

- Create new staff
 - Description : create new staff in database with variable in request body.
 - Path : /company/staff
 - Method : POST
 - Headers : Bearer <credentials>
 - Request Body : JSON

```
{
  firstName : "some_firstname",
  lastName : "some_lastname",
  password : "some_password",
  salary : 1000,
  role : "employer",
}
```
 - Responses
 - Successful
 - Response Body : JSON

```
{
  Id : "123456",
  firstName : "some_firstname",
  lastName : "some_lastname",
  salary : 1000,
  role : "employer",
}
```
 - Status Code : 201 Created
 - Incorrect
 - Status Code : 400 Bad Request
 - Response Body : JSON

```
{  
  message : "{ field } is invalid form (or require)"  
}
```

- Status Code : 401 Unauthorized
 - Response Body : JSON

```
{  
  Message : "Your role doesn't have permission in  
  this action"  
}
```

- Get all staffs
 - Description : Return data of all staff (except password) to client-side
 - Path : /company/staff
 - Method : GET
 - Headers : Bearer <credentials>
 - Responses

- Successful

- Response Body : JSON Array

```
[  
  {  
    Id : "1",  
    firstName : "some_firstname",  
    lastName : "some_lastname",  
    salary : 1000,  
    role : "employer",  
  },  
  {  
    Id : "2",  
    firstName : "some_firstname2",  
    lastName : "some_lastname2",  
    salary : 2000,  
    role : "employer",  
  }  
]
```

- Status Code : 200 OK

- Incorrect

- Status Code : 401 Unauthorized
 - Response Body : JSON

```
{
```

```
  Message : "Your role doesn't have permission in  
  this action."
```

```
}
```

- Get staff followed by fields
 - Description : Return staff that filter by field. Return 404 if length of expected user is 0.
 - Path : /company/staffByFields?firstname="ABC"&lastname="GJH"
 - Method : GET
 - Headers : Bearer <credentials>
 - Request Parameter : firstname ,lastname or other needed fields
 - Responses : JSON
 - Successful
 - Status Code : 200 OK
 - Response Body : JSON Array


```
{
                    [...user from filter following parameters]
                  }
```
 - Incorrect
 - Status Code : 404 Not found
 - Response Body : JSON


```
{
                        Message : "Not found any user"
                      }
```
 - Status Code :


```
Response Body : JSON
{
  Message : "Your role doesn't have permission in this action."
}
```

- Update staff data

- Description : update an staff data
- Path : /company/staff?id="123456"
- Method : PUT
- Headers : Bearer <credentials>
- Request Parameter : ID of staff
- Request Body : JSON (require 1 field for minimum)

```
{  
  new_firstName : "new_first_name",  
  New_lastName : "new_last_name",  
  Role : "new_role",  
  Salary : "new_salary"  
}
```

- Responses

- Successful

- Response Body : JSON

```
{ ...new staff data }
```

- Status Code : 200 OK

- Incorrect

- Status Code : 401 Unauthorized

- Response Body : JSON

```
{ message : "Your role doesn't have permission in  
this action. }
```

- Status Code : 404 Not found

- Response Body : JSON

```
{ message : "{ field } is invalid form.", }
```

- Delete
 - Description : Delete by staff ID.
 - Path : /company/staff?id="213456"
 - Method : DELETE
 - Headers : Bearer <credentials>
 - Request Params : id of staff whom HR want to delete
 - Responses
 - Successful
 - Response Body : JSON

```
{ message : "Deleting successfully" }
```
 - Status Code : 200 OK
 - Incorrect
 - Status Code : 404 Not found
 - Response Body

```
{ message : "Database don't have this user." }
```
 - Status Code : 401 Unauthorized
 - Response Body : JSON

```
{  
  
  message : "Your role doesn't have permission in  
  this action."  
  
}
```

- Login

- Description : Login user to company dashboard
- Path : /login
- METHOD : POST
- Request Body : JSON

```
{  
  Id : "123456",  
  password : "qasknxksnckdjnc",  
}
```

- Response

- Successful

- Status Code : 200 OK

- Response Body : JSON

```
{  
  accessToken : accessToken,  
  refreshToken : refreshToken  
}
```

- Incorrect

- Status Code : 403 Forbidden

- Response Body : JSON

```
{  
  message : "your id or password is wrong."  
}
```


- Reset password
 - Description : Must login before. Reset password of user (other people cannot use this api except owner id) and redirect to login again.
 - Path : /company/staff/password
 - Method : PUT
 - Headers : Bearer <credentials>
 - Request Body : JSON

```
{  
  
  oldPassword : "asdasda",  
  
  newPassword : "egrmrg"  
}
```
 - Response body : JSON
 - Successful
 - Status Code : 200 OK
 - Incorrect
 - Status Code : 401 Unauthorized