

# Epsilon

Startup marketplace.

More information: <https://helloverb.com>

Contact Info: [epsilonc01@yahoo.com](mailto:epsilonc01@yahoo.com)

Version: 1.0.0

BasePath: /epsilonc01/epsilon/1.0.0

All rights reserved

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

## Access

## Methods

[ [Jump to Models](#) ]

## Table of Contents

### [RegisteredUser](#)

- [GET /user/](#)

### [TeamAdmin](#)

- [GET /displayteam](#)
- [GET /jointeamrequest](#)
- [POST /remove](#)
- [POST /updaterole](#)

### [UnregisteredUser](#)

- [POST /userRegistration](#)

### [User](#)

- [POST /registration](#)
- [POST /login](#)
- [GET /previousHome](#)
- [POST /search](#)
- [GET /search](#)

## RegisteredUser

GET /user/ [Up](#)

display user profile with matching UID to url, the profile is available for editing. (**displayUser**)

### Query parameters

**uid (required)**

*Query Parameter* — the id of user to display

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

**200**

operation succeeded.

**400**

database error or no user.

**401**

user not found.

## TeamAdmin

GET /displayteam [Up](#)

---

display all users of a team, with option to remove them or set them as admin. (**displayteam**)

### Query parameters

**tid (required)**

*Query Parameter* — the id of company to display

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

**200**

operation succeeded.

**400**

database error or no team.

**401**

team not found.

---

## GET /jointeamrequest

Up

display all the join requests of a company (**jointeamrequestGet**)

### Query parameters

**tid (required)**

*Query Parameter* — company id

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

operation succeeded.

400

database error.

---

## POST /remove

Up

remove a user from a team (**remove**)

### Query parameters

**tid (required)**

*Query Parameter* — team id

**uid (required)**

*Query Parameter* — user id

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

201

operation succeeded.

400

database error.

---

## POST /updaterole

Up

event handling for removing or set a user as admin button (**testbtn**)

### Query parameters

**tid (required)**

*Query Parameter* – team id of the company

**uid (required)**

*Query Parameter* – user id

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

**Responses**

**200**

operation succeeded.

**400**

database error.

## UnregisteredUser

### POST /userRegistration

Up

create an account with name, contact and password. (**userRegistration**)

**Query parameters****name (required)**

*Query Parameter* –

**contact (required)**

*Query Parameter* –

**password (required)**

*Query Parameter* –

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

operation succeeded.

400

user exists.

## User

### POST /registration

Up

register a company and add at least one company owner. (**companyRegistration**)

#### Query parameters

**teamname (required)**

*Query Parameter* — team name

**teamdesc (required)**

*Query Parameter* — description

**uid (optional)**

*Query Parameter* — founder id

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

201

registration is successful.

400  
database error.

---

## POST /login

Up

check for credential and log in the user (**login**)

Pass in username (email) and password and redirect to the hello page.

### Query parameters

**username (required)**

*Query Parameter* —

**password (required)**

*Query Parameter* —

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

user logged in

400

bad request

401

wrong credential

---

## GET /previousHome

Up

return to home page, and log out if user was previously logged in. (**previousHome**)

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- string

### Responses

**200**

operation succeeded.

**301**

url moved permanently.

## POST /search

Up

search for companies based on search query. (search)

### Query parameters

**keyword (required)**

*Query Parameter* — search query

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

**200**

operation succeeded, return a list of companies.

**400**

database error or no user.

## GET /search

Up

return the search page. (searchGet)



## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

operation succeeded.

400

database error or no user.

---

## Models

[ [Jump to Methods](#) ]

### Table of Contents

1. [Company -](#)
2. [CompanyTags -](#)
3. [Industry -](#)
4. [Request -](#)
5. [Roles -](#)
6. [Rstatus -](#)
7. [Tags -](#)
8. [Teams -](#)
9. [Users -](#)

### Company -

[Up](#)

basic info of each company.

tid

[Integer](#) company id.

name

[String](#)

description

[String](#)

ind\_id

[Integer](#) industry id.

create\_date

[BigDecimal](#) format: date-time

## CompanyTags -

[Up](#)

tag specific of a company.

ctid

[Integer](#) tag id within a company.

tid

[String](#) the company that this tag is associated to.

tag\_id

[Integer](#) id to the generic tag.

## Industry -

[Up](#)

industres available for company to choose from.

ind\_id

[Integer](#)

name

[String](#)

## Request -

[Up](#)

complete info of each request sent by users to a company.

req\_id

[Integer](#)

uid

[Integer](#) id of sender user.

tid

[Integer](#) id of the receiver company.

sid

[Integer](#) status id, can be ACCEPTED, REJECTED or PENDING.

create\_date

[BigDecimal](#) format: date-time

last\_update

[BigDecimal](#) format: date-time

seen

[Boolean](#)

## Roles -

[Up](#)

mandatory role types in the company.

rid

[Integer](#)

role\_type

[String](#)

*example: CEO, team admin*

## Rstatus -

[Up](#)

request status.

sid

[Integer](#) status id, can be ACCEPTED, REJECTED or PENDING.

name

[String](#) detailed description of this request status.

## Tags -

[Up](#)

generic tag of a company.

tag\_id  
[Integer](#)

name  
[String](#)

ind\_id  
[Integer](#) the id of the industry that the tag describes.

## Teams -

[Up](#)

information about individual users, their company and role.

tid  
[Integer](#) the id of company where the user is registered.

uid  
[Integer](#) user id.

rid  
[Integer](#) the id of the role of user in his/her company.

## Users -

[Up](#)

basic info of each user.

uid  
[Integer](#)

rid  
[Integer](#) role id.

name  
[String](#)

contact  
[String](#)  
example: 123435@yahoo.com

password  
[String](#)

description

*String*