

Easyroom

User's Roles BE Services

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

Create User Role

Request

Path: /easyroom/roles

Method: POST

```
Body Request: {
  "roleName": "String",
  "roleDescription": "String",
  "flCreateUser": Boolean,
  "flReadUser": Boolean,
  "flUpdateUser": Boolean,
  "flDeleteUser": Boolean,
  "flCreateBooking": Boolean,
  "flReadBooking": Boolean,
  "flUpdateBooking": Boolean,
  "flDeleteBooking": Boolean,
  "flReadReports": Boolean,
  "flUpdateReports": Boolean,
  "flDeleteReports": Boolean
}
```

Response

In case of success returns:

HTTP_CODE: 201

Body:

```
{
  "code": "OK",
  "message": "Role Successfully Created!",
  "props": {
    "Role ID": Long
  }
}
```

In case of failure returns:

HTTP_CODE: 400

Body:

```
{
  "code": "KO",
  "message": "Validation failed",
  "props": {
    "inpEmail": "String",
    "errors": [
      "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 create a new user role .

Get User Role

Request

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

Response

In case of success returns:

```

HTTP_CODE: 200
Body:
{
    "id": "Long",
    "roleName": "String",
    "roleDescription": "String",
    "flCreateUser": Boolean,
    "flReadUser": Boolean,
    "flUpdateUser": Boolean,
    "flDeleteUser": Boolean,
    "flCreateBooking": Boolean,
    "flReadBooking": Boolean,
    "flUpdateBooking": Boolean,
    "flDeleteBooking": Boolean,
    "flReadReports": Boolean,
    "flUpdateReports": Boolean,
    "flDeleteReports": Boolean
}

```

In case of failure returns:

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "Unable to find User Role!",
  "props": {
    "roleId": "Long"
  }
}
```

HTTP_CODE: 500

Body:

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

Description

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

Update User Role

Request

Path: /easyroom/roles/{roleId(Long)}

Method: PUT

Body Request: {

```
  "roleDescription": "String",           [Mandatory]
  "flCreateUser": Boolean,               [Mandatory]
  "flReadUser": Boolean,                 [Mandatory]
  "flUpdateUser": Boolean,               [Mandatory]
  "flDeleteUser": Boolean,               [Mandatory]
  "flCreateBooking": Boolean,            [Mandatory]
  "flReadBooking": Boolean,              [Mandatory]
  "flUpdateBooking": Boolean,            [Mandatory]
  "flDeleteBooking": Boolean,            [Mandatory]
  "flReadReports": Boolean,              [Mandatory]
  "flUpdateReports": Boolean,            [Mandatory]
  "flDeleteReports": Boolean             [Mandatory]
}
```

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
{
  "code": "OK",
  "message": "User role data successfully updated!",
  "props": {
```

```
        "roleId": Long
    }
}
```

In case of failure returns:

HTTP_CODE: 400

Body:

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

HTTP_CODE: 404

Body:

```
{
    "code": "KO",
    "message": "User role data update failed, unable to find user role!",
    "props": {
        "roleId": "Long"
    }
}
```

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/{username(String)}"
}
```

HTTP_CODE: 500

Body:

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

Description

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

Delete User Role

Request

Path: /easyroom/roles/{roleId(Long)}
Method: DELETE

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
{
  "code": "OK",
  "message": " User role successfully deleted!",
  "props": {
    "roleId": Long
  }
}
```

In case of failure returns:

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "Failed to delete user role, unable to find user role!",
  "props": {
    "roleId": "Long"
  }
}
```

HTTP_CODE: 500

Body:

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

Description

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

Get All User's Roles

Request

Path: /easyroom/roles
Method: GET

Response

In case of success returns:

HTTP_CODE: 200

Body:

```
[
  {
    "id": "Long",
    "roleName": "String",
    "roleDescription": "String",
    "flCreateUser": Boolean,
    "flReadUser": Boolean,
    "flUpdateUser": Boolean,
    "flDeleteUser": Boolean,
    "flCreateBooking": Boolean,
    "flReadBooking": Boolean,
    "flUpdateBooking": Boolean,
    "flDeleteBooking": Boolean,
    "flReadReports": Boolean,
    "flUpdateReports": Boolean,
    "flDeleteReports": Boolean
  },
  {
    "id": "Long",
    "roleName": "String",
    "roleDescription": "String",
    "flCreateUser": Boolean,
    "flReadUser": Boolean,
    "flUpdateUser": Boolean,
    "flDeleteUser": Boolean,
    "flCreateBooking": Boolean,
    "flReadBooking": Boolean,
    "flUpdateBooking": Boolean,
    "flDeleteBooking": Boolean,
    "flReadReports": Boolean,
    "flUpdateReports": Boolean,
    "flDeleteReports": Boolean
  }
]
```

In case of failure returns:

HTTP_CODE: 404

Body:

```
{
  "code": "KO",
  "message": "No user's roles found!",
  "props": {}
}
```

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

Description

The service is responsible to get all available user's roles .