

API Requests

User/Organization Account Login API Specification (Authentication)

Create new user login

POST /register/user/

```
Request
{
  "name": "Kyle Chu",
  "login_email": "kgc42@cornell.edu",
  "login_password": "asdf"
}
```

```
Response
<HTTP STATUS CODE 201>
{
  "user_id": 10,
  "session_token": "5b4bcb15f79244a9c4b4af3342419966f54367ec",
  "session_expiration": "2023-12-08 01:15:26.600314",
  "refresh_token": "336b533d22d33a6cd31cb67b1f23d6682fa16964"
}
```

Login to user

POST /login/user/

```
Request
{
  "login_email": "kgc42@cornell.edu",
  "login_password": "asdf"
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "user_id": 10,
  "session_token": "6a025055696f4deb77625c52ee434d3fbd0b849e",
  "session_expiration": "2023-12-08 01:19:32.625196",
  "refresh_token": "4173ac15b5d801961713280d4ecc1d49f9f312a4"
}
```

Logout from user

POST /logout/user/

```
Request
#pass in bearer token (session_token returned when logging in,
#ex: "4887b7c22f24dc611bdc1cac4f21e00c78927418" <-- from route above)
```

```
Response
<HTTP STATUS CODE 200>
"You have successfully logged out"
```

Refresh user session token

POST /session/user

Request

#pass in bearer token (refresh_token returned when logging in,
#ex: "4887b7c22f24dc611bdc1cac4f21e00c78927418" <-- from route above)

Response

<HTTP STATUS CODE 200>

```
{
  "user_id": 10,
  "session_token": "59a0d9702ad6d2a835c7d70e4f0fd17297464681",
  "session_expiration": "2023-12-08 01:27:15.465664",
  "refresh_token": "de726cf8c4268810c8e386c7011c767295ae6c90"
}
```

Create new organization login

POST /register/org/

Request

```
{
  "name": "dti",
  "login_email": "dti@gmail.com",
  "login_password": "password1"
}
```

Response

<HTTP STATUS CODE 200>

```
{
  "org_id": 1,
  "session_token": "6188485273066ae216a1b2b3f386e288ccf1fef1", #authentication token for the current session
  "session_expiration": "2023-12-07 00:08:31.887720", #expiration time and date for current session
  "refresh_token": "8e4f88d1ee551dc5a2cf5a9ade8a4a6c2635a351" #authentication token required to refresh session (resets timer)
}
```

Login to organization

POST /login/org/

Request

```
{
  "login_email": "dti@gmail.com",
  "login_password": "password1"
}
```

Response

<HTTP STATUS CODE 200>

```
{
  "org_id": 1,
  "session_token": "4887b7c22f24dc611bdc1cac4f21e00c78927418",
  "session_expiration": "2023-12-06 23:47:35.132876",
  "refresh_token": "1eee7adaa1f6e6d763183bb5a128d7a92c69f7d0"
}
```

Logout from organization

POST /logout/org/

Request

#pass in bearer token (session_token returned when logging in,
#ex: "4887b7c22f24dc611bdc1cac4f21e00c78927418" <-- from route above)

```
Response
<HTTP STATUS CODE 200>
"You have successfully logged out"
```

Refresh organization token

POST /session/org/

```
Request
#pass in bearer token (refresh_token returned when logging in,
#ex: "1eee7adaa1f6e6d763183bb5a128d7a92c69f7d0" <-- from route above)
```

```
Response
<HTTP STATUS CODE 200>
{
  "org_id": 1,
  "session_token": "5c69b16bee7110554fe3adf4ecd31a085644c2bd",
  "session_expiration": "2023-12-07 00:16:49.933269",
  "refresh_token": "67b309f2ff2a970f7ca28ac04a544daf0c213bda"
}
```

User API Specification

Get all users

GET /api/users/

```
Response
<HTTP STATUS CODE 200>
{
  "users": [
    {
      "id": 1,
      "name": "Nicole Qiu",
      "pfp": "<new link url>"
    },
    {
      "id": 2,
      "name": "Kyle Chu",
      "pfp": "google.com"
    }
  ]
}
```

Get a specific user

GET /api/users/<int:user_id>/

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "Nicole Qiu",
  "year": 2027,
  "major": "Computer Science",
  "pfp": "google.come",
  "banner": "google.com",
  "bio": "im nicole",
  "instagram": "nicole_qiu",
  "linkedin": "nicole_qiu",
  "public_email": "nfq2@cornell.edu",
  "organizations": [],
  "tags": [],
  "busy times": [],
```

```

    "free times": [],
    "preferred locations": [],
    "org requests": [],
    "chat requests sent": [],
    "chat requests received": []
  }

```

Delete a user (REQUIRES USER SESSION_TOKEN)

DELETE /api/users/<int:user_id>/

```

Response
<HTTP STATUS CODE 200>
{
  "id": 3,
  "name": "Kyle Chu",
  "year": 2027,
  "major": "Computer Science",
  "pfp": "google.com",
  "banner": "google.com",
  "bio": "im nicole",
  "instagram": "kych2204",
  "linkedin": "kyle_chu",
  "public_email": "kgc42@cornell.edu",
  "organizations": [],
  "tags": [],
  "busy times": [],
  "free times": [],
  "preferred locations": [],
  "org requests": [],
  "chat requests sent": [],
  "chat requests received": []
}

```

Update a user (REQUIRES USER SESSION_TOKEN)

PUT /api/users/<int:user_id>/

```

Request
{
  "name": "Kyle"
  "year": 2026,
  "major": "updated_MechE",
  "pfp": "<new link url>",           #can be empty or URL
  "bio": "new written bio here",    #can be empty
  "email": "<new email>",
  "banner": "updated_url",          #can be empty
  "instagram": "updated_ig",
  "linkedin": "updated_linkedin",
  "status": False                   #can be empty --> default to False if empty
}

```

```

Response
<HTTP STATUS CODE 201>
{
  "id": 1,
  "name": "Kyle",
  "year": 2026,
  "major": "updated_MechE",
  "pfp": "<new link url>",
  "banner": "google.com",
  "bio": "new written bio here",
  "instagram": "updated_ig",
  "linkedin": "updated_linkedin",
  "public_email": "nfq2@cornell.edu",
  "organizations": [],
  "tags": [],
  "busy times": [],
  "free times": [],
  "preferred locations": [],
  "org requests": [],
}

```

```
"chat requests sent": [],
"chat requests received": []
}
```

Organization API Specification

Get all organizations

GET /api/orgs/

```
Response
<HTTP STATUS CODE 200>
{
  "organizations": [
    {
      "id": 1,
      "name": "AppDev",
      "org pfp": "link",
      "bio": "This is AppDev",
      "users": [],
      "tags": []
    },
    {
      "id": 2,
      "name": "DTI",
      "org pfp": "link",
      "bio": "This is DTI",
      "users": [],
      "tags": []
    }
  ]
}
```

Get specific organization

GET /api/orgs/<int:org_id>/

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "AppDev",
  "pfp": "link",
  "banner": "link",
  "bio": "This is AppDev",
  "users": [],
  "tags": []
}
```

Update org profile (REQUIRES ORG SESSION_TOKEN)

PUT /api/orgs/<int:org_id>/

```
Request
{
  "name": "AppDev",          #cannot be empty String
  "pfp": "see_comment",
  "banner": "see_comment",
  "bio": "see_comment"
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
```

```
"name": "updated_AppDev",
"pfpp": "pfpp",
"banner": "banner",
"bio": "bio",
"users": [],
"tags": []
}
```

Delete an organization (REQUIRES ORG SESSION_TOKEN)

DELETE /api/orgs/<int:org_id>

```
Response
<HTTP STATUS CODE 200>
{
  "id": 5,
  "name": "AppDev",
  "org pfpp": "link",
  "bio": "This is AppDev",
  "users": [],
  "tags": []
}
```

Location API Specification

Add location preferences to user (REQUIRES USER SESSION_TOKEN)

POST /api/locations/<int:user_id>

```
Request
{
  "location": "philips hall"      #cannot be empty
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "Nicole Qiu",
  "pfpp": "<new link url>",
  "location prefs": [
    {
      "id": 1,
      "location": "duffield hall"
    },
    {
      "id": 2,
      "location": "philips hall"
    }
  ]
}
```

Delete location preference from user (REQUIRES USER SESSION_TOKEN)

DELETE /api/locations/int:l_id/user/int:u_id/

```
Response
<HTTP STATUS CODE 200>
{
  "id": 2,
  "location": "philips hall"
}
```

Tag API Specification

Get all tags

GET /api/tags/

```
Response
<HTTP STATUS CODE 200>
{
  "tags": [
    {
      "id": 1,
      "text": "cooking"
    },
    {
      "id": 2,
      "text": "cs"
    },
    {
      "id": 3,
      "text": "Computer Science"
    }
  ]
}
```

Get specific tag

GET /api/tags/<int:tag_id>/

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "text": "cs",
  "users": [
    {
      "id": 1,
      "name": "Nicole Qiu",
      "pfp": "google.come"
    }
  ],
  "organizations": [
    {
      "id": 1,
      "name": "updated_AppDev",
      "org pfp": "pfp"
    }
  ]
}
```

Get all tags from a user or org

GET /api/tags/<int:user_id>/tags/

```
{
  "type": "user" or "organization"
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "tags": [
    {
      "id": 1,
      "text": "cs"
    },
    {
      "id": 2,
```

```
    "text": "business"
  }
]
}
```

Add tags to user (REQUIRES USER SESSION_TOKEN)

POST /api/tags/user/<int:u_id>/

```
Request
{
  "text" : "english"
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "Nicole Qiu",
  "pfp": "google.come",
  "tags": [
    {
      "id": 2,
      "text": "cs"
    },
    {
      "id": 3,
      "text": "Computer Science"
    },
    {
      "id": 4,
      "text": "english"
    }
  ]
}
```

Delete tag from user (REQUIRES USER SESSION_TOKEN)

DELETE /api/tags/int:t_id/user/int:u_id/

```
Response
<HTTP STATUS CODE 200>
{
  "id": 6,
  "text": "bus"
}
```

Add tag to organization (REQUIRES ORG SESSION_TOKEN)

POST /api/tags/org/<int:org_id>/

```
Request
{
  "text": "Computer Science"
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "updated_AppDev",
  "org pfp": "pfp",
  "tags": [
    {
      "id": 1,
      "text": "cs"
    }
  ]
}
```



```

    },
    {
      "id": 2,
      "text": "business"
    }
  ]
}

```

Delete tag from organization (REQUIRES ORG SESSION_TOKEN)

DELETE /api/tags/<int:t_id>/org/<int:o_id>/

```

Response
<HTTP STATUS CODE 200>
{
  "id": 6,
  "text": "bus"
}

```

Time API Specifications

Add free times to user (REQUIRES USER SESSION_TOKEN)

POST /api/ft/<int:user_id>/

```

Request
{
  "month": 11,
  "day": 28,
  "year": 2023,
  "weekday": 4,
  "start time": 840
}

```

#cannot be empty; this field shouldnt be directly written by the user, rather inputted by frontend dep

#cannot be empty; this is the number of minutes from 12 am to the chosen time (in this example,

```

Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "Nicole Qiu",
  "pfp": "<new link url>",
  "free times": [
    {
      "id": 1,
      "month": 10,
      "day": 28,
      "year": 2023,
      "weekday": 4,
      "start time": 840
    },
    {
      "id": 2,
      "month": 11,
      "day": 28,
      "year": 2023,
      "weekday": 4,
      "start time": 840
    }
  ]
}

```

Remove Free Time from user (REQUIRES USER SESSION_TOKEN)

DELETE /api/ft/<int:ft_id>/user/<int:u_id>/

```

Response
<HTTP STATUS CODE 200>
{
  "id": 3,
  "month": 10,
  "day": 2,
  "year": 2023,
  "weekday": 4,
  "start time": 840
}

```

Add busy times to user (REQUIRES USER SESSION_TOKEN)

POST /api/bt/<int:u_id>/

```

Request
{
  "month": 10,
  "day": 28,
  "year": 2023,
  "weekday": 1,
  "start time": 840
}

```

#cannot be empty; this field shouldnt be directly written by the user, rather inputted by frontend dep

#cannot be empty; this is the number of minutes from 12 am to the chosen time (in this example,

```

Response
<HTTP STATUS CODE 200>
{
  "user id": 1,
  "name": "Nicole Qiu",
  "pfp": "<url link>",
  "busy times": [
    {
      "id": 0,
      "month": 9,
      "day": 2,
      "year": 2023,
      "weekday": 6,
      "start time": 600},
    {
      "id": 1,
      "month": 10,
      "day": 28,
      "year": 2023,
      "weekday": 1,
      "start time": 840}
  ]
}

```

Remove Busy Time from user (REQUIRES USER SESSION_TOKEN)

DELETE /api/bt/<int:bt_id>/user/<int:u_id>/

```

Response
<HTTP STATUS CODE 200>
{
  "id": 0,
  "month": 9,
  "day": 2,
  "year": 2023,
  "weekday": 6,
  "start time": 600
}

```

Coffee Chat Request API Specification

Get Coffee Chat Requests from a specific user

```
GET /api/cr/<int:user_id>/
```

Request

```
{
  "status": "received / sent"          #cannot be empty; either "received" or "sent"
}
```

Response

<HTTP STATUS CODE 200>

```
{
  "id": 1,
  "name": "Nicole Qiu",
  "pfp": "<new link url>",
  "received / sent chat requests": [
    {
      "id": 1,
      "start time": 840,
      "month": 10,
      "day": 28,
      "year": 2023,
      "weekday": 4,
      "message": "i want to chat you",
      "accepted": null,
      "sender": 1,
      "receiver": 2
    },
    {
      "id": 2,
      "start time": 840,
      "month": 10,
      "day": 23,
      "year": 2023,
      "weekday": 4,
      "message": "i want to chat you",
      "accepted": null,
      "sender": 1,
      "receiver": 2
    }
  ]
}
```

Get all confirmed coffee chat requests from a specific user (REQUIRES USER_SESSION_TOKEN)

```
GET /api/c_cr/<int:user_id>/
```

Response

<HTTP STATUS CODE 200>

```
{
  "confirmed chat requests": [
    {
      "id": 0,
      "start time": 600,
      "month": 9,
      "day": 2,
      "year": 2023,
      "weekday": 6,
      "message": "hello",
      "accepted": true,
      "sender": 1,
      "receiver": 3
    },
    {
      "id": 1,
      "start time": 840,
      "month": 9,
      "day": 2,
      "year": 2023,
      "weekday": 7,
      "message": "hi lets chat",
      "accepted": false,
      "sender": 1,
      "receiver": 2
    }
  ]
}
```

```
}  
}
```

Add coffee chat request to user (REQUIRES USER SESSION_TOKEN)

POST /api/cr/<int:u_id>/

Request

```
{  
  "receiver id": 2,                #cannot be empty  
  "month": 10,  
  "day": 23,                        #cannot be empty; this field shouldnt be directly written by the user, rather inputted by frontend dep  
  "year": 2023,  
  "weekday": 4,  
  "start time": 840,               #cannot be empty; this is the number of minutes from 12 am to the chosen time (in this example,  
  "message": "i want to chat you"  
}
```

Response

<HTTP STATUS CODE 200>

```
{  
  "id": 1,  
  "name": "Nicole Qiu",  
  "pfp": "<new link url>",  
  "coffee chat requests": [  
    {  
      "id": 1,  
      "start time": 840,  
      "month": 10,  
      "day": 28,  
      "year": 2023,  
      "weekday": 4,  
      "message": "i want to chat you",  
      "accepted": null,  
      "sender": 1,  
      "receiver": 2  
    },  
    {  
      "id": 2,  
      "start time": 840,  
      "month": 10,  
      "day": 23,  
      "year": 2023,  
      "weekday": 4,  
      "message": "i want to chat you",  
      "accepted": null,  
      "sender": 1,  
      "receiver": 2  
    }  
  ]  
}
```

Delete Coffee Chat request (REQUIRES USER SESSION_TOKEN)

DELETE /api/cr/<int:cr_id>/

Response

<HTTP STATUS CODE 200>

```
{  
  "id": 1,  
  "start time": 840,  
  "month": 10,  
  "day": 28,  
  "year": 2023,  
  "weekday": 4,  
  "message": "i want to chat you",  
  "accepted": null,  
  "sender": 1,  
  "receiver": 2  
}
```

```
"receiver": 2
}
```

Accept or deny coffee chat request (REQUIRES USER SESSION_TOKEN)

POST /api/cr/<int:cr_id>/user/<int:user_id>/

```
Request
{
  "accepted": true          #cannot be empty (true or false)
}
```

```
Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "start time": 840,
  "month": 9,
  "day": 2,
  "year": 2023,
  "weekday": 7,
  "message": "hi lets chat"
  "accepted": true,
  "sender": 1,
  "receiver": 2}
}
```

Organization Requests API Specification

Get all received organization requests

GET /api/or/<int:org_id>/

```
Response
<HTTP STATUS CODE 200>
[
  {
    "id": 1,
    "position": "Software Subteam",
    "accepted": null,
    "user id": 1,
    "organization id": 1
  },
  {
    "id": 2,
    "position": "Business Subteam",
    "accepted": null,
    "user id": 1,
    "organization id": 4
  }
]
```

Add organization request to user (REQUIRES USER SESSION_TOKEN)

POST /api/or/<int:sender_id>/

```
Request
{
  "organization id": 2,          #cannot be empty
  "position": "Software Subteam" #cannot be empty
}
```

```

Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "name": "DTI",
  "org pfp": "pfp",
  "organization requests": [
    {
      "id": 1,
      "position": "Software Subteam",
      "accepted": null,
      "user": 1,
      "organization": 1
    },
    {
      "id": 2,
      "position": "Business Subteam",
      "accepted": null,
      "user": 1,
      "organization": 2
    }
  ]
}

```

Delete organization request from user (REQUIRES USER SESSION_TOKEN)

`DELETE /api/or/<int:og_id>/`

```

Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "position": "Business Subteam",
  "accepted": null,
  "user id": 1,
  "organization id": 1
}

```

Accept or deny user request to join organization (REQUIRES ORG SESSION_TOKEN)

`POST /api/or/requests/<int:req_id>/`

```

Request
{
  "accepted": true                                #cannot be empty (true or false)
}

```

```

Response
<HTTP STATUS CODE 200>
{
  "id": 1,
  "position": "Business Subteam",
  "accepted": true,
  "user id": 1,
  "organization id": 1
}

```