



/v3/api-docs

Explore



Core Banking API

1.0.0 OAS 3.0

[/v3/api-docs](#)

Mock Banking API for Application Modernization Challenge

🎯 Challenge Overview

This API provides endpoints for:

- **JWT Authentication** (with scopes: enquiry, transfer)
- **Fund Transfers** (core requirement)
- **Account Management** (validation, balance inquiry)
- **Transaction History** (bonus feature)

🔒 Authentication

- **Optional for basic functionality**
- **Required for bonus points**
- Use `/authToken` to get JWT token
- Include in `Authorization: Bearer <token>` header

⭐ Bonus Points

- Use JWT tokens for authentication
- Implement proper error handling
- Use account validation endpoints
- Access transaction history

🚀 Quick Start

1. Get a token: `POST /authToken` with `{"username": "alice", "password": "any"}`
2. Transfer funds: `POST /transfer` with transfer details
3. Validate accounts: `GET /accounts/validate/{id}`

Contact Banking Challenge Support

[MIT License](#)

Servers

[http://localhost:8123 - Local Development Server](http://localhost:8123)

Authorize



JWT token generation and validation endpoints



Authentication

POST /authToken Generate JWT Authentication Token



Generates a JWT token with specified claims/scopes. Use ?claim=transfer for transfer operations, default is enquiry.

Parameters

[Try it out](#)

Name	Description
claim string (query)	Claim/scope for the token: 'enquiry' (default) or 'transfer' <i>Default value</i> : enquiry <i>Example</i> : transfer <input type="text" value="transfer"/>

Request body required

application/json

Example Value Schema

```
{  
  "username": "alice",  
  "password": "password123"  
}
```

Responses

Code	Description	Links
200	JWT token successfully generated	No links

Media type

[application/json](#)

Examples

[Token Response](#)

Controls Accept header.

Example Value Schema

```
{  
  "token": "eyJhbGciOiJIUzI1NiJ9...",  
  "username": "alice",  
  "scope": "enquiry",  
  "permissions": "enquiry",  
  "expiresAt": "2024-01-01T12:00:00Z"  
}
```

Code	Description	Links
Example Description		
Token Response		
400	Invalid request body	<i>No links</i>
	Media type	
	<code>*/*</code>	
Example Value Schema		
<pre>{ "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {} }</pre>		

Accounts Account management and validation endpoints - Bonus features

^

GET	/accounts	Get all accounts	Try it out
Returns 100 valid accounts with their balances. JWT token optional but earns bonus points!			
Parameters			
Name	Description		
Authorization <code>JWT token (optional - bonus points if provided)</code> <code>string (header)</code> <input type="text" value="Authorization"/>			
Responses			
Code	Description	Links	
200	List of accounts	<i>No links</i>	
	Media type		
	<code>application/json</code>		
Controls Accept header.			

Code	Description	Links															
	Example Value Schema <pre>[{ "id": "ACC1000", "balance": 1000 }]</pre>																
GET	/accounts/validate/{accountId} Validate account	^															
Checks if the given account ID is valid/active. JWT token optional but earns bonus points!																	
Parameters		Try it out															
<table border="1"> <thead> <tr> <th>Name</th><th>Description</th></tr> </thead> <tbody> <tr> <td>accountId * required <small>string (path)</small></td><td>accountId</td></tr> <tr> <td>Authorization <small>string (header)</small></td><td>JWT token (optional - bonus points if provided) Authorization</td></tr> </tbody> </table>			Name	Description	accountId * required <small>string (path)</small>	accountId	Authorization <small>string (header)</small>	JWT token (optional - bonus points if provided) Authorization									
Name	Description																
accountId * required <small>string (path)</small>	accountId																
Authorization <small>string (header)</small>	JWT token (optional - bonus points if provided) Authorization																
Responses																	
<table border="1"> <thead> <tr> <th>Code</th><th>Description</th><th>Links</th></tr> </thead> <tbody> <tr> <td>200</td><td>OK</td><td>No links</td></tr> <tr> <td colspan="2"> Media type <input type="text" value="/*"/> </td><td></td></tr> <tr> <td colspan="3">Controls Accept header.</td></tr> <tr> <td colspan="2"> Example Value Schema <pre>{ "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {} }</pre> </td><td></td></tr> </tbody> </table>			Code	Description	Links	200	OK	No links	Media type <input type="text" value="/*"/>			Controls Accept header.			Example Value Schema <pre>{ "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {} }</pre>		
Code	Description	Links															
200	OK	No links															
Media type <input type="text" value="/*"/>																	
Controls Accept header.																	
Example Value Schema <pre>{ "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {} }</pre>																	

GET /accounts/balance/{accountId} Get account balance ^

Returns the current balance of a specific account. JWT token optional but earns bonus points!

Parameters

[Try it out](#)

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

accountId * required

string
(path)

accountId

Authorization

string
(header)

JWT token (optional - bonus points if provided)

Authorization

Responses

Code	Description	Links
------	-------------	-------

200 OK

No links

Media type

/

Controls Accept header.

Example Value Schema

```
{
  "additionalProp1": {},
  "additionalProp2": {},
  "additionalProp3": {}
}
```

Transfer Fund transfer operations - Core challenge endpoint ^

POST /transfer  Transfer Funds (Core Challenge Endpoint) 🔒 ^

PRIMARY CHALLENGE ENDPOINT - Transfers funds between accounts.

 **Works without authentication** (minimum requirement)  **JWT token earns bonus points!**

 **Challenge Requirements:**

- Implement HTTP client to call this endpoint
- Handle success/failure responses
- Use valid account IDs (ACC1000-ACC1099)
- Add JWT authentication for bonus points

Bonus Points:

- Include **Authorization: Bearer <token>** header
- Use transfer token (?claim=transfer) for maximum points

Parameters

[Try it out](#)

Name	Description
Authorization <i>(header)</i>	JWT token for bonus points. Format: 'Bearer'. Get token from /authToken endpoint. <i>Example :</i> Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWliOiJhbGljZSIsImV4cCI6MTY0MDk5NTIwMCwic2NvcGUiOiJ0cmFuc2ZlciJ9...

Bearer eyJhbGciOiJIUzI1NiJ9eyJzdWliOiJhbGljZSIsImV4cCI6MTY0MDk5NTIwMCwic2NvcGUiOiJ0cmFuc2ZlciJ9...

Request body required

[application/json](#)

Example Value Schema

```
{
  "fromAccount": "ACC1000",
  "toAccount": "ACC1001",
  "amount": 150
}
```

Responses

Code	Description	Links
200	Transfer processed successfully	<i>No links</i>

Media type

[application/json](#)

Examples

[Successful Transfer](#)

Controls Accept header.

Example Value Schema

```
{
  "transactionId": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "status": "SUCCESS",
  "message": "Transfer completed successfully",
  "fromAccount": "ACC1000",
```

Code	Description	Links
	<pre>"toAccount": "ACC1001", "amount": 100, "bonusPoints": "⭐ JWT Authentication Bonus!"}</pre>	
Example Description		
Successful Transfer		
400	Invalid request - validation errors	<i>No links</i>
<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json</div>		
<p>Example Value</p> <pre>{ "status": "FAILED", "message": "Invalid account format", "errors": ["fromAccount must match pattern ACC[0-9]{4}"] }</pre>		

transaction-controller

^

GET /transactions/history Get transaction history

^

Returns recent transaction history. JWT token required for this bonus endpoint!

Parameters

[Try it out](#)

Name	Description
Authorization	JWT token (required for this endpoint)
string (header)	Authorization
limit	Number of transactions to retrieve (default: 10, max: 20)
integer(\$int32) (query)	Default value : 10
	10

Responses

Code	Description	Links
200	<p>OK</p> <p>Media type</p> <div style="border: 2px solid green; padding: 2px; display: inline-block;">/*</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "additionalProp1": {}, "additionalProp2": {}, "additionalProp3": {}}</pre>	No links

Schemas



```
TransferRequest {  fromAccount* [...]  toAccount* [...]  amount* [...]}
```

```
AuthRequest {  username* [...]  password* [...]}
```

```
Account {  id [...]  balance [...]}
```