You can either use the link below or see the screenshots
https://app.swaggerhub.com/apis/epsilonc01/epsilon/1.0.0

Epsilon

1.0.0

[Base URL: virtserver.swaggerhub.com/epsilonc01/epsilon/1.0.0]

startup marketplace. Right now the Users definition is a bit vague since it contains empolyees, CEO and admin, even the job seekers. However, they will be classified in later sprints.

Contact the developer

Schemes

HTTPS ⌄

Group Information and members of a start up company. ⌄

POST `/testbtn` event handling for removing or set a user as admin button ↵

Parameters   Try it out

Name	Description
val * required string *(query)*	string indicating whether remove/set admin, and the team id, user id val - string indicating whether remove/set admin, and t

Responses   Response content type   application/json ⌄

Code	Description
200	operation succeeded.
400	database error.

POST   `/remove`   remove a user from a team   ↵

Parameters   [Try it out]

Name	Description
tid * required string *(query)*	team id [tid - team id]
uid * required string *(query)*	user id [uid - user id]

Responses   Response content type   [application/json ▾]

Code	Description
201	operation succeeded.
400	database error.

GET `/jointeamrequest` display all the requests to join a certain company ↵

Parameters

<div style="text-align:right">**Try it out**</div>

Name	Description
tid * required string *(query)*	company id `tid - company id`

Responses

Response content type application/json ⌄

Code	Description
200	operation succeeded.
400	database error.

POST /jointeamrequest accept or decline pending join requests ↵

Parameters		Try it out

Name	Description
val * required string *(query)*	string representation of option and team id [val - string representation of option and team id]

Responses	Response content type	application/json ⌄

Code	Description
201	operation succeeded.
400	database error.

User Operations available to regular users, including company founder, team member, empolyee, team admin, job seekers. ⌄

POST /login check for credential and log in the user ↵

Pass in username and password redirect to the hello page.

Parameters	Try it out

Name	Description
username * required string *(query)*	username
password * required string *(query)*	password

Responses	Response content type	application/json ⌄

Code	Description
200	user logged in
400	bad request
401	wrong credential

POST `/registration` register a company and add at least one company owner. ↵

Parameters

Try it out

Name	Description
teamname * required string *(query)*	team name teamname - team name
teamdesc * required string *(query)*	description teamdesc - description
uid string *(query)*	founder id uid - founder id

Responses

Response content type   application/json ⌄

Code	Description
201	registration is successful.
400	database error.

GET `/displayteam` display all users of a team ↵

Parameters

Try it out

Name	Description
tid * required string *(query)*	the id of company to display tid - the id of company to display

Responses   **Response content type** application/json ⌄

Code	Description
200	operation succeeded.
400	database error or no team.
401	team not found.

SiteOwner Operations available to site owner. ⌄

POST `/create` batch create some employees ↵

Parameters	Try it out

No parameters

Responses   Response content type   **application/json** ⌄

Code	Description
201	creation is successful.
400	database error.

GET `/test_get_base_url` get the base url of site ↵

Parameters	Try it out

No parameters

Responses   Response content type   **string** ⌄

Code	Description
200	operation succeeded.
301	url moved permanently.

| GET | /deleteAll | delete all the databases(Users, Teams, Request, Roles, Company, RStatus) | ↵ |

Parameters

Try it out

No parameters

Responses

Response content type string

Code	Description
201	operation succeeded.
400	databases error.

Roles ⌄ {

rid*	string($uuid)
	example: 12345
type*	string
	example: director, employee admin

}

Teams ⌄ {

tid*	string($uuid)
	example: 12345
uid*	string
	example: director, employee admin
role*	

Roles ⌄ {

rid*	string($uuid)
	example: 12345
type*	string
	example:
	director, employee admin

}

}

Users ⌄ {

uid*	string
	example: 12345
rid*	string
	example: 12345
name*	string
	example: John
contact*	string
	example: 123435@yahoo.com

}

Company ⌄ {

tid*	string($uuid)
	example: 12345
name*	string
	example: Awesome Company Limited
description*	string
	example: target at creating Saas application for legal services.
create_date*	string($date-time)
	example: OrderedMap {}

}

Request ⌄ {

req_id*	string($uuid)
	example: 12345
uid*	string
	example: 12345
tid*	string
	example: 12345

}

Sid ⌄ {

create_date*	string($date-time)
	example: OrderedMap {}
last_update*	string($date-time)
	example: OrderedMap {}
seen*	boolean
	example: true

}

Rstatus ⌄ {

sid*	string
	example: 12345
name*	string
	example: I request to join company delta.

}