



/api-docs

Explore

# DummyPay API

/api-docs

API documentation for DummyPay

## Servers

http://localhost:8080

## payment-controller



GET /api/payment/{paymentId} Get payment session



Returns the payment session view for the user containing amount, currency, status, merchantName and timestamps.

### Parameters

[Try it out](#)

#### Name Description

<b>paymentId</b> * required integer(\$int64) (path)	Identifier of the payment session
	<input type="text" value="paymentId"/>

### Responses

Code	Description	Links
200	Payment session retrieved successfully	No links

Code	Description	Links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p>Controls Accept header.</p> <p><b>Example Value</b> Schema</p> <pre>{   "amount": 0,   "currency": "string",   "status": "OPEN",   "merchantName": "string",   "createdAt": "2025-12-11T21:27:54.467Z",   "closedAt": "2025-12-11T21:27:54.467Z" }</pre>	
404	Payment not found	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p><b>Example Value</b> Schema</p> <pre>{   "status": 0,   "message": "string",   "path": "string",   "errorCode": "string",   "timestamp": "2025-12-11T21:27:54.468Z" }</pre>	
500	Internal server error	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json</div> <p><b>Example Value</b> Schema</p> <pre>{   "status": 0,   "message": "string",   "path": "string",   "errorCode": "string",   "timestamp": "2025-12-11T21:27:54.468Z" }</pre>	

POST /api/payment Create a new payment session ^

Creates a new payment session. Required fields: amount (number), currency (string), merchant (string), updateWebhook (string). The updateWebhook is where DummyPay will POST webhook notifications when the payment status changes (see the pay/close endpoints for payload details).

## Parameters

[Try it out](#)

No parameters

### Request body required

[application/json](#)

New payment session payload

#### Example Value Schema

```
{  
    "amount": 1099,  
    "currency": "USD",  
    "merchant": "acme-store",  
    "updateWebhook": "https://merchant.example.com/webhooks/payment-update"  
}
```

## Responses

Code	Description	Links
200	Payment session created successfully	<i>No links</i>
	<p>Media type</p> <p><a href="#">application/json</a></p> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{     "sessionId": 123 }</pre>	
400	Validation error in the request payload	<i>No links</i>
	<p>Media type</p> <p><a href="#">application/json</a></p> <p>Example Value Schema</p> <pre>{     "errors": {         "updateWebhook": "updateWebhook is required"     } }</pre>	
500	Unexpected error	<i>No links</i>

Code	Description	Links						
	<p>Media type</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">application/json</div>							
	<p><b>Example Value</b> Schema</p> <pre>{     "status": 0,     "message": "string",     "path": "string",     "errorCode": "string",     "timestamp": "2025-12-11T21:27:54.470Z" }</pre>							
PUT	/api/payment/{paymentId}/pay Mark payment as paid	^						
<p>Set a payment's status to 'paid'. On success, DummyPay issues a webhook POST to the 'updateWebhook' provided when the session was created. Webhook details: method=POST, Content-Type=application/json, payload schema: {"sessionId": number, "status": "PAID"}. Example webhook payload: {"sessionId":123,"status":"PAID"}.</p>								
<p><b>Parameters</b></p>		<b>Try it out</b>						
<table border="1"> <thead> <tr> <th>Name</th><th>Description</th><th></th></tr> </thead> <tbody> <tr> <td><b>paymentId</b> * required integer(\$int64) (path)</td><td>paymentId</td><td></td></tr> </tbody> </table>			Name	Description		<b>paymentId</b> * required integer(\$int64) (path)	paymentId	
Name	Description							
<b>paymentId</b> * required integer(\$int64) (path)	paymentId							
<p><b>Responses</b></p>								
Code	Description	Links						
200	Payment marked as paid	No links						
	<p>Media type</p> <div style="border: 1px solid green; padding: 2px; display: inline-block;">application/json</div>							
	Controls Accept header.							
	<p><b>Example Value</b> Schema</p> <pre>"string"</pre>							

Code	Description	Links
400	Invalid request or invalid payment state  Media type <div style="border: 1px solid black; padding: 2px; display: inline-block;">application/json</div>  <b>Example Value</b> Schema  <pre>{     "status": 0,     "message": "string",     "path": "string",     "errorCode": "string",     "timestamp": "2025-12-11T21:27:54.471Z" }</pre>	No links
404	Payment not found  Media type <div style="border: 1px solid black; padding: 2px; display: inline-block;">application/json</div>  <b>Example Value</b> Schema  <pre>{     "status": 0,     "message": "string",     "path": "string",     "errorCode": "string",     "timestamp": "2025-12-11T21:27:54.472Z" }</pre>	No links
500	Internal server error  Media type <div style="border: 1px solid black; padding: 2px; display: inline-block;">application/json</div>  <b>Example Value</b> Schema  <pre>{     "status": 0,     "message": "string",     "path": "string",     "errorCode": "string",     "timestamp": "2025-12-11T21:27:54.472Z" }</pre>	No links

PUT /api/payment/{paymentId}/close Mark payment as closed ^

Set a payment's status to 'closed'. On success, DummyPay issues a webhook POST to the 'updateWebhook' provided when the session was created. Webhook details: method=POST, Content-

Type=application/json, payload schema: {"sessionId": number, "status": "CLOSED"}. Example webhook payload: {"sessionId":123,"status":"CLOSED"}.

## Parameters

[Try it out](#)

Name	Description
<b>paymentId</b> * required integer(\$int64) (path)	paymentId

## Responses

Code	Description	Links
200	Payment marked as closed	No links
	Media type <b>application/json</b> Controls Accept header.  <b>Example Value</b> Schema  <code>"string"</code>	
400	Invalid request	No links
	Media type <b>application/json</b>  <b>Example Value</b> Schema  <pre>{     "status": 0,     "message": "string",     "path": "string",     "errorCode": "string",     "timestamp": "2025-12-11T21:27:54.474Z" }</pre>	
404	Payment not found	No links
	Media type <b>application/json</b>	

**Links**

Code	Description	Links
	<b>Example Value</b> Schema	
500	<p>Internal server error</p> <p>Media type</p> <p><b>application/json</b></p> <p><b>Example Value</b> Schema</p> <pre>{     "status": 0,     "message": "string",     "path": "string",     "errorCode": "string",     "timestamp": "2025-12-11T21:27:54.474Z" }</pre>	<i>No links</i>

## Schemas



```
ErrorResponse  {  
    status          integer($int32)  
    message        string  
    path           string  
    errorCode      string  
    timestamp      string($date-time)  
}
```

```
NewPaymentSessionDTO {
    amount*
        integer($int32)
        example: 1099
        Amount in the smallest currency unit

    currency*
        string
        example: USD
        ISO currency code

    merchant*
        string
        example: acme-store
        Merchant name

    updateWebhook*
        string
        example: https://merchant.example.com/webhooks/payment-update
        Webhook URL for payment status updates
}
```

```
CreatedSessionResponseDTO {
    description: Response returned when a payment session is created

    sessionId
        integer($int64)
        example: 123
        Identifier of the created session
}
```

```
ValidationErrorResponseDTO {
    description: Validation errors keyed by field name

    errors
        {
            description: Map of field names to validation error messages
            < * >:
                string
                example: {"updateWebhook": "updateWebhook is required"}
                Map of field names to validation error messages
        }
    example: OrderedMap { "updateWebhook": "updateWebhook is required" }
}
```

```
PaymentSessionUserViewDTO {
    amount           integer($int32)
    currency        string
    status          string
    Enum:
        [ OPEN, PAID, CLOSED ]
    merchantName   string
    createdAt       string($date-time)
    closedAt       string($date-time)
}
```