": "18810393672"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
registerDomain	RegisterDomain	body	true	RegisterDomain	RegisterDomain
username			true	string	
password1			true	string	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
password2			true	string	
email			false	string	
phone			false	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

入库任务管理

创建入库任务

接口地址:inbound-tasks

请求方式:POST

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

创建一个新的入库任务。

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

删除入库任务

接口地址:/inbound-tasks/{taskId}

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

删除指定ID的入库任务。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
taskId		path	true	integer(int32)	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

更新入库任务状态

接口地址:/inbound-tasks/{taskId}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

更新入库任务的状态。

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
------	------	------	------	------	--------

参数名称	参数说明	请求类型	是否必须	数据类型	schema
taskId		path	true	integer(int32)	
status		query	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

数据库User接口

getCollectionResource-user-get_1

接口地址:/UserRepository

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

get-user

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
page	Zero-based page index (0..N)	query	false	integer	
size	The size of the page to be returned	query	false	integer	
sort	Sorting criteria in the format: property,(asc desc). Default sort order is ascending. Multiple sort criteria are supported.	query	false	array	

响应状态:

状态码	说明	schema
200	OK	PagedModelEntityModelUser

响应参数:

参数名称	参数说明	类型	schema
_embedded		PagedModelEntityModelUser__embedded	PagedModelEntityModelUser__embedded
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	
page		PageMetadata	PageMetadata
size		integer(int64)	
totalElements		integer(int64)	
totalPages		integer(int64)	

参数名称	参数说明	类型	schema
number		integer(int64)	

响应示例:

```
{
  "_embedded": {
    "users": [
      {
        "name": "15",
        "password": "15",
        "email": "2334579301@qq.com",
        "phone": "133",
        "address": "北京市海淀区北京交通大学",
        "_links": [
          {
            "href": "",
            "hreflang": "",
            "title": "",
            "type": "",
            "deprecation": "",
            "profile": "",
            "name": "",
            "templated": true
          }
        ]
      }
    ]
  },
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ],
  "page": {
    "size": 0,
    "totalElements": 0,
    "totalPages": 0,
    "number": 0
  }
}
```

postCollectionResource-user-post

接口地址: /UserRepository

请求方式: POST

请求数据类型: application/x-www-form-urlencoded, application/json

响应数据类型: application/hal+json

接口描述:

create-user

请求示例:

```
{
  "id": 0,
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
userRequestBody	User	body	true	UserRequestBody	UserRequestBody
id			false	integer(int32)	
name			false	string	
password			false	string	
email			false	string	
phone			false	string	
address			false	string	

响应状态:

状态码	说明	schema
201	Created	EntityModelUser

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	

参数名称	参数说明	类型	schema
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

getItemResource-user-get

接口地址:/UserRepository/{id}

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

get-user

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelUser
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
```



```
"address": "北京市海淀区北京交通大学",
"_links": [
  {
    "href": "",
    "hreflang": "",
    "title": "",
    "type": "",
    "deprecation": "",
    "profile": "",
    "name": "",
    "templated": true
  }
]
```

putItemResource-user-put

接口地址:/UserRepository/{id}

请求方式:PUT

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

update-user

请求示例:

```
{
  "id": 0,
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	
userRequestBody	User	body	true	UserRequestBody	UserRequestBody
id			false	integer(int32)	
name			false	string	
password			false	string	

参数名称	参数说明	请求类型	是否必须	数据类型	schema
email			false	string	
phone			false	string	
address			false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelUser
201	Created	EntityModelUser
204	No Content	

响应状态码-200:

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
```

```
    "address": "北京市海淀区北京交通大学",
    "_links": [
      {
        "href": "",
        "hreflang": "",
        "title": "",
        "type": "",
        "deprecation": "",
        "profile": "",
        "name": "",
        "templated": true
      }
    ]
  }
```

响应状态码-201:

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学",
```

```
    "_links": [
      {
        "href": "",
        "hreflang": "",
        "title": "",
        "type": "",
        "deprecation": "",
        "profile": "",
        "name": "",
        "templated": true
      }
    ]
  }
```

deleteItemResource-user-delete

接口地址:/UserRepository/{id}

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:*/*

接口描述:

delete-user

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	

响应状态:

状态码	说明	schema
204	No Content	
404	Not Found	

响应参数:

暂无

响应示例:

patchItemResource-user-patch

接口地址:/UserRepository/{id}

请求方式:PATCH

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

patch-user

请求示例:

```
{
  "id": 0,
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
id		path	true	string	
userRequestBody	User	body	true	UserRequestBody	UserRequestBody
id			false	integer(int32)	
name			false	string	
password			false	string	
email			false	string	
phone			false	string	
address			false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelUser
204	No Content	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	

参数名称	参数说明	类型	schema
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

通过用户名删除用户

接口地址:[/UserRepository/search/deleteByName](#)

请求方式:[GET](#)

请求数据类型:[application/x-www-form-urlencoded](#)

响应数据类型:*/*

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		query	false	string	

响应状态:

状态码	说明	schema
200	OK	
404	Not Found	

响应参数:

暂无

响应示例:

是否存在名字

接口地址:/UserRepository/search/existsByName

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		query	false	string	

响应状态:

状态码	说明	schema
200	OK	
404	Not Found	

响应参数:

暂无

响应示例:

是否存在用户名和密码

接口地址:/UserRepository/search/existsByNameAndPassword

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		query	false	string	
password		query	false	string	

响应状态:

状态码	说明	schema
200	OK	
404	Not Found	

响应参数:

暂无

响应示例:

通过用户名查找

接口地址:/UserRepository/search/findByName

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		query	false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelUser
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
    }
  ]
}
```

```
        "deprecation": "",
        "profile": "",
        "name": "",
        "templated": true
      }
    ]
  }
```

通过用户名和密码查找出一个用户

接口地址: /UserRepository/search/findByNameAndPassword

请求方式: GET

请求数据类型: application/x-www-form-urlencoded

响应数据类型: application/hal+json

接口描述:

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		query	false	string	
password		query	false	string	

响应状态:

状态码	说明	schema
200	OK	EntityModelUser
404	Not Found	

响应参数:

参数名称	参数说明	类型	schema
name		string	
password		string	
email		string	
phone		string	
address		string	
_links		array	Link
href		string	
hreflang		string	
title		string	

参数名称	参数说明	类型	schema
type		string	
deprecation		string	
profile		string	
name		string	
templated		boolean	

响应示例:

```
{
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
  "phone": "133",
  "address": "北京市海淀区北京交通大学",
  "_links": [
    {
      "href": "",
      "hreflang": "",
      "title": "",
      "type": "",
      "deprecation": "",
      "profile": "",
      "name": "",
      "templated": true
    }
  ]
}
```

用户管理

查询所有用户

接口地址:/user

请求方式:GET

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

查询所有用户

请求参数:

暂无

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

创建用户

接口地址:/user

请求方式:POST

请求数据类型:application/x-www-form-urlencoded,application/json

响应数据类型:application/hal+json

接口描述:

创建用户

请求示例:

```
{
  "id": 0,
  "name": "15",
  "password": "15",
  "email": "2334579301@qq.com",
}
```

```
"phone": "133",
"address": "北京市海淀区北京交通大学"
}
```

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
user	User	body	true	User	User
id			false	integer(int32)	
name			false	string	
password			false	string	
email			false	string	
phone			false	string	
address			false	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

查询单个用户

接口地址:`/user/{name}`

请求方式:`GET`

请求数据类型:`application/x-www-form-urlencoded`

响应数据类型:`application/hal+json`

接口描述:

查询单个用户

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		path	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```

删除用户

接口地址:/user/{name}

请求方式:DELETE

请求数据类型:application/x-www-form-urlencoded

响应数据类型:application/hal+json

接口描述:

删除用户

请求参数:

参数名称	参数说明	请求类型	是否必须	数据类型	schema
name		path	true	string	

响应状态:

状态码	说明	schema
200	OK	Result

响应参数:

参数名称	参数说明	类型	schema
success		boolean	
message		string	
data		object	
errors		array	
devMessages		array	
status		string	

响应示例:

```
{
  "success": true,
  "message": "",
  "data": {},
  "errors": [],
  "devMessages": [],
  "status": ""
}
```