

CheckMate | Team 3 - Ethan Ho, Sarvani Yoganand Lakshmitulasi, Allen Chacko Johny, Carlos Vanegas Hinojosa, Mina Messad

CS 673: 101

Backlog - <https://github.com/WokFriedE/projects/7/views/7>

Github - <https://github.com/WokFriedE/CheckMate>

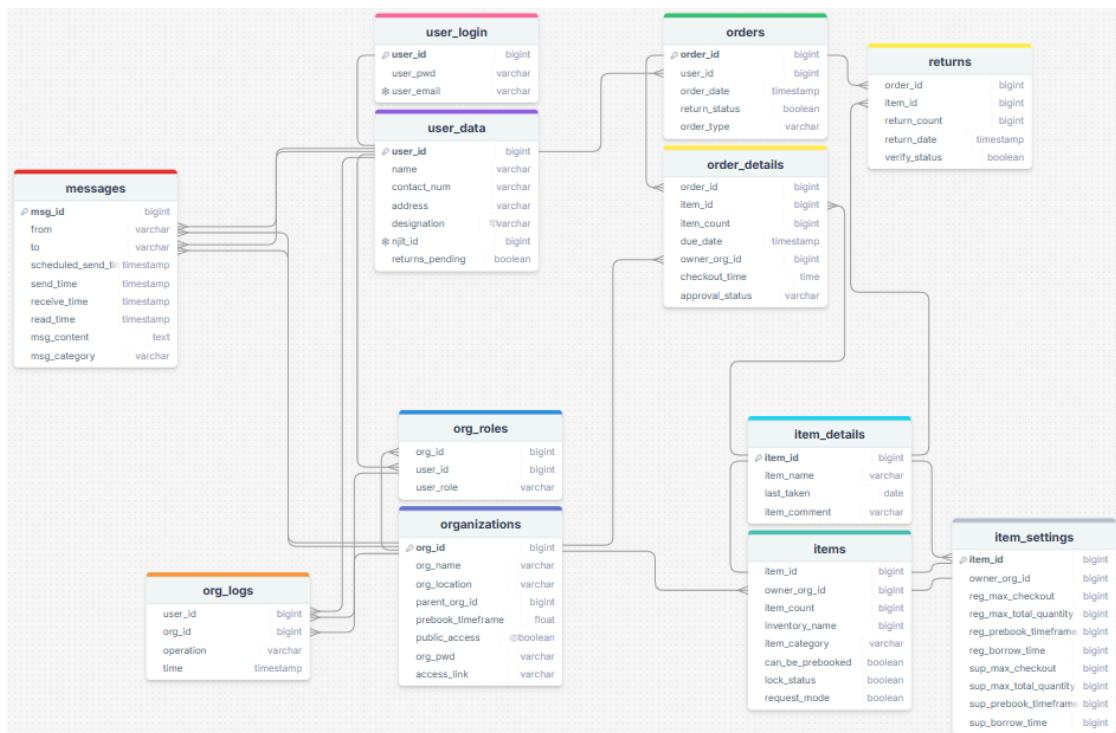
Backlog Items WIP:

- Login
- Active Records
- Items CRUD operations

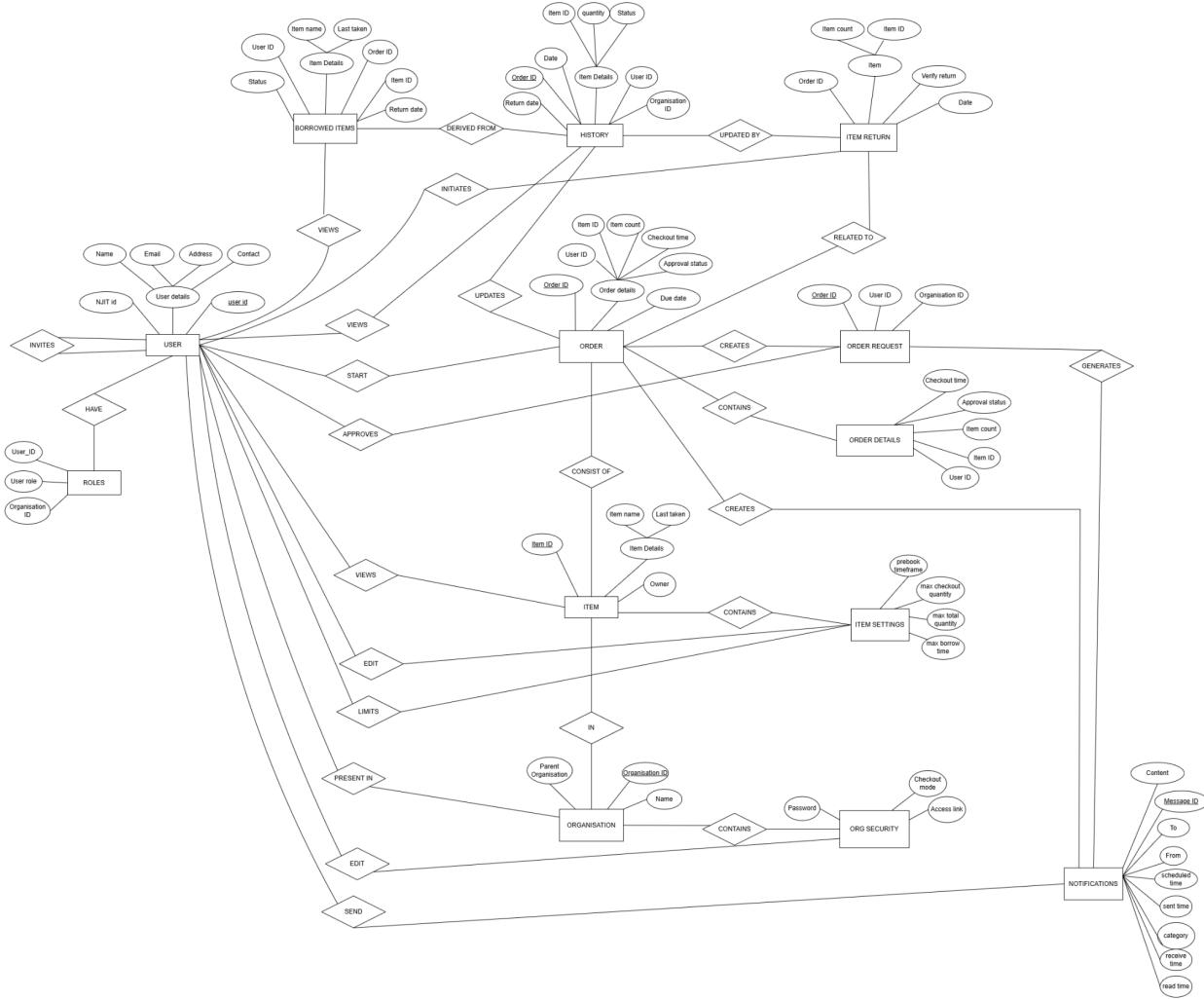
Note: OpenAPI export of APIs will be attached at the end of this document for clearer view of estimated bodies and responses

Database Entities

Database Tables



Entities Relationships



Scenarios / Features

Scenario: Getting Started

- **Personas Involved:** Professor
- **Scenario Description**
 - Smith wants a simple system to track his textbooks, lab tools, and equipment loaned to students with a quick start. He would also like the set up to include a way to restrict or limit access to the application with something like passwords.
- **User Stories**

- As a professor, I want to import my old inventory sheets in under a minute so I can quickly go onboard with my 40 items.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/items/transfer	Item + Item Details	Format: CSV,...	200 <Format export file>
POST: orgs/<org_id>/items/transfer	Item + Item Details	File upload CSV	201:{message: "complete"} 415 wrong media 413 too large 500 server error {message: <sanitized error message>}

View Route	Reason
orgs/<org_id>/dashboard/inventory/	View the inventory of items

- As a professor, I want to set a password for item requests so that only my students and club members can access them.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/settings/password	Organizations or organization_passwords	{password: <hash>, type: create verify }	create 201: {message: success} Verify 200: {access: True, message: ""} Verify 401: {access: False, message: "incorrect password"}

View Route	Reason
orgs/<org_id>/dashboard/settings	View the settings for an organization

- o As a less tech-savvy person, I want a clear, simple display of my items so I don't get overwhelmed by taking stock.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/dashboard/items	Items	sort filter search	[{ <item entity> }]

View Route	Reason
orgs/<org_id>/dashboard/inventory	View the items

- o As a professor, I want to allow any student to check out items, so that I will not need to approve them during hectic periods.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/settings mode	Organizations		200: {message: "success", mode: "self"} 200: {message: "success", mode: "password"} 200: {message: "success", mode: "request"}
POST: orgs/<org_id>/settings mode	Organizations	{mode: "self"} {mode: "password"} {mode: "request"}	create 200: {message: "updated", mode: "self"} Invalid payload 422: {message: "invalid mode"}

View Route	Reason
orgs/<org_id>/dashboard/settings	View the settings for an organization

Scenario: Separating Items in an Inventory

- **Personas Involved:** Professor, Student
- **Scenario Description**
 - Smith has a bunch of items in his inventory and wants to ensure each item has different levels of requirements.
- **User Stories**
 - As a professor, I want to be able to set some items as request-only so that I can ensure expensive equipment isn't taken out without my knowledge.

API Route	Entity	Query/Body	Response
Same as below			

- As a professor, I want to be able to set the maximum amount of items taken out so that students do not take out the entire stock.

API Route	Entity	Query/Body	Response
Same as below			

- As a professor, I want to be able to allow students to only ask for items within a certain amount of time so that they don't book everything in advance.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/item/<item_id>/settings	Item / Item_Settings	{"max_checkout": <int>, "max_quantity": <int>, "can_be_prebooked":<bool>, "prebook_timeframe":<duration>, "item_category": <string> }	201: { "message": "settings created" } **400:** Invalid request **500:** Server error
PATCH / PUT: orgs/<org_id>/item/<item_id>/settings	Item / Item_Settings	{ "max_checkout": <int>, "max_quantity": <int>, "can_be_prebooked":<bool>, "prebook_timeframe":<duration>, "item_category": <string> }	200: { "message": "settings updated" } **404:** Item not found **500:** Server error

View Route	Reason
orgs/<org_id>/dashboard/inventory/<item_id>	View or update item-level settings

Scenario: Prepping For Club Event

- **Personas Involved:** Club Leader
- **Scenario Description**
 - Mary leads a team of 15 students in her club. Before competitions, she must request tools in advance and coordinate who picks them up. She needs a streamlined process to reduce back-and-forth emails and paperwork.
- **User Stories**
 - As a club leader, I want to request more than two items per reservation so that I can gather all necessary tools and materials for club activities in one request.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/item/order	Order / Item_Order	[{ "item_id": <uuid>, "org_id": <uuid>, "item_count": <int>, }] Auth bearer:auth_token": <jwt>	201: { "message": "order created" } **400:** Invalid order **403:** Unauthorized **500:** Server error

View Route	Reason
orgs/<org_id>/checkout	Check out items digitally

- o As a club leader, I want to assign borrowed equipment to specific club members so that I can track who is responsible for each item during an event.

Note: (Handled by assigning orders related table a comment on the request with who got it.)

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/item/order	Order / Item_Order	[{ "item_id": <uuid>, "org_id": <uuid>, "item_count": <int>, Comment: "owned by student xyz" }]	201: { "message": "order created" } **400:** Invalid order **403:** Unauthorized **500:** Server error

View Route	Reason
orgs/<org_id>/checkout	Check out items digitally

- o As a club leader, I want to duplicate past equipment requests so that I can easily prepare for recurring events without re-entering all the items.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/inventory /order/history	Order_History	None	200: [{ "order_id": "<uuid>", "user_email": "<email>", "return_status": "<status>", "order_date": "<iso_date>" }]
POST: orgs/<org_id>/ inventory/order	Order	[{ "item_id": <uuid>, "org_id": <uuid>, "item_count": <int>, "auth_token": <jwt> }]	201: { "message": "order recreated" }

View Route	Reason
orgs/<org_id>/order/history	View history of orders
orgs/<org_id>/order/<order_id>	See details of the order and rerequest again
/order	Global view of item availability across organizations

Note: to see all the organizations, the APIs will first pull all the organizations associated with a user and then return the order history of all sorting it by date

- o As a club leader, I want to view the availability calendar for equipment so that I can plan events around item availability.

API Route	Entity	Query/Body	Response
GET (view): /order	Order_Availability	Optional date filters	200: [{ "item_id": <uuid>, "available_dates": [<iso_date>], "reserved": <bool> }]

Scenario: Handling Order Requests

- **Personas Involved:** Professor
- **Scenario Description**
 - Smith wants a system for professors to handle requests seamlessly from students and receive real-time email notifications to approve or reject them quickly.
- **User Stories**
 - As a professor, I want to approve or deny student requests as soon as I receive them so that I have full control of my inventory.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/item/request/<order_id>	Order_Request	None	200: `{ "order_id": "", "item_id": "", "status": "
POST: orgs/<org_id>/item/request/<order_id>	Order_Request	`{ "status": "approve"	"deny", "due_date": "<iso_date>" }`

View Route	Reason
orgs/<org_id>/dashboard/orders/<order_id>/	View or modify order request status

- As a professor, I should receive request notifications within 5 minutes, so I don't have to manually check the dashboard.

Note: Notification delivery is handled through SMTP (outside HTTP flow).

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/notifications	Notifications	None	200: `[{ "status": sent read pending, "created_at": "", "message": "", "author": "", entity_id: "id or none", entity_type: "order_id" }]`

POST: orgs/<org_id>/notifications	Notifications	{ "target": " <uid or="" org_id="">", "message": "<text>" }</uid>	201: { "message": "notification created" }
--------------------------------------	---------------	---	--

View Route	Reason
/orgs/<org_id>/dashboard/notifications/	View previous notifications if implemented with UI, otherwise email

- o As a professor, I want to give club leaders higher limits so that they are not restricted like students for events.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/users	org_roles	{ "user_id": "<uid>", "role": "upper" }	201: { "message": "user created" }
GET: orgs/<org_id>/settings/users	org_roles	None	200: [{ "user_id": "<uid>", "role": "<role>" }]
PUT: orgs/<org_id>/users	org_roles	{ "user_id": "<uid>", "role": "upper" }	200: { "message": "role assigned" }

Notes:

- Upper = club leader
- Higher limits for quantity, checkout duration, prebook timeframe

View Route	Reason
orgs/<org_id>/dashboard/settings/users	View organization users and roles

Scenario: Multi-tasked Management of Inventory and Users

- **Personas Involved:** Teacher Assistant
- **Scenario Description**
 - Arnold is involved in multiple organizations and wants to efficiently manage inventories across them. He wants a clear way to organize the items he has borrowed, view item statuses, and manage other users' borrowed items for the organizations he oversees.
- **User Stories**
 - As a teaching assistant, I want the system to process my item return in under 20 seconds so that I can use the remaining time to physically store the items.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/item/return	Item_Return	{ "item_id": "<uuid>", "user_id": "<uid>", "return_time": "<iso_date>" }	200: { "message": "return processed" } 400: Invalid request 500: Server error

View Route	Reason
orgs/<org_id>/dashboard/orders	Process and confirm quick item return

- As a teaching assistant, I want to filter my borrowed items by organization or event so I can quickly identify which items belong to which group.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/items/borrowed	Borrowed_Items	?filter_by=organization_id OR ?filter_by=event_id	200: [{ "item_id": "<uuid>", "organization": "<name>", "event": "<name>", "status": "<checked_out>" }]

View Route	Reason
orgs/<org_id>/dashboard/orders	View borrowed items filtered by organization or event

- o As a teaching assistant, I want to generate an invite link or code within 1 minute of creating a new organization so that students can join immediately.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/invite	Organization Using the organization hash_link	<pre>{ "org_id": "<uuid>", "expiration": "<timestamp>" } "created_by": "<uid>", → set by backend </pre>	201: { "invite_code": "<code>", "expires_at": "<timestamp>" } 400: Invalid data 500: Server error

Note: make a hash for the organization id with the time and make the hash link

View Route	Reason
/orgs/<org_id>/dashboard/users/invites	Create and manage organization invite codes

- o As a teaching assistant, I want to have higher-level access above students so I can oversee and update the inventory when needed.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/users/roles	org_roles	<pre>{ "user_id": "<uid>", "role": "organizer" }</pre>	201: { "message": "role assigned" } 400: Invalid request 500: Server error

View Route	Reason
orgs/<org_id>/dashboard/settings/users	Assign and view organization roles

Scenario: Taking Out Equipment

- **Personas Involved:** Student
- **Scenario Description**
 - Emily is preparing for a cultural club event and needs to take out camera equipment to take pictures for the event. She wants to see available items, reserve them online, and get automatic reminders to return them.
- **User Stories**
 - As a student, I want to search for available equipment that takes less than 5 seconds to show results so that I can quickly find what I need for club activities.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/items	Items / item_details	?search=<item_name>	200: [{ "item_id": "<uuid>", "item_name": "<string>", "availability": true }] 500: Server error

View Route	Reason
orgs/<org_id>/items	Search and view available items

- As a student, I want to check out items through the website and take less than a minute to process so that I don't need to fill out manual forms.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/item /checkout	order_details	{ "item_ids": ["<uuid>"]} "due_date": "<iso_date>" → handled by server "user_id": "<uid>" → handled by bearer token	201: { "message": "checkout complete" } 400: Invalid item or user 500: Server error

View Route	Reason
orgs/<org_id>/checkout	Check out items digitally

- o As a student, I want to receive return reminders at least a day before it's due so that I avoid penalties or issues.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/notifications	notifications	{ "user_id": "<uid>", "message": "Reminder: Item due tomorrow", "type": "return" }	201: { "message": "set notification" }

View Route	Reason
orgs/<org_id>/notifications	See automated item return reminders

- o As a student, I want to track my borrowing history so that I know which items I have used and when they were returned.

API Route	Entity	Query/Body	Response
GET: orgs/<org_id>/history	Borrow_History	?user_id=<uid> Empty query would default to bearer token	200: [{ "item_id": "<uuid>", "item_name": "<string>", "borrow_date": "<iso_date>", "return_date": "<iso_date>" }]

View Route	Reason
/order	View borrowing history for user

Scenario: Separation of Access

- **Personas Involved:** Staff
- **Scenario Description**
 - Miranda has been working in Higher Education and Student Affairs for several years. She manages multiple moving parts including documents, time sheets, inventory items, and event planning. The current system is inefficient. A system that provides glanceable information and visibility into future plans will facilitate her workflow. She wants to ensure the inventory is properly separated so that students and her workers do not have the same level of access.
- **User Stories**
 - As a director, I want role based permissions to restrict access so that students cannot view the organization's inventory.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/settings/visibility	organizations	{ org_id: "org" visibility: "public" } { org_id: "org" visibility: "private" }	create 200: {message: success}
GET: orgs/<org_id>/settings/visibility	organizations		200: {visibility: public / private}

View Route	Reason
orgs/<org_id>/dashboard/settings	View the settings for an organization

- o As a director, I want to add any student workers to my team so that they can also handle the inventory.

API Route	Entity	Query/Body	Response
POST: orgs/<org_id>/users	org_roles	{ "user_id": "<uid>", "role": "user" }	201: { "message": "role assigned" }

View Route	Reason
orgs/<org_id>/dashboard/settings/users	View the settings for an organization

- o As a director, I want to view a clear, organized timeline of upcoming events, each with its own plans, notes, and assigned individuals so that I can efficiently manage and prepare for future activities.

API Route	Entity	Query/Body	Response
GET /org/<org_id>/orders/calender	Orders / order_details Reference user_data	Params: Month Year	[{ order_id: ... due_date: ... item_count: ... User_id: ... Name: ... }, ...]

View Route	Reason
orgs/<org_id>/dashboard/orders/upcoming	View the settings for an organization

Test Cases

Scenario 1: Getting Started

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC1.1	Verify importing 40 items from CSV works correctly.	1. Log in as Professor. 2. Upload valid CSV with 40 items.	System returns 201 Created with message 'complete'. All 40 items appear in the inventory list.	Ensures import and initialization reliability.	≤ 1 minute
TC1.2	Validate password protection setup for item requests.	1. Set password with POST /settings/password. 2. Verify using the created password.	Creation returns 201 success; verification 200 'access: True'; incorrect returns 401.	Ensures secure access control.	≤ 3 seconds
TC1.3	Confirm item list displays with sort, filter, and search.	1. Access GET /items. 2. Apply sort, filter, and search features.	Returns 200 with accurate data and correct filters applied.	Confirms UI responsiveness.	≤ 2 seconds
TC1.4	Verify invalid CSV upload rejected.	1. Upload malformed CSV. 2. Observe response.	415 Unsupported Media Type with descriptive error.	Ensures input validation.	≤ 5 seconds

TC1.5	Validate incorrect password handling.	1. Attempt login with wrong password. 2. Observe response.	401 Unauthorized 'access: False'.	Verifies authentication security.	≤ 3 seconds
TC1.6	Verify responsive UI for item list.	1. Open inventory on desktop/mobile. 2. Observe layout.	UI adjusts properly without distortion.	Ensures accessibility.	Instant
TC1.7	Verify duplicate CSV imports are prevented.	1. Upload the same CSV twice. 2. Observe response.	409 Conflict returned, no duplicates created.	Maintains data integrity.	≤ 10 seconds

Scenario 2: Separating Items in an Inventory

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC2.1	Verify ability to mark items as request-only.	1. Open item settings. 2. Set can_be_prebooked = false.	201 Created. Item flagged as request-only.	Restricts unauthorized access.	Instant update
TC2.2	Verify max quantity per student enforced.	1. Attempt checkout exceeding max_quantity. 2. Observe response.	400 Invalid requests returned.	Prevents overuse of stock.	Immediate validation
TC2.3	Verify booking timeframe enforcement.	1. Attempt prebooking outside the timeframe. 2. Observe response.	400 Invalid timeframe returned.	Maintains booking control.	≤ 3 seconds

Scenario 3: Prepping for Club Event

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC3.1	Verify multi-item request functionality.	1. Log in as Club Leader. 2. Submit POST /item/order with >2 items.	201 Created with 'order created'.	Validates batch ordering.	≤ 5 seconds
TC3.2	Verify assignment of borrowed items to members.	1. Include member name in comment. 2. View order log.	Comment saved as 'owned by student xyz'.	Improves accountability.	≤ 5 seconds
TC3.3	Verify ability to duplicate past requests.	1. Access order history. 2. Recreate past order.	201 Created, 'order recreated'.	Simplifies recurring setup.	≤ 10 seconds
TC3.4	Verify availability calendar displays accurate data.	1. GET /order. 2. Review available/reserved flags.	200 OK with correct availability data.	Supports event planning.	≤ 3 seconds

Scenario 4: Handling Order Requests

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC4.1	Verify professors can approve/deny student requests.	1. View pending requests. 2. Approve or deny.	Order status updates (200 OK).	Confirms approval workflow.	≤ 2 seconds
TC4.2	Verify professors receive timely notifications.	1. Submit student requests. 2. Check GET /notifications.	Notification appears ≤ 5 minutes.	Ensures prompt communication.	≤ 5 minutes

TC4.3	Verified professors can assign higher limits to club leaders.	1. Update role via POST /users role='club'. 2. Confirm via GET /settings/users.	201 Created; role visible in list.	Enables role flexibility.	Immediate effect
-------	---	--	------------------------------------	---------------------------	------------------

Scenario 5: Multi-tasked Management of Inventory and Users

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC5.1	Verify items return within 20 seconds.	1. POST /item/return. 2. Check response time.	200 OK 'return processed' $\leq 20s$.	Ensures smooth return process.	≤ 20 seconds
TC5.2	Verify filtering borrowed items by organization/event.	1. GET /items/borrowed?filter_by=organization_id. 2. Compare output.	200 OK with correct filtered subset.	Simplifies tracking.	≤ 2 seconds
TC5.3	Verify invite code creation for organization.	1. POST /orgs/<org_id>/invite. 2. Observe response.	201 Created with valid code and expiration.	Supports fast onboarding.	≤ 1 minute

Scenario 6: Taking Out Equipment

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC6.1	Verify successful checkout of an available item.	1. Log in as a Student. 2. Search for available items by ID. 3. Check out items.	Checkout successfully and recorded in the system.	Confirms basic borrowing workflow.	≤ 5 seconds

TC6.2	Verify checkout is rejected for items marked as 'request-only'.	1. Attempt to check out an item marked as request-only.	System rejects checkout with appropriate message.	Protects restricted inventory.	Instant response
-------	---	---	---	--------------------------------	------------------

Scenario 7: Separation of Access

Test Case ID	Test Objective	Steps to Execute	Expected Result	Criteria	Time-bound
TC7.1	Verify a student cannot access the Admin/Settings endpoint.	1. Log in as a Student. 2. Attempt to access Admin/Settings section.	Access denied with error message 403 Forbidden.	Ensures system integrity through access control.	Immediate validation
TC7.2	Verify an Admin can successfully update a user's role.	1. Log in as Admin. 2. Navigate to User Management. 3. Update a user's role.	User role updated successfully and reflected in list.	Supports hierarchical management.	Immediate effect

UI Design

Admin

The image displays a grid of 10 wireframe screenshots of a web-based administration interface for a system named "CheckMate". The interface is designed for managing bookings, inventory, requests, and users.

- Login:** A simple login form with fields for "Username" and "Password", and buttons for "Login" and "Signup".
- Inventory option 1 - list view - hifi:** A table-based list view of items with columns for ID, Name, Description, and Action buttons.
- Dashboard:** A dashboard showing current statistics: 39 checkouts, 46 pending, 2 late, and 1 stuck. It also lists "Current Borrowers" and a calendar.
- Notifications:** A list of notifications from various users, each with a timestamp and a brief message.
- Inventory option 1 - list view:** Another list view of items, similar to the first one but with a different layout.
- All Orders:** A list of all orders, categorized by status (All, Current, Upcoming, History).
- User invite and role controls:** A screen for managing user invites and roles, showing a list of users and their assigned roles.
- Inventory option 2 - card view:** A card-based list view of items, where each item is represented by a small card with a picture and details.
- Item details / Item settings:** A detailed view of an item, showing its properties and settings.
- Settings:** A general settings screen for managing organization-level configurations.

User

User Dashboard

Welcome, Username

- [Items](#)
- [Soon to be returned items](#)
- [all other item](#)
- [Organizations](#)
- [recent organizations](#)

Login

/login

CheckMate

-
-
-
-

Register

/register

CheckMate

-
-
-
-
-

User Org Search Up

/orgs

Organizations

- [recent](#)
- [History](#)
- [My orgs](#)
- [Favorites](#)
- [Category](#)
- [Other filters](#)
- [search](#)
- [org 1](#)
- [org 2](#)
- [org 3](#)
- [org 4](#)
- [org 5](#)
- [org 6](#)

User Notifications

/notifications

Welcome, Username

Notifications

-
-
-
- [Message 1](#)
- [mark all as read](#)

User Checkout / Cart

/orgs/<org_id>/checkout

Items

- [Current](#)
- [upcoming reservations](#)
- [History](#)
- [Favorites](#)
-
-
-
- [Group 1](#)
- [Item](#)
- [Item](#)
- [...](#)
- [checkout](#)

Looking Up Items In Organization

/orgs/<org_id>/items

Organizations

- [recent](#)
- [History](#)
- [My orgs](#)
- [Favorites](#)
-
-
- [picture](#)
- [...](#)
- [...](#)
- [...](#)
- [...](#)

User Order History

/order

Items

- [Current](#)
- [upcoming reservations](#)
- [History](#)
- [Favorites](#)
-
-
-
- [Group 1](#)
- [Item](#)
- [Item](#)
- [...](#)
- [More>](#)
- [Group 3](#)
- [Item](#)
- [Item](#)
- [...](#)
- [More>](#)

Item Checkout Details

/orgs/<org_id>/checkout

Items

- [Current](#)
- [upcoming reservations](#)
- [History](#)
- [Favorites](#)
-
-
-
- [Group 1](#)
- [picture](#)
- [description](#)
- [quantity](#)
- [checked out](#)
- [available](#)
- [reserved](#)
- [checked out](#)
- [...](#)
- [...](#)
- [...](#)
- [...](#)

CheckMate Equipment Management API

API and SDK Documentation

Version: 1.0.0

Comprehensive OpenAPI 3.0 specification for the CheckMate Equipment Management System. This API enables organizations, professors, club leaders, teaching assistants, and students to manage inventory, reservations, notifications, and user roles.

Each scenario represents a story grouping of endpoints that address real-world workflows within the system.

Getting Started

orgsOrgIdDashboardItemsGet

List items with sort, filter, and search

GET operation for /orgs/{org_id}/dashboard/items.

GET

/orgs/{org_id}/dashboard/items

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/dashboard/items?sort=&filter=&search="
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Query parameters

Name	Description
sort	String
filter	String
search	String

Responses

Status: 200 - Items

Schema

▼ []

```

▼ { []
  Required: availability,item_id,name
  item_id: string
  name: string
  description: string
  category: string
  availability: boolean
}
]

```

orgsOrgIdItemsTransferGet

Export items and details for an organization

GET operation for /orgs/{org_id}/items/transfer.

GET

/orgs/{org_id}/items/transfer

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```

curl -X GET \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: text/csv,application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/items/transfer?format="

```

Parameters

Path parameters

Name	Description
org_id*	String Required

Query parameters

Name	Description
format	String <i>Export format</i>

Responses

Status: 200 - Export file

Schema

▼ [] string (binary)

Status: 500 - Server error

Schema

```
▼ { []
  Required: message
  message: string
}
```

orgsOrgIdItemsTransferPost

Import items and details for an organization

POST operation for /orgs/{org_id}/items/transfer.

POST

/orgs/{org_id}/items/transfer

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST
-H "X-Api-Key: [[apiKey]]"\"
-H "Accept: application/json"\"
-H "Content-Type: multipart/form-data"\"
"https://api.checkmate.example.com/orgs/{org_id}/items/transfer"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Form parameters

Name	Description
file*	String (binary) Required

Responses

Status: 201 - Complete

Schema

```
▼ { []
  message: string
}
```

Status: 415 - Wrong media

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Status: 413 - Too large

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Status: 500 - Server error

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

orgsOrgIdSettingsModeGet

Get request mode

GET operation for /orgs/{org_id}/settings/mode.

GET

/orgs/{org_id}/settings/mode

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
-H "X-Api-Key: [[apiKey]]"  
-H "Accept: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/settings/mode"
```

Parameters

Path parameters

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

org_id*	String Required
---------	--------------------

Responses

Status: 200 - Mode

Schema

```
▼ { [] }
  Required: message,mode
  message: string
  mode: string
    Enum: self, password, request
}
```

orgsOrgIdSettingsModePost

Update request mode

POST operation for /orgs/{org_id}/settings mode.

POST

/orgs/{org_id}/settings mode

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/settings	mode"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	▼ { [] } Required: mode mode: string Enum: self, password, request }

Responses

Status: 200 - Updated

Schema

```
▼ { []  
  Required: message,mode  
  message: string  
  mode: string  
    Enum: self, password, request  
}
```

Status: 422 - Invalid mode

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

orgsOrgIdSettingsPasswordPost

Create or verify a password for item requests

POST operation for /orgs/{org_id}/settings/password.

POST

/orgs/{org_id}/settings/password

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n-H "Content-Type: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/settings/password"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

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

body *

▼ { [] }

Required: password,type

password: ▼ [] string
hash

type: string

Enum: `create`, `verify`

}

Responses

Status: 201 - Created

Schema

▼ { [] }

Required: message

message: string

}

Status: 200 - Verified

Schema

▼ { [] }

Required: access,message

access: boolean

message: string

}

HandlingOrderRequests

orgsOrgIdItemRequestOrderIdGet

Get a request details

GET operation for /orgs/{org_id}/item/request/{order_id}.

GET

/orgs/{org_id}/item/request/{order_id}

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/item/request/{order_id}"
```

Parameters

Path parameters

Name	Description
org_id*	String Required
order_id*	String Required

Responses

Status: 200 - Request

Schema

```
▼ { []
  Required: item_id,order_id,status
  order_id: string
  item_id: string
  status: string
}
```

orgsOrgIdItemRequestOrderIdPost

Approve or deny a request

POST operation for /orgs/{org_id}/item/request/{order_id}.

POST

/orgs/{org_id}/item/request/{order_id}

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/item/request/{order_id}"
```

Parameters

Path parameters

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

org_id*	String Required
order_id*	String Required

Body parameters

Name	Description
body *	<p>▼ { []</p> <p>Required: status</p> <p>status: string</p> <p style="margin-left: 20px;">Enum: <code>approve</code>, <code>deny</code></p> <p>due_date: ▼ [] string (date-time)</p> <p>}</p>

Responses

Status: 200 - Updated

Schema

```
▼ { []
  Required: message
  message: string
}
```

orgsOrgIdNotificationsGet

List notifications

GET operation for /orgs/{org_id}/notifications.

GET

/orgs/{org_id}/notifications

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "X-Api-Key: [[apiKey]]"\ 
-H "Accept: application/json"\ 
"https://api.checkmate.example.com/orgs/{org_id}/notifications"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Responses

Status: 200 - Notifications

Schema

```
▼ [ ]  
  ▼ { }  
    status: string  
      Enum: sent , read , pending  
    created_at: ▼ [] string (date-time)  
    message: string  
    author: string  
    entity_id: string  
    entity_type: string  
  }  
]
```

orgsOrgIdNotificationsPost

Create a notification

POST operation for /orgs/{org_id}/notifications.

POST

/orgs/{org_id}/notifications

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n-H "Content-Type: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/notifications"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description

body *

▼ { [] }

Required: message,target

target: string

message: string

}

Responses

Status: 201 - Created

Schema

▼ { [] }

Required: message

message: string

}

orgsOrgIdSettingsUsersGet

List organization users and roles

GET operation for /orgs/{org_id}/settings/users.

GET

/orgs/{org_id}/settings/users

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/settings/users"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Responses

Status: 200 - Users

Schema

▼ []

▼ { [] }

user_id: string

```
        role: string  
    }  
]
```

orgsOrgIdUsersPost

Create user role

POST operation for /orgs/{org_id}/users.

POST

/orgs/{org_id}/users

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST\  
-H "X-Api-Key: [[apiKey]]"\\  
-H "Accept: application/json"\\  
-H "Content-Type: application/json"\\  
"https://api.checkmate.example.com/orgs/{org_id}/users"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	▼ { [] Required: role,user_id user_id: string role: string Enum: <code>upper</code> , <code>user</code> }

Responses

Status: 201 - User created

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

orgsOrgIdUsersPut

Assign role to user

PUT operation for /orgs/{org_id}/users.

PUT

/orgs/{org_id}/users

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X PUT\  
-H "X-Api-Key: [[apiKey]]"\\  
-H "Accept: application/json"\\  
-H "Content-Type: application/json"\  
"https://api.checkmate.example.com/orgs/{org_id}/users"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	▼ { [] Required: role,user_id user_id: string role: string Enum: upper , user }

Responses

Status: 200 - Role assigned

Schema

```
▼ { []  
  Required: message  
  message: string
```

}

MultiTaskedManagement

orgsOrgIdInvitePost

Create invite code

POST operation for /orgs/{org_id}/invite.

POST

/orgs/{org_id}/invite

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST  
-H "X-Api-Key: [[apiKey]]"  
-H "Accept: application/json"\n-H "Content-Type: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/invite"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	▼ { [] Required: expiration,org_id org_id: string expiration: integer created_by: string }

Responses

Status: 201 - Invite created

Schema

```
▼ { []  
Required: expires_at,invite_code
```

```
invite_code: string  
expires_at: integer  
}
```

Status: 400 - Invalid data

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Status: 500 - Server error

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

orgsOrgIdItemReturnPost

Process an item return

POST operation for /orgs/{org_id}/item/return.

POST

/orgs/{org_id}/item/return

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n-H "Content-Type: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/item/return"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	<pre>▼ { [] } Required: item_id,return_time,user_id item_id: string user_id: string return_time: ▼ [] string (date-time) }</pre>

Responses

Status: 200 - Return processed

Schema

```
▼ { [] }
  Required: message
    message: string
}
```

Status: 400 - Invalid request

Schema

```
▼ { [] }
  Required: message
    message: string
}
```

Status: 500 - Server error

Schema

```
▼ { [] }
  Required: message
    message: string
}
```

orgsOrgIdItemsBorrowedGet

List borrowed items with optional filters

GET operation for /orgs/{org_id}/items/borrowed.

GET

/orgs/{org_id}/items/borrowed

Usage and SDK Samples

```
curl -X GET  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/items/borrowed?filter_by="
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Query parameters

Name	Description
filter_by	String

Responses

Status: 200 - Borrowed items

Schema

```
▼ [ ]  
  ▼ { }  
    item_id: string  
    organization: string  
    event: string  
    status: string  
  }  
]
```

orgsOrgIdUsersRolesPost

Assign organizer role

POST operation for /orgs/{org_id}/users/roles.

POST

/orgs/{org_id}/users/roles

Usage and SDK Samples

```

curl -X POST \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/users/roles"

```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	<ul style="list-style-type: none"> ▼ { [] } <ul style="list-style-type: none"> Required: role,user_id user_id: string role: string Enum: organizer

Responses

Status: 201 - Role assigned

Schema

```

▼ { []
  Required: message
  message: string
}

```

Status: 400 - Invalid request

Schema

```

▼ { []
  Required: message
  message: string
}

```

Status: 500 - Server error

Schema

```

▼ { []
  Required: message
  message: string
}

```

PreppingForClubEvent

orderAvailabilityGet

Global availability view

GET operation for /order/availability.

GET

/order/availability

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
-H "X-Api-Key: [[apiKey]]"  
-H "Accept: application/json"  
"https://api.checkmate.example.com/order/availability?from=&to="
```

Parameters

Query parameters

Name	Description
from	date (date)
to	date (date)

Responses

Status: 200 - Availability list

Schema

```
▼ [ ]  
  ▼ { }  
    Required: available_dates,item_id,reserved  
    item_id:      string  
    available_dates: ▼ [ ]  
      ▼ [] string (date)  
    ]  
    reserved:      boolean  
  }  
]
```

orgsOrgIdInventoryOrderHistoryGet

List past orders for organization

GET operation for /orgs/{org_id}/inventory/order/history.

GET

/orgs/{org_id}/inventory/order/history

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "X-Api-Key: [[apiKey]]" \
-H "Accept: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/inventory/order/history"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Responses

Status: 200 - History

Schema

```
▼ [ ]  
  ▼ { }  
    Required: order_date,order_id,user_email  
    order_id: string  
    user_email: ▼ [] string (email)  
    return_status: string  
    order_date: ▼ [] string (date-time)  
  }  
]
```

orgsOrgIdInventoryOrderPost

Recreate a past order

POST operation for /orgs/{org_id}/inventory/order.

POST

/orgs/{org_id}/inventory/order

Usage and SDK Samples

```
curl -X POST \
-H "X-Api-Key: [[apiKey]]"\ 
-H "Accept: application/json"\ 
-H "Content-Type: application/json"\ 
"https://api.checkmate.example.com/orgs/{org_id}/inventory/order"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	<ul style="list-style-type: none"> ▼ [] ▼ { [] } <ul style="list-style-type: none"> Required: item_count,item_id,org_id item_id: string org_id: string item_count: integer auth_token: string }

Responses

Status: 201 - Order recreated

Schema

```

▼ { [] }
  Required: message
  message: string
}
```

orgsOrgIdItemOrderPost

Create an order with multiple items

POST operation for /orgs/{org_id}/item/order.

POST

/orgs/{org_id}/item/order

Usage and SDK Samples

```

curl -X POST \
-H "Authorization: Bearer [[accessToken]]" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
"https://api.checkmate.example.com/orgs/{org_id}/item/order"

```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	<ul style="list-style-type: none"> ▼ [] ▼ { [] } <ul style="list-style-type: none"> Required: item_count,item_id,org_id item_id: string org_id: string item_count: integer comment: string }

Responses

Status: 201 - Order created

Schema

```

▼ { []
  Required: message
  message: string
}

```

Status: 400 - Invalid order

Schema

```

▼ { []
  Required: message
  message: string
}

```

Status: 403 - Unauthorized

Schema

```
▼ { }
  Required: message
  message: string
}
```

Status: 500 - Server error

Schema

```
▼ { }
  Required: message
  message: string
}
```

SeparatingItemsInInventory

orgsOrgIdItemItemIdSettingsPatch

Partial update of item settings

PATCH operation for /orgs/{org_id}/item/{item_id}/settings.

PATCH

/orgs/{org_id}/item/{item_id}/settings

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X PATCH \
-H "X-Api-Key: [[apiKey]]"
-H "Accept: application/json"
-H "Content-Type: application/json"
"https://api.checkmate.example.com/orgs/{org_id}/item/{item_id}/settings"
```

Parameters

Path parameters

Name	Description
org_id*	String Required
item_id*	String Required

Body parameters

Name	Description
body *	<pre>▼ { [] } max_checkout: integer max_quantity: integer can_be_prebooked: boolean prebook_timeframe: ▼ [] string ISO 8601 duration or human readable item_category: string }</pre>

Responses

Status: 200 - Updated

Schema

```
▼ { []
  Required: message
  message: string
}
```

Status: 404 - Item not found

Schema

```
▼ { []
  Required: message
  message: string
}
```

orgsOrgIdItemItemIdSettingsPost

Create item settings

POST operation for /orgs/{org_id}/item/{item_id}/settings.

POST

/orgs/{org_id}/item/{item_id}/settings

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST
-H "X-Api-Key: [[apiKey]]"
-H "Accept: application/json"
-H "Content-Type: application/json"
"https://api.checkmate.example.com/orgs/{org_id}/item/{item_id}/settings"
```

Parameters

Path parameters

Name	Description
org_id*	String Required
item_id*	String Required

Body parameters

Name	Description
body *	<pre>▼ { [] }</pre> <p>max_checkout: integer max_quantity: integer can_be_prebooked: boolean prebook_timeframe: ▼ [] string <i>ISO 8601 duration or human readable</i> item_category: string }</p>

Responses

Status: 201 - Created

Schema

```
▼ { [] }
```

Required: message
message: string
}

Status: 400 - Invalid

Schema

```
▼ { [] }
```

Required: message
message: string
}

Status: 500 - Server error

Schema

```
▼ { [] }
```

Required: message
message: string
}

orgsOrgIdItemItemIdSettingsPut

Update item settings

PUT operation for /orgs/{org_id}/item/{item_id}/settings.

PUT

/orgs/{org_id}/item/{item_id}/settings

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X PUT  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n-H "Content-Type: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/item/{item_id}/settings"
```

Parameters

Path parameters

Name	Description
org_id*	String Required
item_id*	String Required

Body parameters

Name	Description
body *	▼ { [] max_checkout: integer max_quantity: integer can_be_prebooked: boolean prebook_timeframe: ▼ [] string <i>ISO 8601 duration or human readable</i> item_category: string }

Responses

Status: 200 - Updated

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Status: 404 - Item not found

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Status: 500 - Server error

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

SeparationOfAccess

orgsOrgIdOrdersCalendarGet

Upcoming orders calendar view

GET operation for /orgs/{org_id}/orders/calendar.

GET

/orgs/{org_id}/orders/calendar

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET\  
-H "X-Api-Key: [[apiKey]]"  
-H "Accept: application/json"  
"https://api.checkmate.example.com/orgs/{org_id}/orders/calendar?month=&year="
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Query parameters

Name	Description
month	Integer
year	Integer

Responses

Status: 200 - Upcoming orders

Schema

```
▼ [ ]  
  ▼ { }  
    order_id: string  
    due_date: ▼ [] string (date-time)  
    item_count: integer  
    user_id: string  
    name: string  
  }  
]
```

orgsOrgIdSettingsVisibilityGet

Get organization visibility

GET operation for /orgs/{org_id}/settings/visibility.

GET

/orgs/{org_id}/settings/visibility

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/settings/visibility"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Responses

Status: 200 - Visibility

Schema

```
▼ { }  
  Required: visibility  
  visibility: string  
  Enum: public, private
```

}

orgsOrgIdSettingsVisibilityPost

Set organization visibility

POST operation for /orgs/{org_id}/settings/visibility.

POST

/orgs/{org_id}/settings/visibility

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n-H "Content-Type: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/settings/visibility"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	<pre>▼ { [] }</pre> <p>Required: org_id,visibility</p> <p>org_id: string</p> <p>visibility: string</p> <p style="text-align: center;">Enum: <code>public</code> , <code>private</code></p>

Responses

Status: 200 - Updated

Schema

```
▼ { [] }
```

Required: message

message: string

TakingOutEquipment

orgsOrgIdHistoryGet

Borrowing history for a user

GET operation for /orgs/{org_id}/history.

GET

/orgs/{org_id}/history

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET \
-H "Authorization: Bearer [[accessToken]]"
-H "Accept: application/json"
"https://api.checkmate.example.com/orgs/{org_id}/history?user_id="
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Query parameters

Name	Description
user_id	String <i>Defaults to bearer if empty</i>

Responses

Status: 200 - History

Schema

```
▼ [ ]
  ▼ { }
    item_id: string
    item_name: string
    borrow_date: ▼ [] string (date-time)
    return_date: ▼ [] string (date-time)
  }
]
```

orgsOrgIdItemCheckoutPost

Checkout items

POST operation for /orgs/{org_id}/item/checkout.

POST

/orgs/{org_id}/item/checkout

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X POST  
-H "Authorization: Bearer [[accessToken]]"\\"  
-H "Accept: application/json"\\"  
-H "Content-Type: application/json"\\"  
"https://api.checkmate.example.com/orgs/{org_id}/item/checkout"
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Body parameters

Name	Description
body *	<pre>▼ { [] Required: item_ids item_ids: ▼ [[] string] due_date: ▼ [] string (date-time) }</pre>

Responses

Status: 201 - Checkout complete

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Status: 400 - Invalid item or user

Schema

```
▼ { []  
  Required: message  
  message: string
```

}

Status: 500 - Server error

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

orgsOrgIdItemsGet

Search available items

GET operation for /orgs/{org_id}/items.

GET

/orgs/{org_id}/items

Usage and SDK Samples

Curl Java Android Obj-C JavaScript C# PHP Perl Python

```
curl -X GET  
-H "X-Api-Key: [[apiKey]]"\n-H "Accept: application/json"\n"https://api.checkmate.example.com/orgs/{org_id}/items?search="
```

Parameters

Path parameters

Name	Description
org_id*	String Required

Query parameters

Name	Description
search	String

Responses

Status: 200 - Items

Schema

```
▼ [ []  
  ▼ { []  
    Required: availability,item_id,item_name  
    item_id: string
```

```
    item_name: string  
    availability: boolean  
  }  
]
```

Status: 500 - Server error

Schema

```
▼ { []  
  Required: message  
  message: string  
}
```

Suggestions, contact, support and error reporting:

Information URL: <https://helloreverb.com> (<https://helloreverb.com>)

Contact Info: hello@helloreverb.com (hello@helloreverb.com)

All rights reserved

<http://apache.org/licenses/LICENSE-2.0.html>