## ICT302 Project (Appointment Mate)

### API Documentation

Base URL: `http://api.fyp.darren.li`

#### Authentication

POST `/auth/register`

```

Request:

{

"email": "test02@gmail.com",

"password": "password",

"role": "attendee",

"first\_name": "Darren",

"last\_name": "Li"

}

Response:

{

"email": "test02@gmail.com",

"first\_name": "Darren",

"last\_name": "Li",

"role": "attendee",

"updated\_at": "2018-10-20 16:31:32",

"created\_at": "2018-10-20 16:31:32",

"id": 10,

“status”:OK (can be OK and fail, if status is fail, please provide me with error message, just say duplicate user email or something),

“message”:”this user name is existed” (don’t want message also can, but need status )

}

```

POST `/auth/login`

```

Request:

{

"email": "test02@gmail.com",

"password": "password"

}

Response:

{

"access\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOlwvXC8xMjcuMC4wLjE6ODAwMFwvYXV0aFwvbG9naW4iLCJpYXQiOjE1NDAwMjQ1MzksImV4cCI6MTU0MDA2NzczOSwibmJmIjoxNTQwMDI0NTM5LCJqdGkiOiJsMzRGdzI0YTlOVlF3Q041Iiwic3ViIjoxMCwicHJ2IjoiODdlMGFmMWVmOWZkMTU4MTJmZGVjOTcxNTNhMTRlMGIwNDc1NDZhYSJ9.wzK7AXa6k6868pZTPUv1DhGi7brFP3L6Qb91RumNLxg",

"token\_type": "bearer",

"expires\_in": 43200

}

```

POST `/auth/logout`

```

Response:

{

"message": "Successfully logged out"

}

```

### forgetPassword

Post ‘/auth/forget-password’

{email:”user@gmail.com”}

Response:

{

Status:”OK”,

Message:”email not existed in system” (if fail, please provide the message

}

#### Profile

GET `/auth/me`

```

Response:

{

"id": 10,

"email": "test02@gmail.com",

"first\_name": "Darren",

"last\_name": "Li",

"role": "attendee",

"created\_at": "2018-10-20 16:31:32",

"updated\_at": "2018-10-20 16:31:32"

}

```

POST `/auth/me`

```

Request:

{

"first\_name": "Derek"

}

Response:

{

"id": 10,

"email": "test02@gmail.com",

"first\_name": "Derek",

"last\_name": "Li",

"role": "attendee",

"created\_at": "2018-10-20 16:31:32",

"updated\_at": "2018-10-20 16:39:39"

}

```

#### Users

GET `/users`

```

Response:

[

{

"id": 1,

"email": "admin@gmail.com",

"first\_name": "Bennie",

"last\_name": "Rowe",

"role": "convenor",

"created\_at": "2018-10-20 16:27:15",

"updated\_at": "2018-10-20 16:27:15"

},

...

]

```

POST `/users`

```

Request:

{

"email": "test003@gmail.com",

"password": "password",

"first\_name": "Gowen",

"last\_name": "Smith",

"role": "attendee"

}

Response:

{

"email": "test003@gmail.com",

"first\_name": "Gowen",

"last\_name": "Smith",

"role": "attendee",

"updated\_at": "2018-10-20 16:46:03",

"created\_at": "2018-10-20 16:46:03",

"id": 11

}

```

UPDATE `/users/{id}`

```

Request:

{

"first\_name": "Chad"

}

Response:

{

"id": 11,

"email": "test003@gmail.com",

"first\_name": "Chad",

"last\_name": "Smith",

"role": "attendee",

"created\_at": "2018-10-20 16:46:03",

"updated\_at": "2018-10-20 16:47:38"

}

```

DELETE `/users/{id}`

```

Response: 200 OK

```

#### Locations

GET `/locations`

```

Response:

[

{

"id": 1,

"name": "Jaskolski Locks",

"address": "47771 Burdette Light\nLake Erna, WI 05006-6933",

"postal": "768936",

"created\_at": "2018-10-20 16:27:15",

"updated\_at": "2018-10-20 16:27:15"

},

...

]

```

POST `/locations`

```

Request:

{

"name": "JasGelski Locks",

"address": "47792 Burdette Light 05006-6933",

"postal": "768986"

}

Response:

{

"name": "JasGelski Locks",

"address": "47792 Burdette Light 05006-6933",

"postal": "768986",

"updated\_at": "2018-10-20 16:49:20",

"created\_at": "2018-10-20 16:49:20",

"id": 6

}

```

UPDATE `/locations/{id}`

```

Request:

{

"name": "JasGelski Locks 2"

}

Response:

{

"id": 6,

"name": "JasGelski Locks 2",

"address": "47792 Burdette Light 05006-6933",

"postal": "768986",

"created\_at": "2018-10-20 16:49:20",

"updated\_at": "2018-10-20 16:49:44"

}

```

DELETE `/locations/{id}`

```

Response: 200 OK

```

#### Appointments

GET `/appointments`

```

Response:

[

{

"id": 1,

"attendee\_id": 8,

"location\_id": 5,

"starts\_at": "2018-02-21 04:33:01",

"ends\_at": "2018-02-21 05:03:01",

"confirmed": 1,

"created\_at": "2018-10-20 16:27:15",

"updated\_at": "2018-10-20 16:27:15"

},

...

]

```

POST `/appointments`

```

Request:

{

"attendee\_id": 7,

"location\_id": 3,

"starts\_at": "2018-10-21 14:30:00",

"ends\_at": "2018-10-21 15:00:00",

"confirmed": 1

}

Response:

{

"attendee\_id": 7,

"location\_id": 3,

"starts\_at": "2018-10-21 14:30:00",

"ends\_at": "2018-10-21 15:00:00",

"confirmed": 1,

"updated\_at": "2018-10-20 16:50:55",

"created\_at": "2018-10-20 16:50:55",

"id": 11

}

```

UPDATE `/appointments/{id}`

```

Request:

{

"starts\_at": "2018-10-21 10:30:00",

"ends\_at": "2018-10-21 11:00:00"

}

Response:

{

"id": 11,

"attendee\_id": 7,

"location\_id": 3,

"starts\_at": "2018-10-21 10:30:00",

"ends\_at": "2018-10-21 11:00:00",

"confirmed": 1,

"created\_at": "2018-10-20 16:50:55",

"updated\_at": "2018-10-20 16:51:28"

}

```

DELETE `/appointments/{id}`

```

Response: 200 OK

```

###-------------------------------------- AppointmentCreate page--------------------------

0------------------------get-all-appointment-slot’ for user----------------------

Get all the available appointment in that month (available)

Post ‘/auth/get-all-appointment-slot’;

{month:”09”,

year:’2018’,

}

Response:

{slots:[

{monday:

{status:'available',

day:27,

isInMonth:'YES',

slots:[{

id:appoinmentID

location:location

status:

startTime:'10:30 am',

endTime:'11:00 am'

},{

id:appoinmentID

location:location

status:

startTime:'11:00 am',

endTime:'11:30 am'

} ] },

Tuesday:{},

……..

}]

(response data will be ResultDTO<Week>, in side week, we have Monday, Tuesday… to Sunday (object, or class),

Inside Monday, we have {status:'available',(**indicate whether have any available slot for user** to book)

day:27, (date of the month)

isInMonth:'YES', (inside selected month,)

slots:[{ (list of available slot of that day

id:appoinmentID

location:location

status:

startTime:'10:30 am',

endTime:'11:00 am'

location:string

},{

startTime:'11:00 am',

endTime:'11:30 am'

location:string

} ] },

Status:”OK”,

Message:”email not existed in system” (if fail, please provide the message)

}

-------------------------------------get all appointment list ---------------

this.getAllappointmentList = function(id){

return Restangular.one(prefix + '/get-all-appointmentList'+'/'+id).get();

};

Path:????/get-all-appointmentList

Post {

userEmail

}

//return list <appointmentDTO>

Response[

{

appointmentID:int

statusName:string

startTime:date

endTime:date

location:string

}

]

----------------------------------------get all appointment status--------------------

this.getAllappointmentStatus = function(id){

return Restangular.one(prefix + '/get-all-appointment-status').get();

};

Post path/???

Response:

[

{statusCode:string

statusName:string

}

]

this.getAllAppointmentListBySearch = function (params) {

return Restangular.all(prefix + '/get-all-appointment-list-by-search').post(params);

};

Post path/???

Request{

statusCode:string

startTime:date

endTime:date,

pageSide:integer

pageNumber:integer

userID:userID

}

//--------------------------according to search criteria to return the appointment list----------------

//return list <appointmentDTO>

Response[

{

appointmentID:int

statusName:string

startTime:date

endTime:date

location:string

}

]

this.downloadCsv = function (month){

return Restangular.all(prefix + '/download-all-appointment-list').post(month);

};

Post path/???

Request{

statusCode:string

startTime:date

endTime:date,

pageSide:integer

pageNumber:integer

userID:userID

}

//------------------------according to search criteria to return the appointment list in CSV format

//return list <appointmentDTO>

Response[

{

appointmentID:int

statusName:string

startTime:date

endTime:date

location:string

}

]

----------------------------get all available slots for that day-----------------

this.getAllAppointmentListByDay=function (param){

return Restangular.all(prefix + '/get-all-available-slot-for-day').post(param);

};

var param={

startTime:date,

endTime: date }

//return list of available slot

response

[{

statusName:

startTime:date,

endTime:date,

location:

id:

}]

--------------------------------------cancel appointment----------------------------------------

this.cancelAppointment=function(id){

return Restangular.all(prefix + '/cancel-appoinment').post(id);

}

//-appointment id

Param{

Id:integer}

Response:{

Status: YES/NO

Message: string

} indicate whether function success or fail

-----change appointment

So I will pass current appointment ID as previousAppoinmentID, and list <integer> of selected appointment id, change status current appointment and use userID as reference to user table or using user id (prefer use user id

this.changeAppointment=function(param){

return Restangular.all(prefix + '/change-appoinment').post(param);

}

var param={

currentAppoinmentID:Integer,

selectedappointmentIDs:List<integer>

userID:userID

}

Response={

Status:’’YES/NO’,

Message:string}

//-------------------------------------Create appointment

I will return list of selected appointment ID together with userEmail or userID to backend set default status as pending

var param={

selectedSlots:List <integer>,

userID: userID

}

Response{

Status:’YES/NO’,

Message:}

----------------------------------- Retrieve user detail by id------------------------------------

this.getUserDetail = function (param) {

return Restangular.all(prefix + ' /get-user-detail-by-id’).post(param);

};

Path:???

var param={

userID: userID

}

Response:

{

"id": 10,

"email": "test02@gmail.com",

"first\_name": "Darren",

"last\_name": "Li",

"role": "attendee",

"created\_at": "2018-10-20 16:31:32",

"updated\_at": "2018-10-20 16:31:32"

Status:’Yes’

Message:string

}

--------------------------------update user detail allow user to update all his detail,

this.updateUserDetail=function(param){

return Restangular.all(prefix + '/update-user-detail').post(param);

}

var param={

userId:integer

firstName:string,

lastName:string,

userEmail:string,

roleName: string,

contactNo:Integer,

};

Response{

Status:’YES/NO’,

Message:}

/// -------------------get total number of pending appointments for admin

this.getAllPendingAppointments = function () {

return Restangular.one(prefix + '/get-all-pengding-appointments').get();

};

Response{

totalnum:integer,

status:’’YES/NO”,

message:string}

//-----------------------get all users ---------------------------------

this.getAllUser = function () {

return Restangular.all(prefix + '/ get -all-user').post();

};

Response: (list of user with status )

{

Status:’YES’

Message:string,

userList[{

"id": 10,

"email": "test02@gmail.com",

"first\_name": "Darren",

"last\_name": "Li",

"role": "attendee",

"created\_at": "2018-10-20 16:31:32",

"updated\_at": "2018-10-20 16:31:32"

Status:’Yes’

Message:string

}]

}

-----------------------------Create user--------------

this.createUserDetail=function(param){

return Restangular.all(prefix + '/create-user').post(param);

}

var param={

firstName:String,

lastName:String,

userEmail:String,

roleName:String,

contactNo:Integer

};

Response{

Status:’YES’

Message:string}

###-------------------------------------- AppointmentCreate page--------------------------

0------------------------get-all-appointment-slot-admin’ for admin ----------------------

Get all the available appointment in that month (all)

Post ‘/auth/get-all-appointment-slot- admin’ ’;

{month:”09”,

year:’2018’}

Response:

{slots:[

{monday:

{

userList:List<username> list of user who has appointment in that day

day:27,

isInMonth:'YES',

slots:[{

id:appoinmentID

location:location

status:

startTime:'10:30 am',

endTime:'11:00 am'

},{

id:appoinmentID

location:location

status:

startTime:'11:00 am',

endTime:'11:30 am'

} ] },

Tuesday:{},

……..

}]

(response data will be ResultDTO<Week>, in side week, we have Monday, Tuesday… to Sunday (object, or class),

Inside Monday, we have {status:'available',(**indicate whether have any available slot for user** to book)

day:27, (date of the month)

isInMonth:'YES', (inside selected month,)

slots:[{ (list of available slot of that day

id:appoinmentID

location:location

status:

startTime:'10:30 am',

endTime:'11:00 am'

location:string

},{

startTime:'11:00 am',

endTime:'11:30 am'

location:string

} ] },

Status:”OK”,

Message:”email not existed in system” (if fail, please provide the message)

}

-----------------------save and update appointment (admin approve and create appointment)

Pass list of appointment, please create a new appointment for those appointment without id

Var param= [{ (list of available slot of that day

id:appoinmentID

location:location

status:

startTime:'10:30 am',

endTime:'11:00 am'

location:string

}

Response{

Status:’YES’

message}