

Easyroom

Users BE Services

End-point: <http://185.120.37.29:1273/eaglebeds-extranet>

Create User

Request

Path: /easyroom/users

Method: POST

Body Request: {

"roleId": Long,	[Mandatory]	[Positive]
"supervisorId": Long,	[Mandatory]	[Positive]
"hotelId": Long,	[Mandatory]	[Positive]
"cityId": Long,		
"countryId": Long,		
"name": "String",	[Mandatory]	
"surname": "String",	[Mandatory]	
"password": "String",	[Mandatory]	
"sex": "String",	[Mandatory]	
"birthday": "String YYYY-MM-DD",		
"placeOfBirth": "String",		
"address": "String",		
"email": "String",	[Mandatory]	
"telephoneNo": "String",	[Mandatory]	
"celullarNo": "String",	[Mandatory]	
"faxNo": "String",	[Mandatory]	
"status": "String"	[Mandatory]	

}

Response

In case of success returns:

HTTP_CODE: 201

Body:

```
{
  "code": "OK",
  "message": "User successfully Created!",
  "props": {
    "Username": "String"
  }
}
```

In case of failure returns:

HTTP_CODE: 400

Body:

```
{
  "code": "KO",
  "message": "Validation failed",
  "props": {

```

```

        "E-mail": "String",
        "props": []
    }
}

HTTP_CODE: 409
Body:
{
    "code": "KO",
    "message": "This email is already registered!",
    "props": {
        "E-mail": "String"
    }
}

HTTP_CODE: 405
Body:
{
    " timestamp ": 1583531483348,
    "status": 405,
    "error": "Method Not Allowed",
    "exception": "org.springframework.web.HttpRequestMethodNotSupportedException",
    "message": "Request method 'POST' not supported",
    "path": "/easyroom/users"
}

HTTP_CODE: 500
Body:
{
    "code": "KO",
    "message": "Internal Server Error",
    "props": {}
}

```

Description

The service is responsible to register(create) a new user .

Get User

Request

```

Path:      /easyroom/users/{username(String)}
Method:    GET

```

Response

In case of success returns:

```

HTTP_CODE: 200
Body:
{
    "name": "String",
    "surname": "String",
    "username": "String",
    "email": "String",
    "birthday": "String",

```

```

    "placeOfBirth": "String",
    "sex": "String",
    "address": "String",
    "celullarNo": "String",
    "telephoneNo": "String",
    "faxNo": "String",
    "status": "String",
    "roleId": Long,
    "supervisorId": Long,
    "cityId": Long,
    "hotelId": Long,
    "countryId": Long,
    "chngPwdFirstLoginFlag": Boolean,
    "confirmedAccFlag": Boolean,
    "userRegDt": "String YYYY-MM-DD HH:mm:ss"
}

```

In case of failure returns:

HTTP_CODE: 404

Body:

```

{
    "code": "KO",
    "message": "Unable to find User!",
    "props": {
        "username": "String"
    }
}

```

HTTP_CODE: 500

Body:

```

{
    "code": "KO",
    "message": "Internal Server Error",
    "props": {}
}

```

Description

The service is responsible to get a user by it's username .

Update User

Request

Path: /easyroom/users/{username(String)}

Method: PUT

Body Request: {

"roleId": Long,	[Mandatory]	[Positive]
"supervisorId": Long,	[Mandatory]	[Positive]
"hotelId": Long,	[Mandatory]	[Positive]
"cityId": Long,	[Mandatory]	[Positive]
"countryId": Long,	[Mandatory]	[Positive]
"name": "String",	[Mandatory]	
"surname": "String",	[Mandatory]	
"password": "String",	[Mandatory]	

```

        "sex": "String", [Mandatory]
        "birthday": "String YYYY-MM-DD", [Mandatory]
        "placeOfBirth": "String", [Mandatory]
        "address": "String", [Mandatory]
        "email": "String", [Mandatory]
        "telephoneNo": "String", [Mandatory]
        "celullarNo": "String", [Mandatory]
        "faxNo": "String", [Mandatory]
        "status": "String" [Mandatory]
    }
}

```

Response

In case of success returns:

HTTP_CODE: 200

Body:

```

{
    "code": "OK",
    "message": "User data successfully updated!",
    "props": {
        "Username": "String"
    }
}

```

In case of failure returns:

HTTP_CODE: 400

Body:

```

{
    "code": "KO",
    "message": "Validation failed",
    "props": {
        "Username": "String",
        "props": []
    }
}

```

HTTP_CODE: 404

Body:

```

{
    "code": "KO",
    "message": "User data update failed, unable to find user!",
    "props": {
        "Username": "String"
    }
}

```

HTTP_CODE: 405

Body:

```

{
    "timestamp ": 1583531483348,
    "status": 405,
    "error": "Method Not Allowed",
    "exception":
    "org.springframework.web.HttpRequestMethodNotSupportedException",
    "message": "Request method 'PUT' not supported",
    "path": "/easyroom/users/{username(String)}"
}

```

```
}

HTTP_CODE: 500
Body:
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible to update(change) user's data by it's username .

Delete User

Request

Path: /easyroom/users/{username(**String**)}
Method: DELETE

Response

In case of success returns:

```
HTTP_CODE: 200
Body:
{
  "code": "OK",
  "message": " User successfully deleted!",
  "props": {
    "Username": "String"
  }
}
```

In case of failure returns:

```
HTTP_CODE: 404
Body:
{
  "code": "KO",
  "message": "Failed to delete user, unable to find user!",
  "props": {
    "username": "String"
  }
}
```

```
HTTP_CODE: 500
Body:
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible to delete a user by it's username .

Get All Users

Request

Path: /easyroom/users
Method: GET

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
[
  {
    "name": "String",
    "surname": "String",
    "username": "String",
    "email": "String",
    "birthday": "String",
    "placeOfBirth": "String",
    "sex": "String",
    "address": "String",
    "celullarNo": "String",
    "telephoneNo": "String",
    "faxNo": "String",
    "status": "String",
    "roleId": Long,
    "supervisorId": Long,
    "cityId": Long,
    "hotelId": Long,
    "countryId": Long,
    "chngPwdFirstLoginFlag": Boolean,
    "confirmedAccFlag": Boolean,
    "userRegDt": "String YYYY-MM-DD HH:mm:ss"
  },
  {
    "name": "String",
    "surname": "String",
    "username": "String",
    "email": "String",
    "birthday": "String",
    "placeOfBirth": "String",
    "sex": "String",
    "address": "String",
    "celullarNo": "String",
    "telephoneNo": "String",
    "faxNo": "String",
    "status": "String",
    "roleId": Long,
    "supervisorId": Long,
    "cityId": Long,
    "hotelId": Long,
    "countryId": Long,
    "chngPwdFirstLoginFlag": Boolean,
    "confirmedAccFlag": Boolean,
    "userRegDt": "String YYYY-MM-DD HH:mm:ss"
  }
]
```

```
]
```

In case of failure returns:

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "No users found!",
  "props": {}
}
```

HTTP_CODE: 500

Body:

```
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible to get all available users .

Get Supervisor's Users

Request

Path: /easyroom/users/sv/{username(*String*)}

Method: GET

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
[
  {
    "name": "String",
    "surname": "String",
    "username": "String",
    "email": "String",
    "birthday": "String",
    "placeOfBirth": "String",
    "sex": "String",
    "address": "String",
    "celullarNo": "String",
    "telephoneNo": "String",
    "faxNo": "String",
    "status": "String",
    "roleId": Long,
    "supervisorId": Long,
    "cityId": Long,
    "hotelId": Long,
    "countryId": Long,
    "chngPwdFirstLoginFlag": Boolean,
  }
]
```

```

        "confirmedAccFlag": Boolean,
        "userRegDt": "String YYYY-MM-DD HH:mm:ss"
    },
    {
        "name": "String",
        "surname": "String",
        "username": "String",
        "email": "String",
        "birthday": "String",
        "placeOfBirth": "String",
        "sex": "String",
        "address": "String",
        "cellularNo": "String",
        "telephoneNo": "String",
        "faxNo": "String",
        "status": "String",
        "roleId": Long,
        "supervisorId": Long,
        "cityId": Long,
        "hotelId": Long,
        "countryId": Long,
        "chngPwdFirstLoginFlag": Boolean,
        "confirmedAccFlag": Boolean,
        "userRegDt": "String YYYY-MM-DD HH:mm:ss"
    }
]

```

In case of failure returns:

HTTP_CODE: 404

Body:

```

{
    "code": "KO",
    "message": "No users found!",
    "props": {}
}

```

HTTP_CODE: 500

Body:

```

{
    "code": "KO",
    "message": "Internal Server Error",
    "props": {}
}

```

Description

The service is responsible to get all supervisor's users, a supervisor can be an admin user or a supervisor user.

For example if you use this service to get:

- 1) admin's users than it will return all supervisor users.
- 2) supervisor's users than it will return all receptionist users .

Get Users By Role

Request

Path: /easyroom/users/role/{roleId(Long)}
Method: GET

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
[
  {
    "name": "String",
    "surname": "String",
    "username": "String",
    "email": "String",
    "birthday": "String",
    "placeOfBirth": "String",
    "sex": "String",
    "address": "String",
    "celullarNo": "String",
    "telephoneNo": "String",
    "faxNo": "String",
    "status": "String",
    "roleId": Long,
    "supervisorId": Long,
    "cityId": Long,
    "hotelId": Long,
    "countryId": Long,
    "chngPwdFirstLoginFlag": Boolean,
    "confirmedAccFlag": Boolean,
    "userRegDt": "String YYYY-MM-DD HH:mm:ss"
  },
  {
    "name": "String",
    "surname": "String",
    "username": "String",
    "email": "String",
    "birthday": "String",
    "placeOfBirth": "String",
    "sex": "String",
    "address": "String",
    "celullarNo": "String",
    "telephoneNo": "String",
    "faxNo": "String",
    "status": "String",
    "roleId": Long,
    "supervisorId": Long,
    "cityId": Long,
    "hotelId": Long,
    "countryId": Long,
    "chngPwdFirstLoginFlag": Boolean,
    "confirmedAccFlag": Boolean,
    "userRegDt": "String YYYY-MM-DD HH:mm:ss"
  }
]
```

```
]
```

In case of failure returns:

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "No users found!",
  "props": {}
}
```

HTTP_CODE: 500

Body:

```
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible to get all users by role type .

User Authentication

Request

Path: /easyroom/users/authentication

Method: POST

Body Request: {

```
    "emailOrUsername": "String",    [Mandatory]
    "userpass": "String"            [Mandatory]
}
```

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
{
  "code": "OK",
  "message": " Valid Credentials!",
  "props": {
    "Username": "String"
  }
}
```

In case of failure returns:

HTTP_CODE: 400

Body:

```
{
  "code": "KO",
  "message": "Validation failed",
  "props": {
    "Username": "String",

```

```

        "props": []
    }
}

HTTP_CODE: 404
Body:
{
    "code": "KO",
    "message": "Invalid Credentials!",
    "props": {
        "Username": "String"
    }
}

HTTP_CODE: 405
Body:
{
    "timestamp ": 1583531483348,
    "status": 405,
    "error": "Method Not Allowed",
    "exception":
    "org.springframework.web.HttpRequestMethodNotSupportedException",
    "message": "Request method 'PUT' not supported",
    "path": "/easyroom/users"
}

HTTP_CODE: 500
Body:
{
    "code": "KO",
    "message": "Internal Server Error",
    "props": {}
}

```

Description

The service is responsible to check if the user's credentials are valid or invalid, in such way to allow or disallow user's log on .

Change User's Password

Request

```

Path:          /easyroom/users/chngpwd
Method:        PUT
Body Request: {
                "username": "String",           [Mandatory]
                "userpass": "String"            [Mandatory]
            }

```

Response

In case of success returns:

```
HTTP_CODE: 200
Body:
{
  "code": "OK",
  "message": "User's password successfully changed!",
  "props": {
    "Username": "String"
  }
}
```

In case of failure returns:

```
HTTP_CODE: 400
Body:
{
  "code": "KO",
  "message": "Validation failed",
  "props": {
    "Username": "String",
    "props": []
  }
}
```

```
HTTP_CODE: 404
Body:
{
  "code": "KO",
  "message": "Failed to change user's password, unable to find User!",
  "props": {
    "Username": "String"
  }
}
```

```
HTTP_CODE: 405
Body:
{
  "timestamp": 1583531483348,
  "status": 405,
  "error": "Method Not Allowed",
  "exception": "org.springframework.web.HttpRequestMethodNotSupportedException",
  "message": "Request method 'PUT' not supported",
  "path": "/easyroom/users"
}
```

```
HTTP_CODE: 500
Body:
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible update(change) user's password.

Confirm User's Account

Request

Path: /easyroom/users/acccconfirm
Method: PUT
Body Request: {
 "jwt": "String" [Mandatory]
}

Response

In case of success returns:

HTTP_CODE: 200
Body:
{
 "code": "OK",
 "message": "User Account Successfully Confirmed!",
 "props": null
}

In case of failure returns:

HTTP_CODE: 400
Body:
{
 "code": "KO",
 "message": "Validation failed",
 "props": {
 "Username": "String",
 "props": []
 }
}

HTTP_CODE: 400
Body:
{
 "code": "KO",
 "message": "Token Expired!",
 "props": {}
}

HTTP_CODE: 404
Body:
{
 "code": "KO",
 "message": "Failed to confirm user's account, unable to find User!",
 "props": {
 "Username": "String"
 }
}

HTTP_CODE: 405
Body:
{
 "timestamp": 1583531483348,
 "status": 405,
 "error": "Method Not Allowed",
 "exception":
 "org.springframework.web.HttpRequestMethodNotSupportedException",
 "message": "Request method 'PUT' not supported",
 "path": "/easyroom/users"

```

}

HTTP_CODE: 500
Body:
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}

```

Description

The service is responsible to confirm user's account after the user has received the e-mail for confirmation post account creation .

Change User's Status

Request

```

Path: /easyroom/users/chngstatus
Method: PUT
Body Request: {
  "username": "String", [Mandatory]
  "status": "String" [Mandatory]
}

```

Response

In case of success returns:

```

HTTP_CODE: 200
Body:
{
  "code": "OK",
  "message": "User's status successfully changed!",
  "props": {
    "Username": "String"
  }
}

```

In case of failure returns:

```

HTTP_CODE: 400
Body:
{
  "code": "KO",
  "message": "Validation failed",
  "props": {
    "Username": "String",
    "props": []
  }
}

```

```

HTTP_CODE: 404
Body:
{
  "code": "KO",

```

```

        "message": "Failed to change user's status, unable to find User!",
        "props": {
            "Username": "String"
        }
    }

HTTP_CODE: 405
Body:
{
    "timestamp ": 1583531483348,
    "status": 405,
    "error": "Method Not Allowed",
    "exception":
    "org.springframework.web.HttpRequestMethodNotSupportedException",
    "message": "Request method 'PUT' not supported",
    "path": "/easyroom/users"
}

HTTP_CODE: 500
Body:
{
    "code": "KO",
    "message": "Internal Server Error",
    "props": {}
}

```

Description

The service is responsible update(change) user's status .

Recover User's Password

Request

```

Path:          /easyroom/users/recoverpwd
Method:        PUT
Body Request: {
                "email": "String",
                }

```

[Mandatory]

Response

In case of success returns:

```

HTTP_CODE: 200
Body:
{
    "code": "OK",
    "message": "Token Successfully Generated!",
    "props": {
        "jwt": "String",
        "email": "String"
    }
}

```

In case of failure returns:

HTTP_CODE: 400

Body:

```
{
  "code": "KO",
  "message": "Validation failed",
  "props": {
    "Username": "String",
    "props": []
  }
}
```

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "Unable to find User email:[ermal.asllanaj15@jakala.com]",
  "props": {
    "Username": "String"
  }
}
```

HTTP_CODE: 405

Body:

```
{
  " timestamp ": 1583531483348,
  "status": 405,
  "error": "Method Not Allowed",
  "exception":
    "org.springframework.web.HttpRequestMethodNotSupportedException",
  "message": "Request method 'PUT' not supported",
  "path": "/easyroom/users"
}
```

HTTP_CODE: 500

Body:

```
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible to recover user's password. This service send an email to the user with a password reset link. The reset link is valid only for 24 hours.

Change User's Password on Recover Request

Request

Path: /easyroom/users/recover/chngpwd

Method: PUT

Body Request: {

```
  "jwt": "String", [Mandatory]
  "newPassword": "String", [Mandatory]
}
```


Response

In case of success returns:

HTTP_CODE: 200

Body:

```
{
  "code": "OK",
  "message": "Password successfully changed!",
  "props": null
}
```

In case of failure returns:

HTTP_CODE: 400

Body:

```
{
  "code": "KO",
  "message": "Validation failed",
  "props": {
    "Username": "String",
    "props": []
  }
}
```

HTTP_CODE: 400

Body:

```
{
  "code": "KO",
  "message": "Token Expired!",
  "props": {
    "inpEmail": "String"
  }
}
```

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "Unable to find User email:[ermal.asllanaj15@jakala.com]",
  "props": {
    "Username": "String"
  }
}
```

HTTP_CODE: 405

Body:

```
{
  " timestamp ": 1583531483348,
  "status": 405,
  "error": "Method Not Allowed",
  "exception":
    "org.springframework.web.HttpRequestMethodNotSupportedException",
  "message": "Request method 'PUT' not supported",
  "path": "/easyroom/users"
}
```

HTTP_CODE: 500

Body:

```
{
  "code": "KO",
  "message": "Internal Server Error",
  "props": {}
}
```

Description

The service is responsible to change user's password. This service changes the user's password after he has clicked the received email with a password reset link. The reset link is valid only for 24 hours.