s-titan-gym-v1 [Titan Gym System API]

API Endpoint Catalogue

S.No.	Endpoint	Description
1	GET /invoices	Retrieves invoices data from Titan Gym API
2	GET /transactions/{id}	Retrieves general-ledger accounting data from Titan Gym API
3	GET /contractCharges/{id}	Retrieves contract-charges data from Titan Gym API
4	POST/payments	Marks the invoices as PAID in Titan Gym API using invoiceId as Body

API Endpoint Details

GET /invoices

Type	Description
Description	This endpoint retrieves the Invoice data from Titan Gym API
Request Attributes/Payload	URI Params: • NA Query Params: • “$count=true&$filter=date ge {{StartDate}} and date lt {{EndDate}} and isDeleted eq false&$expand=receiver,items($expand=Transactions)”
Response Entities	Invoice Data
Response Example	[{ "number": "12345.67", "invoiceDate": "2025-08-28T10:15:30Z",

```
"clubId": 101,
"isDeleted": false,
"invoiceId": 98765,
"receiver": {
  "name": "Acme Corporation",
  "notes": "Please send payment within 30 days",
  "contactPerson": "Jane Doe"
},
"invoiceLineItems": [
  {
    "itemNumber": 1,
    "name": "Subscription Fee",
    "value": {
      "net": 100.0,
      "gross": 119.0,
      "vat": 19.0
    }
  },
  {
    "itemNumber": 2,
    "name": "Setup Fee",
    "value": {
      "net": 50.0,
      "gross": 59.5,
      "vat": 9.5
    }
  }
],
"transactions": [
  {
    "transactionId": 1111,
    "contractChargeId": 2001
  },
  {
    "transactionId": 1112,
    "contractChargeId": 2002
```

	}] }]
Connectors	HTTPS Listener (For Inbound) HTTPS Requester (For outbound)
Assumptions	The Daily invoice data is limited to maximum 60-70 invoices, regardless of the number of associated line items contained within each invoice (This data is received from the Project Team)
Integration Pattern	Synchronous Request-Reply
Target System	Titan Gym
Target System API endpoint	GET “https://titansportuat.titangym.com/Api/v2.2/odata/CompanyInvoi ces” (dev/uat)
Queue/Topic to publish	NA
Implementation Guidlines	1. Define the RAML specification for this endpoint in alignment with the provided field-level mapping and query parameter details. 2. Create the flow for this endpoint with necessary processors, components and ensure proper error handling. 3. Log start, flow, end, error information using standard error handling wherever required to ensure traceability. 4. Extract and save the incoming query parameters from the request payload into the variables. These parameters will serve as dynamic filters for the downstream invocation of the Titan Gym API to retrieve invoice data. 5. Invoke Titan Gym API to retrieve Invoice data, this will be a GET call based on the filters (queryParams) mentioned in Request Attributes section. Also, Implement Retry logic for unsuccessful GET call.

	6. Map and Transform the response payload received from the Titan Gym API into a structured JSON format as defined by the data mapping rules. This transformed response should be returned to the upstream calling API i.e. **p-finance-mgmt-events-v1.** (Use DataWeave to transform Titan Gym's API response into your canonical model expected by p- finance-mgmt-events-v1). 7. Create MUnit test cases for all key scenarios and flows. The Test coverage should be greater than 80%.
Data Translations	Data Mapping
Error Handling	Error should be transformed to the standard error object and passed back to the caller. Error logging should be logged to trace in Log Consolidation tool
Inbound Security	Client ID Enforcement
Outbound Security	Client ID Enforcement

Data Mapping

CDM	CDM Data Type	Vendor API Fields	Comments
number	string	number	
invoiceDate	datetime	date	
clubId	integer	clubId	
isDeleted	boolean	isDeleted	
invoiceId	integer	id	
receiver.name	string	receiver.name	

receiver.notes	string	receiver.notes	
receiver.contactPerson	string	receiver.contactPerson	
invoice-line-items[].itemNumber	integer	item[]. itemNumber	
invoice-line-items[].name	string	item[]. name	
invoice-line-items[].value.net	number	item[].value.net	
invoice-line-items[].value.gross	number	item[].value.gross	
invoice-line-items[].value.vat	number	item[].value.vat	
transactions[].transactionId	integer	transactions [].transactionId	
transactions[].contractChargeId	integer	transactions[].contractChargeId	

GET /transactions/{id}

Type	Description
Description	This endpoint retrieves the Transaction data associated for each invoice using 'transactionPaymentId' from Titan Gym API
Request Attributes/Payload	URI Params: • transactionId Query Params: • $ expand=accountingDetails
Response Entities	Transaction Data for Invoice

Response Example	{
	  "clubId": 101,
	  "description": "Monthly Hosting Fee",
	  "amount": {
	    "net": 500.0,
	    "gross": 595.0,
	    "vat": 95.0
	  },
	  "accountingDetails": [
	    {
	      "accountNumber": "4000-999-99",
	      "ledger": "Mscla Gl",
	      "split": 70.0
	    },
	    {
	      "accountNumber": "4001-666-66",
	      "ledger": "Mscau Gl",
	      "split": 30.0
	    }
	  ]
	}

Connectors	HTTPS Listener (For Inbound) HTTPS Requester (For outbound)

Assumptions	This endpoint is invoked once per line item (inputNumber) from the CompanyInvoices data. For example, an invoice with 100 line items will trigger 100 separate calls. The Titan Gym API currently does not support batch processing for multiple line items in a single request.
Integration Pattern	Synchronous Request-Reply
Target System	Titan Gym
Target System API endpoint	GET https://titansportuat.titangym.com/Api/v2.2/odata/Transactions/{id} (dev/uat)
Queue/Topic to publish	NA

Implementation Guidelines	1. Define the RAML specification for this endpoint in alignment with the provided field-level mapping and query parameter details. 2. Create the flow for this endpoint with necessary processors, components and ensure proper error handling. 3. Log start, flow, end, error information using standard error handling wherever required to ensure traceability. 4. Extract and save the incoming query parameters from the request payload into the variables. These parameters will serve as dynamic filters for the downstream invocation of the Titan Gym API to retrieve invoice data. 5. Invoke Titan Gym API to retrieve transaction data per invoice using 'transactionPaymentId' , this will be a GET call based on the filters (queryParams) mentioned in the Request Attributes section. Also, Implement Retry logic for unsuccessful GET call. 6. Map and Transform the response payload received from the Titan Gym API into a structured JSON format as defined by the data mapping rules. This transformed response should be returned to the upstream calling API i.e. **p-finance-mgmt-events-v1.** (Use DataWeave to transform Titan Gym's API response into your canonical model expected by p- finance-mgmt-events-v1). 7. Create MUnit test cases for all key scenarios and flows. The Test coverage should be greater than 80%.

Data Translations	Data Mapping
Error Handling	Error should be transformed to the standard error object and passed back to the caller. Error logging should be logged to trace in Log Consolidation tool
Inbound Security	Client ID Enforcement
Outbound Security	Client ID Enforcement

Data Mapping

CDM	CDM Type	Vendor API Fields	Comments
clubId	number	clubId	
description	string	description	
amount.net	number	amount.net	
amount.gross	number	amount.gross	
amount.vat	number	amount.vat	
accountingDetails[].accountNumber	string	accountingDetails[].accountNumber	
accountingDetails[].ledger	string	accountingDetails[].ledger	
accountingDetails[].split	number	accountingDetails[].split	

GET /contractCharges/{id}

Type	Description
Description	This endpoint retrieves the Contract Charges for a particular Invoice using ‘contractChargeId’ from Titan Gym API
Request Attributes/Payload	URI Params: - contractChargeId Query Params: - $expand=contract ($select = clubId), accountingDetails
Response Entities	Contract-Charges Data

Response Example	{
	  "clubId": 101,
	  "description": "Monthly Hosting Fee",
	  "amount": {
	    "net": 500.0,
	    "gross": 595.0,
	    "vat": 95.0
	  },
	  "accountingDetails": [
	    {
	      "accountNumber": "4000",
	      "ledger": "Main",
	      "split": 70.0
	    },
	    {
	      "accountNumber": "4001",
	      "ledger": "Secondary",
	      "split": 30.0
	    }
	  ]
	}

Connectors	HTTPS Listener (For Inbound) HTTPS Requester (For outbound)
Assumptions	This endpoint is invoked once per line item (inputNumber) from the CompanyInvoices data. For example, an invoice with 100 line items will trigger 100 separate calls. The Titan Gym API currently does not support batch processing for multiple line items in a single request.
Integration Pattern	Synchronous Request-Reply
Target System	Titan Gym
Target System API endpoint	GET https://titansportuat.titangym.com/Api/v2.2/odata/ContractCharges/{id} (dev/uat)
Queue/Topic to publish	NA

Implementation Guidelines	1. Define the RAML specification for this endpoint in alignment with the provided field-level mapping and query parameter details. 2. Create the flow for this endpoint with necessary processors, components and ensure proper error handling. 3. Log start, flow, end, error information using standard error handling wherever required to ensure traceability. 4. Extract and save the incoming query parameters from the request payload into the variables. These parameters will serve as dynamic filters for the downstream invocation of the Titan Gym API to retrieve invoice data. 5. Invoke Titan Gym API to retrieve contract-charges data, this will be a GET call based on the filters (queryParams) mentioned in Request Attributes section. Also, Implement Retry logic for unsuccessful GET call. 6. Map and Transform the response payload received from the Titan Gym API into a structured JSON format as defined by the data mapping rules. This transformed response should be returned to the upstream calling API i.e. **p-finance-mgmt-events-v1.** (Use DataWeave to transform Titan Gym's API response into your canonical model expected by p- finance-mgmt-events-v1). 7. Create MUnit test cases for all key scenarios and flows. The Test coverage should be greater than 80%.

Data Translations	Data Mappings
Error Handling	Error should be transformed to the standard error object and passed back to the caller. Error logging should be logged to trace in Log Consolidation tool
Inbound Security	Client ID Enforcement
Outbound Security	Client ID Enforcement

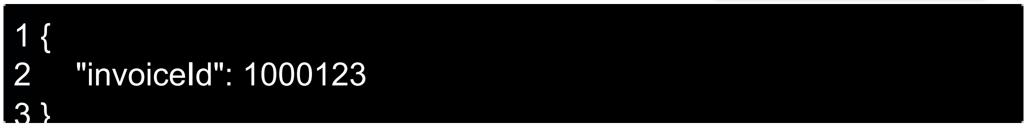
Data Mapping

CDM	CDM Type	Vendor API Fields	Comments
clubId	number	clubId	
description	string	description	
amount.net	number	amount.net	
amount.gross	number	amount.gross	
amount.vat	number	amount.vat	
accountingDetails[].accountNumber	string	accountingDetails[].accountNumber	
accountingDetails[].ledger	string	accountingDetails[].ledger	
accountingDetails[].split	number	accountingDetails[].split	

POST /payments

Type	Description
Description	This endpoint marks the Payments (against each distinct invoiceId) as PAID to Titan Gym API using POST call.

Request Attributes/Payload	URIParam - invoiceId Query Params: NA Body : NA
Response Entities	200 OK
Connectors	HTTPS Listener (For Inbound) HTTPS Requester (For outbound)
Assumptions	The Daily Invoice-Payment data is limited to maximum 150 line-items (This data is received from the Project Team)
Integration Pattern	Synchronous Request-Reply
Target System	Titan Gym

Target System API endpoint	POST
	https://titansportuat.titangym.com/Api/v2/Invoices/MarkCompanyInvoiceAsPaid (dev/uat)
	Request Body:
	
	`{`
	`    "invoiceId": 1000123`
	`}`

Queue/Topic to publish	NA
Implementation Guidelines	1. Define the RAML specification for this endpoint in alignment with the provided field-level mapping and URI parameter details. 2. Create the flow for this endpoint with necessary processors, components and ensure proper error handling. 3. Log start, flow, end, error information using standard error handling wherever required to ensure traceability. 4. Extract and save the incoming URI parameter (invoiceId) from the input request into a variable. This invoiceId will be sent as body during the POST call to Titan Gym API. 5. Send a POST request to Titan Gym API with the invoiceId as in

	input body as mentioned in Target System API endpoint section. Place the HTTP Request connector inside an Until Successful scope and wrap it inside a Try scope for better error handling i.e. for only interim issues like CONNECTIVITY retry via Until Successful upto 3 times, for the rest, catch them and propagate them to global-error-handler. 6. Create MUnit test cases for all key scenarios and flows. The Test coverage should be greater than 80%.
Data Translations	Data Mapping
Error Handling	Error should be transformed to the standard error object and passed back to the caller. Error logging should be logged to trace in Log Consolidation tool
Inbound Security	Client ID Enforcement
Outbound Security	Client Id and Client Secret as Header

Data Mapping

S.No.	Input Field	CDM Field	Vendor API field	Notes
1	invoiceId	invoiceId	invoiceId	type is ‘Number’