

API STRUCTURE - Swagger

INDEX

[Main screen](#)

[EndPoints](#)

[Schemas](#)

[LOGIN SCREEN details](#)

[POST. /api/login/create](#)

[GET. /api/login/{cardId}](#)

[GET. /api/login/{cardId}/{cardPassCode}](#)

[WELCOME SCREEN details](#)

[GET. /api/{cardId}/welcome/logout/{cardIdSecondTime}](#)

[GET. /api/{cardId}/welcome/order/allProducts](#)

[POST. /api/{cardId}/welcome/order/createOrder](#)

[POST. /api/{cardId}/welcome/order/createOrderProduct](#)

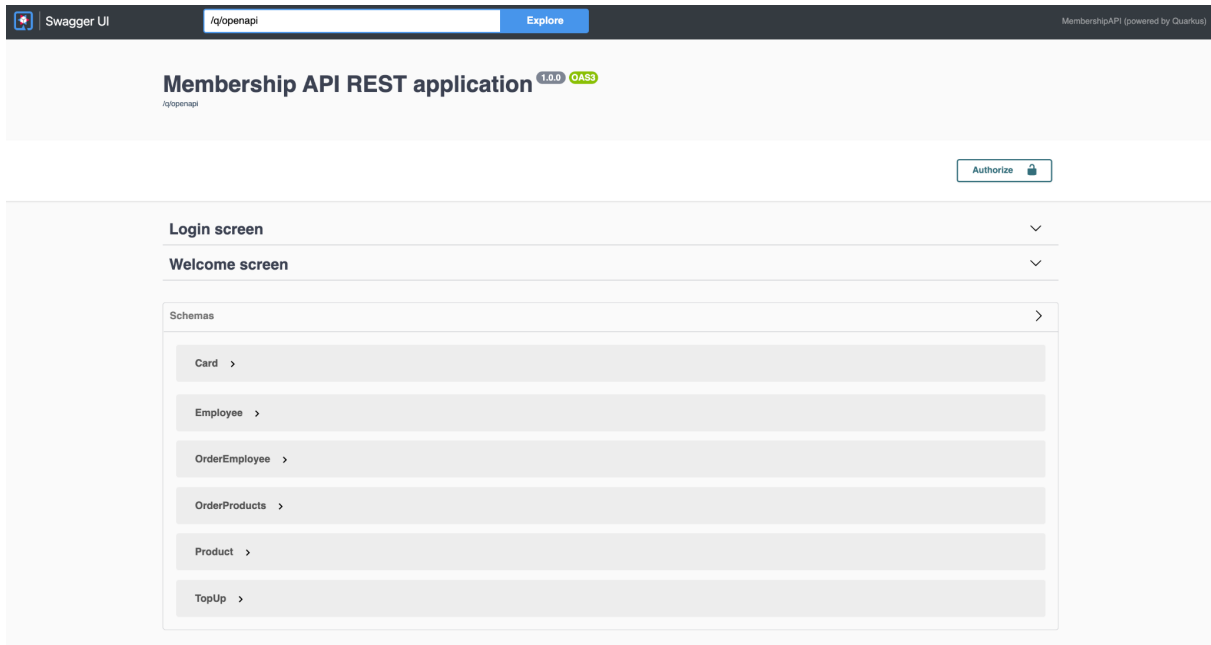
[PUT. /api/{cardId}/welcome/order/{orderEmployeeId}/payment](#)

[PUT. /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}](#)

[DELETE. /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}](#)

[POST. /api/{cardId}/welcome/topUp](#)

Main screen



EndPoints

Login screen			^
POST	/api/login/create	Registers the card and stores Employee details	v
GET	/api/login/{cardId}	Checks time since last interaction and logs out after N minutes of inactivity	v
GET	/api/login/{cardId}/{cardPassCode}	Checks if user is registered	v
Welcome screen			^
GET	/api/{cardId}/welcome/logout/{cardIdSecondTime}	Gets users cardId input and logs out	v
GET	/api/{cardId}/welcome/order/allProducts	Retrieves all the products	v
POST	/api/{cardId}/welcome/order/createOrder	Registers the card and stores Employee details	v
POST	/api/{cardId}/welcome/order/createOrderProduct	Creates an OrderProduct to track selected products	v
PUT	/api/{cardId}/welcome/order/{orderEmployeeId}/payment	Takes care of the payment and updates the card's balance if successful	v
PUT	/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}	Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.	v
DELETE	/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}	Deletes orderProduct and updates Order total	v
POST	/api/{cardId}/welcome/topUp	Top ups the card	v

Schemas

Schemas

>

```
Card {
  cardId
  employee
  cardPassCode integer($int32)
}
```

string

Employee {

cardId string

employeeId string

employeeName string

employeeEmail string

employeeMobileNumber string

}

```
Employee {
  cardId string
  employeeId string
  employeeName string
  employeeEmail string
  employeeMobileNumber string
}
```

```
OrderEmployee {
  orderEmployeeId integer($int64)
  card
  orderTotal number($double)
}
```

Card {

cardId string

employee

Employee {

cardId string

employeeId string

employeeName string

employeeEmail string

employeeMobileNumber string

}

cardPassCode integer(\$int32)

}

```
OrderProducts {
  orderProductsId integer($int64)
  product
  orderEmployee
  orderProductsPrice number($double)
}
```

Product {

productId integer(\$int64)

productName string

productPrice number(\$double)

}

OrderEmployee {

orderEmployeeId integer(\$int64)

card

Card {

cardId string

employee

Employee > {...}

cardPassCode integer(\$int32)

}

orderTotal number(\$double)

}

```
Product {
  productId integer($int64)
  productName string
  productPrice number($double)
}
```

```
TopUp {
  card
  topUpAmount number($double)
}
```

Card {

cardId string

employee

Employee {

cardId string

employeeId string

employeeName string

employeeEmail string

employeeMobileNumber string

}

cardPassCode integer(\$int32)

}

LOGIN SCREEN details

POST. /api/login/create

Login screen

POST

/api/login/create

Registers the card and stores Employee details

Parameters

No parameters

Try it out

Request body

application/json

.JSON example input: { "cardId": "r7jTG7dqBy5wGO6A", "employee": { "cardId": "r7jTG7dqBy5wGO6A", "employeeId": "testId", "employeeName": "testName", "employeeEmail": "test@test.com", "employeeMobileNumber": "00000 000000" }, "cardPassCode": 1111 }

Example Value | Schema

```
{
  "cardId": "string",
  "employee": {
    "cardId": "string",
    "employeeId": "string",
    "employeeName": "string",
    "employeeEmail": "string",
    "employeeMobileNumber": "string"
  },
  "cardPassCode": 0
}
```

Responses

Code	Description	Links
201	Card registration successful	No links
500	Internal Server Error	No links

GET

/api/login/{cardId}

Checks time since last interaction and logs out after N minutes of inactivity

GET

/api/login/{cardId}/{cardPassCode}

Checks if user if registered

GET. /api/login/{cardId}

Login screen

POST

/api/login/create

Registers the card and stores Employee details

GET

/api/login/{cardId}

Checks time since last interaction and logs out after N minutes of inactivity

Parameters

Try it out

Name	Description
cardId <small>required</small> string (path)	input example: r7jTG7dqBy5wGO4L <input type="text" value="cardId"/>

Responses

Code	Description	Links
200	OK	No links

GET. /api/login/{cardId}/{cardPassCode}

Login screen

POST /api/login/create Registers the card and stores Employee details

GET /api/login/{cardId} Checks time since last interaction and logs out after N minutes of inactivity

GET /api/login/{cardId}/{cardPassCode} Checks if user is registered

Parameters

Cancel

Name	Description
cardId * required string (path)	input example: r7JTG7dqBy5wGQ4L <input type="text" value="cardId"/>
cardPassCode * required integer (\$int32) (path)	input example: 1234 <input type="text" value="cardPassCode"/>

Execute

Responses

Code	Description	Links
200	Card is registered	No links
404	Card is not found	No links
500	Internal Server Error	No links

WELCOME SCREEN details

GET. /api/{cardId}/welcome/logout/{cardIdSecondTime}

Welcome screen

GET /api/{cardId}/welcome/logout/{cardIdSecondTime} Gets users cardId input and logs out

Parameters

Name	Description
cardId * required string (path)	<input type="text" value="cardId"/>
cardIdSecondTime * required string (path)	<input type="text" value="cardIdSecondTime"/>

Execute

Responses

Code	Description	Links
201	Log Out	No links
500	Internal Server Error	No links

GET /api/{cardId}/welcome/order/allProducts Retrieves all the products

POST /api/{cardId}/welcome/order/createOrder Registers the card and stores Employee details

POST /api/{cardId}/welcome/order/createOrderProduct Creates an OrderProduct to track selected products

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/payment Takes care of the payment and updates the card's balance if successful

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Deletes orderProduct and updates Order total

POST /api/{cardId}/welcome/topUp Top ups the card

GET. /api/{cardId}/welcome/order/allProducts

Welcome screen

GET /api/{cardId}/welcome/logout/{cardIdSecondTime} Gets users cardId input and logs out

GET /api/{cardId}/welcome/order/allProducts Retrieves all the products

Parameters

No parameters

Execute

Responses

Code	Description	Links
200	Products Found <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div>Example Value</div><pre>[{"productId": 1, "productName": "Tuna sandwich", "productPrice": 2}, {"productId": 2, "productName": "Orange juice", "productPrice": 1}]</pre></div>	No links
500	Internal Server Error	No links

POST /api/{cardId}/welcome/order/createOrder Registers the card and stores Employee details

POST /api/{cardId}/welcome/order/createOrderProduct Creates an OrderProduct to track selected products

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/payment Takes care of the payment and updates the card's balance if successful

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Deletes orderProduct and updates Order total

POST /api/{cardId}/welcome/topUp Top ups the card

POST. /api/{cardId}/welcome/order/createOrder

Welcome screen

GET /api/{cardId}/welcome/logout/{cardIdSecondTime} Gets users cardId input and logs out

GET /api/{cardId}/welcome/order/allProducts Retrieves all the products

POST /api/{cardId}/welcome/order/createOrder Registers the card and stores Employee details

Parameters

No parameters

Request body

application/json

Not all the fields are required, so use this JSON example input to create an empty order: {
"card": {
 "cardId": "r7jTG7dqBySwG04L",
 "employee": {
 "cardId": "r7jTG7dqBySwG04L"
 }
}}

{
 "orderEmployeeId": 0,
 "card": {
 "cardId": "string",
 "employee": {
 "cardId": "string",
 "employeeId": "string",
 "employeeName": "string",
 "employeeEmail": "string",
 "employeeMobileNumber": "string"
 },
 "cardPassCode": 0
 },
 "orderTotal": 0
}
}

Execute

Responses

Code	Description	Links
201	Order created <div>Media type application/json Controls Accept header. Example Value { "orderEmployeeId": 1, "card": { "cardId": "r7jTG7dqBySwG04L", "employee": { "cardId": "r7jTG7dqBySwG04L" }, "cardPassCode": 0 }, "orderTotal": 0 }</div>	

 No links || 500 | Internal Server Error | No links |

POST /api/{cardId}/welcome/order/createOrderProduct Creates an OrderProduct to track selected products

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/payment Takes care of the payment and updates the card's balance if successful

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Deletes orderProduct and updates Order total

POST. /api/{cardId}/welcome/order/createOrderProduct

Welcome screen

GET /api/{cardId}/welcome/logout/{cardIdSecondTime} Gets users cardId input and logs out

GET /api/{cardId}/welcome/order/allProducts Retrieves all the products

POST /api/{cardId}/welcome/order/createOrder Registers the card and stores Employee details

POST /api/{cardId}/welcome/order/createOrderProduct Creates an OrderProduct to track selected products

Parameters

No parameters

Request body

application/json

Input example: { "product": { "productId": 2 }, "orderEmployee": { "orderEmployeeId": 1 } }

```
{  "orderProductsId": 0,  "product": {    "productId": 0,    "productName": "string",    "productPrice": 0  },  "orderEmployee": {    "orderEmployeeId": 0,    "card": {      "cardId": "string",      "employee": {        "cardId": "string",        "employeeId": "string",        "employeeName": "string",        "employeeEmail": "string",        "employeeMobileNumber": "string"      },      "cardPassCode": 0    }  } }
```

Execute

Responses

Code	Description	Links
201	OrderProduct created	No links
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value</div> <div><pre>{ "orderProductsId": 1, "product": { "productId": 1, "productPrice": 0 }, "orderEmployee": { "orderEmployeeId": 1, "orderTotal": 0 }, "orderProductsPrice": 2 }</pre></div>	
500	Internal Server Error	No links

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/payment Takes care of the payment and updates the card's balance if successful

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Deletes orderProduct and updates Order total

PUT. /api/{cardId}/welcome/order/{orderEmployeeId}/payment

Welcome screen

GET /api/{cardId}/welcome/logout/{cardIdSecondTime} Gets users cardId input and logs out

GET /api/{cardId}/welcome/order/allProducts Retrieves all the products

POST /api/{cardId}/welcome/order/createOrder Registers the card and stores Employee details

POST /api/{cardId}/welcome/order/createOrderProduct Creates an OrderProduct to track selected products

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/payment Takes care of the payment and updates the card's balance if successful

Parameters

Cancel

Name	Description
cardId * required string (path)	Input example: r7JTg7dqBy5wGO4L <input type="text" value="cardId"/>
orderEmployeeId * required integer (\$int64) (path)	Input example: 1 <input type="text" value="orderEmployeeId"/>

Execute

Responses

Code	Description	Links
200	Payment successful	No links
400	Insufficient funds - Bad request	No links
500	Internal Server Error	No links

PUT /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE /api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId} Deletes orderProduct and updates Order total

POST /api/{cardId}/welcome/topUp Top ups the card

PUT.
/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Welcome screen

GET

/api/{cardId}/welcome/logout/{cardIdSecondTime}

Gets users cardId input and logs out

GET

/api/{cardId}/welcome/order/allProducts

Retrieves all the products

POST

/api/{cardId}/welcome/order/createOrder

Registers the card and stores Employee details

POST

/api/{cardId}/welcome/order/createOrderProduct

Creates an OrderProduct to track selected products

PUT

/api/{cardId}/welcome/order/{orderEmployeeId}/payment

Takes care of the payment and updates the card's balance if successful

PUT

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

Parameters

Cancel

Name	Description
<div>orderEmployeeId</div> <div><div>integer(\$int64)</div><div>(path)</div></div> <div><div>*</div>required</div>	<div>Input example: 1</div> <div>orderEmployeeId</div>
<div>orderProductsId</div> <div><div>integer(\$int64)</div><div>(path)</div></div> <div><div>*</div>required</div>	<div>Input example: 1</div> <div>orderProductsId</div>

Execute

Responses

Code	Description	Links
200	<div>OrderProduct updated</div> <div><div>Media type</div><div>text/plain</div><div>Controls Accept header.</div><div>Example Value</div><div>OrderProduct updated in order</div></div>	No links
500	<div>Internal Server Error</div>	No links

DELETE

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Deletes orderProduct and updates Order total

POST

/api/{cardId}/welcome/topUp

Top ups the card

DELETE.

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Welcome screen

GET

/api/{cardId}/welcome/logout/{cardIdSecondTime}

Gets users cardid input and logs out

GET

/api/{cardId}/welcome/order/allProducts

Retrieves all the products

POST

/api/{cardId}/welcome/order/createOrder

Registers the card and stores Employee details

POST

/api/{cardId}/welcome/order/createOrderProduct

Creates an OrderProduct to track selected products

PUT

/api/{cardId}/welcome/order/{orderEmployeeId}/payment

Takes care of the payment and updates the card's balance if successful

PUT

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Deletes orderProduct and updates Order total

Parameters

Cancel

Name	Description
<div><div>orderEmployeeId</div><div><div>integer(\$int64)</div><div>(path)</div></div><div><div>*</div>required</div></div> <div>Input example: 1</div> <div>orderEmployeeId</div>	
<div><div>orderProductsId</div><div><div>integer(\$int64)</div><div>(path)</div></div><div><div>*</div>required</div></div> <div>Input example: 1</div> <div>orderProductsId</div>	

Execute

Responses

Code	Description	Links
204	No Content, OrderProduct removed	No links
500	Internal Server Error	No links

POST

/api/{cardId}/welcome/topUp

Top ups the card

POST. /api/{cardId}/welcome/topUp

Welcome screen

GET

/api/{cardId}/welcome/logout/{cardIdSecondTime}

Gets users cardId input and logs out

GET

/api/{cardId}/welcome/order/allProducts

Retrieves all the products

POST

/api/{cardId}/welcome/order/createOrder

Registers the card and stores Employee details

POST

/api/{cardId}/welcome/order/createOrderProduct

Creates an OrderProduct to track selected products

PUT

/api/{cardId}/welcome/order/{orderEmployeeId}/payment

Takes care of the payment and updates the card's balance if successful

PUT

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Updates orderProduct quantity and price when another item of the same product is added. It updates the order total.

DELETE

/api/{cardId}/welcome/order/{orderEmployeeId}/{orderProductsId}

Deletes orderProduct and updates Order total

POST

/api/{cardId}/welcome/topUp

Top ups the card

Parameters

No parameters

Request body

application/json

Input example: { "card": { "cardId": "r7jTG7dqBy5wG04L" }, "topUpAmount": 20 }

```
{
  "card": {
    "cardId": "string",
    "employee": {
      "cardId": "string",
      "employeeId": "string",
      "employeeName": "string",
      "employeeEmail": "string",
      "employeeMobileNumber": "string"
    },
    "cardPassCode": 0
  },
  "topUpAmount": 0
}
```

Execute

Responses

Code	Description	Links
201	Top up successful	No links
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value</div> <div><pre>{ "card": { "cardId": "r7jTG7dqBy5wG04L", "cardPassCode": 0 }, "topUpAmount": 20 }</pre></div>	
500	Internal Server Error	No links