



Select a definition

default

Api Documentation 1.0

[Base URL: localhost:8030/]
<http://localhost:8030/v2/api-docs>

Api Documentation

[Terms of service](#)

[Apache 2.0](#)

basic-error-controller Basic Error Controller



eureka-controller Eureka Controller



GET / status

Parameters

Try it out

Name	Description
model object (query)	model { }

//

Responses

Response content type

/

Code	Description
200	OK
	<div>Example ValueModel</div> <div>string</div>
401	Unauthorized
403	Forbidden
404	Not Found

GET

/lastnlastn

Parameters

Try it out

Name	Description
model object (query)	model {}

//

Responses		Response content type
		<div>*/*</div>
Code	Description	
200	OK	
	Example Value	Model
	<code>string</code>	
401	Unauthorized	
403	Forbidden	
404	Not Found	

operation-handler

Operation Handler

>

payment-controller-impl

Payment Controller Impl

▼

GET

/api/v1/payment

findAll

Parameters

Try it out

No parameters

Responses

Response content type

/

Code

Description

Code	Description
200	<div>OK</div> <div>Example Value Model</div> <div><pre>[{ "id": "string", "list": [{ "amount": 0, "buyerId": "string", "id": "string", "price": 0, "sellerId": "string", "stockId": "string" }], "resMessage": { "id": "string", "textMessage": "string" } }]</pre></div>
401	Unauthorized
403	Forbidden
404	Not Found

POST /api/v1/payment Create

Parameters

Try it out

Name	Description
------	-------------

Name	Description
conE ★ required	
object (body)	conE
	<div>Example ValueModel</div> <div><pre>{ "amount": 0, "available": 0, "buyerId": "string", "buyerName": "string", "buyerPhone": "string", "id": "string", "price": 0, "sellerId": "string", "sellerName": "string", "sellerPhone": "string", "stockId": "string"}</pre></div> <div>Parameter content type<div>application/json</div></div>
Responses	Response content type <div>*/*</div>
Code	Description
201	<div>Created</div> <div>Example ValueModel</div> <div><pre>[{ "connectionEntity": { "amount": 0, "available": 0, "buyerId": "string", "buyerName": "string", "buyerPhone": "string", "id": "string", "price": 0, "sellerId": "string", "sellerName": "string", "sellerPhone": "string", "stockId": "string" }, "id": "string", "payment": { "amount": 0, "buyerId": "string", "id": "string", "price": 0, "sellerId": "string", "stockId": "string" }, "resMessage": { "id": "string", "textMessage": "string" } }]</pre></div>

Code	Description
401	Unauthorized
403	Forbidden
404	Not Found

GET

/api/v1/payment/{id} find

Parameters

Try it out

Name	Description
id ★ required	id
string (path)	<div>id - id</div>

Responses

Response content type

/

Code	Description
200	OK

Example Value

Model

```
[
  {
    "id": "string",
    "list": [
      {
        "amount": 0,
        "buyerId": "string",
        "id": "string",
        "price": 0,
        "sellerId": "string",
        "stockId": "string"
      }
    ],
    "resMessage": {
      "id": "string",
      "textMessage": "string"
    }
  }
]
```

Code	Description
401	Unauthorized
403	Forbidden
404	Not Found

PUT

/api/v1/payment/{id} update

Parameters

Try it out

Name	Description
id * required	
string (path)	id
	<div>id - id</div>
payment * required	
object (body)	payment
	<div>Example Value Model</div> <div><pre>{ "amount": 0, "buyerId": "string", "id": "string", "price": 0, "sellerId": "string", "stockId": "string"}</pre></div>
	<div>Parameter content type</div> <div><div>application/json</div></div>

Responses

Response content type

/

Code	Description
------	-------------

Code	Description
200	OK
	<div>Example ValueModel</div> <pre>[{ "id": "string", "list": [{ "amount": 0, "buyerId": "string", "id": "string", "price": 0, "sellerId": "string", "stockId": "string" }], "resMessage": { "id": "string", "textMessage": "string" } }]</pre>
201	Created
401	Unauthorized
403	Forbidden
404	Not Found

DELETE

/api/v1/payment/{id} delete

Parameters

Try it out

Name	Description
<div><div>id * required</div><div>string</div><div>(path)</div></div>	<div>id</div> <div>id - id</div>

Responses

Response content type

/

Code	Description
200	OK
	<div>Example ValueModel</div> <pre>[{ "id": "string", "list": [{ "amount": 0, "buyerId": "string", "id": "string", "price": 0, "sellerId": "string", "stockId": "string" }], "resMessage": { "id": "string", "textMessage": "string" } }]</pre>
204	No Content
401	Unauthorized
403	Forbidden

web-mvc-links-handler

Web Mvc Links Handler



GET

/actuator links

Parameters

Try it out

No parameters

Responses

Response content type

application/vnd.spring-boot.actuator.v3+json

Code	Description
------	-------------

Code	Description
200	OK
<div>Example Value Model</div> <pre>"additionalProp1": { "href": "string", "templated": true }, "additionalProp2": { "href": "string", "templated": true }, "additionalProp3": { "href": "string", "templated": true } }, "additionalProp3": { "additionalProp1": { "href": "string", "templated": true }, "additionalProp2": { "href": "string", "templated": true }, "additionalProp3": { "href": "string", "templated": true } } }</pre>	
401	Unauthorized
403	Forbidden
404	Not Found

Models	▼

```

ConnectionEntity {
    amount          integer($int32)
    available        integer($int32)
    buyerId          string
    buyerName        string
    buyerPhone       string
    id               string
    price            number($double)
    sellerId         string
    sellerName       string
    sellerPhone      string
    stockId          string
}

CreateReturnValue {
    connectionEntity ConnectionEntity {
        amount          integer($int32)
        available        integer($int32)
        buyerId          string
        buyerName        string
        buyerPhone       string
        id               string
        price            number($double)
        sellerId         string
        sellerName       string
        sellerPhone      string
        stockId          string
    }
    id                string
    payment           Payment {
        amount          integer($int32)
        buyerId          string
        id               string
        price            number($double)
        sellerId         string
        stockId          string
    }
    resMessage        Message {
        id               string
        textMessage      string
    }
}

Link {
    href              string
    templated         boolean
}

Map<<string,Link>> {
    < * >:
        Link {
            href          string
            templated      boolean
        }
}

```

```

Message    {
    id          string
    textMessage string
}

ModelAndView {
    empty      boolean
    model      {
    }
    modelMap   {
        < * >:
    }
    reference  boolean
    status     string
    Enum:
        [ CONTINUE, SWITCHING_PROTOCOLS, PROCESSING, CHECKPOINT,
        OK, CREATED, ACCEPTED, NON_AUTHORITATIVE_INFORMATION,
        NO_CONTENT, RESET_CONTENT, PARTIAL_CONTENT, MULTI_STATUS,
        ALREADY_REPORTED, IM_USED, MULTIPLE_CHOICES,
        MOVED_PERMANENTLY, FOUND, MOVED_TEMPORARILY, SEE_OTHER,
        NOT_MODIFIED, USE_PROXY, TEMPORARY_REDIRECT,
        PERMANENT_REDIRECT, BAD_REQUEST, UNAUTHORIZED,
        PAYMENT_REQUIRED, FORBIDDEN, NOT_FOUND, METHOD_NOT_ALLOWED,
        NOT_ACCEPTABLE, PROXY_AUTHENTICATION_REQUIRED,
        REQUEST_TIMEOUT, CONFLICT, GONE, LENGTH_REQUIRED,
        PRECONDITION_FAILED, PAYLOAD_TOO_LARGE,
        REQUEST_ENTITY_TOO_LARGE, URI_TOO_LONG,
        REQUEST_URI_TOO_LONG, UNSUPPORTED_MEDIA_TYPE,
        REQUESTED_RANGE_NOT_SATISFIABLE, EXPECTATION_FAILED,
        I_AM_A_TEAPOT, INSUFFICIENT_SPACE_ON_RESOURCE,
        METHOD_FAILURE, DESTINATION_LOCKED, UNPROCESSABLE_ENTITY,
        LOCKED, FAILED_DEPENDENCY, TOO_EARLY, UPGRADE_REQUIRED,
        PRECONDITION_REQUIRED, TOO_MANY_REQUESTS,
        REQUEST_HEADER_FIELDS_TOO_LARGE,
        UNAVAILABLE_FOR_LEGAL_REASONS, INTERNAL_SERVER_ERROR,
        NOT_IMPLEMENTED, BAD_GATEWAY, SERVICE_UNAVAILABLE,
        GATEWAY_TIMEOUT, HTTP_VERSION_NOT_SUPPORTED,
        VARIANT_ALSO_NEGOTIATES, INSUFFICIENT_STORAGE,
        LOOP_DETECTED, BANDWIDTH_LIMIT_EXCEEDED, NOT_EXTENDED,
        NETWORK_AUTHENTICATION_REQUIRED ]

    view      View {
        contentType      string
    }
    viewName    string
}

Payment    {
    amount      integer($int32)
    buyerId     string
    id          string
    price       number($double)
    sellerId    string
    stockId     string
}

```

```
ReturnValue {
  id      string
  list    [Payment {
    amount      integer($int32)
    buyerId     string
    id          string
    price       number($double)
    sellerId    string
    stockId     string
  }]
  resMessage  Message {
    id          string
    textMessage string
  }
}

View {
  contentType string
}
```