

# User Model Project

User Model Project but in more detail

More information: <http://www.lambdaschool.com>

Contact Info: [john@lambdaschool.com](mailto:john@lambdaschool.com)

Version: 1.0.0

BasePath: /

MIT

<https://github.com/LambdaSchool/java-usermodel/blob/master/LICENSE>

## Access

## Methods

[ [Jump to Models](#) ]

## Table of Contents

### [AuthorController](#)

- [GET /authors/authors](#)

### [BookController](#)

- [GET /books/books](#)

### [DataController](#)

- [POST /data/books/{bookid}/authors/{authorid}](#)
- [DELETE /data/books/{bookid}](#)
- [PUT /data/books/{bookid}](#)

### [Default](#)

- [POST /login](#)

### [LogoutEndpoint](#)

- [GET /logout](#)
- [GET /oauth/revoke-token](#)

### [OpenEndpoint](#)

- [POST /createnewuser](#)

### [RolesEndpoints](#)

- [POST /roles/role](#)
- [PUT /roles/role/{roleid}](#)
- [DELETE /roles/role/{id}](#)
- [GET /roles/role/{roleId}](#)
- [GET /roles/role/name/{roleName}](#)
- [GET /roles/roles](#)

### [UserEndpoints](#)

- [POST /users/user](#)
- [DELETE /users/user/{id}](#)
- [DELETE /users/user/{userid}/role/{roleid}](#)

- [GET /users/getuserinfo](#)
- [GET /users/getusername](#)
- [GET /users/user/email/count](#)
- [GET /users/user/{userId}](#)
- [GET /users/user/name/{userName}](#)
- [GET /users/user/name/like/{userName}](#)
- [GET /users/users](#)
- [POST /users/user/{userid}/role/{roleid}](#)
- [GET /users/allusers](#)
- [PUT /users/user/{id}](#)

### UseremailEndpoints

- [DELETE /useremails/useremail/{useremailid}](#)
- [GET /useremails/username/{userName}](#)
- [GET /useremails/useremail/{useremailid}](#)
- [GET /useremails/useremails](#)
- [PUT /useremails/useremail/{useremailid}/email/{emailaddress}](#)

## AuthorController

Up

GET /authors/authors

returns all Authors ([listAllAuthorsUsingGET](#))

### Return type

array[[Author](#)]

### Example data

Content-Type: application/json

```
[ {
  "fname" : "fname",
  "lname" : "lname",
  "wrote" : [ {
    "book" : {
      "isbn" : "123456789",
      "wrote" : [ null, null ],
      "section" : {
        "book" : [ null, null ],
        "name" : "Fiction",
        "sectionid" : 1
      },
      "copy" : 2011,
      "title" : "Android",
      "bookid" : 6
    }
  }, {
    "book" : {
      "isbn" : "123456789",
      "wrote" : [ null, null ],
      "section" : {
        "book" : [ null, null ],
        "name" : "Fiction",
        "sectionid" : 1
      },
      "copy" : 2011,
      "title" : "Android",
      "bookid" : 6
    }
  } ],
  "authorid" : 0
}
```

```

}, {
  "fname" : "fname",
  "lname" : "lname",
  "wrote" : [ {
    "book" : {
      "isbn" : "123456789",
      "wrote" : [ null, null ],
      "section" : {
        "book" : [ null, null ],
        "name" : "Fiction",
        "sectionid" : 1
      },
      "copy" : 2011,
      "title" : "Android",
      "bookid" : 6
    }
  } ],
  "authorid" : 0
} ]

```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200  
OK

## BookController

GET /books/books

Up

returns all Books (`listAllBooksUsingGET`)

### Return type

array[[Book](#)]

### Example data

Content-Type: application/json

```

[ {
  "isbn" : "123456789",
  "wrote" : [ null, null ],
  "section" : {
    "book" : [ null, null ],
    "name" : "Fiction",
    "sectionid" : 1
  },

```

```

    "copy" : 2011,
    "title" : "Android",
    "bookid" : 6
  }, {
    "isbn" : "123456789",
    "wrote" : [ null, null ],
    "section" : {
      "book" : [ null, null ],
      "name" : "Fiction",
      "sectionid" : 1
    },
    "copy" : 2011,
    "title" : "Android",
    "bookid" : 6
  } ]

```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200  
OK

## DataController

POST /data/books/{bookid}/authors/{authorid}

[Up](#)

Assign a book to an author (assignBookToAuthorUsingPOST)

### Path parameters

authorid (required)

*Path Parameter* — authorid format: int64

bookid (required)

*Path Parameter* — bookid format: int64

### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

### Return type

Object

### Example data

Content-Type: application/json

```
" {} "
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200  
Book Successfully Assigned [Object](#)  
404

Book or Author Not Found [ErrorDetail](#)

[Up](#)

DELETE /data/books/{bookid}

Delete an existing book (`deleteBookByIdUsingDELETE`)

#### Path parameters

**bookid (required)**

*Path Parameter* — bookid format: int64

#### Return type

Object

#### Example data

Content-Type:

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

#### Responses

200

Book Successfully Deleted [Object](#)

404

Book Not Found [ErrorDetail](#)

[Up](#)

PUT /data/books/{bookid}

Update an existing book (`updateBookUsingPUT`)

#### Path parameters

**bookid (required)**

*Path Parameter* — bookid format: int64

#### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

#### Request body

**updateBook [Book](#) (required)**

*Body Parameter* — updateBook

#### Return type

Object

#### Example data

Content-Type:

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

Book Successfully Updated [Object](#)

404

Book Not Found [ErrorDetail](#)

# Default

POST /login

[Up](#)

login (userLogin)

This is the login endpoint to request an authentication token

## Request body

userLogin [UserLogin](#) (required)  
*Body Parameter* — Login Parameter

## Return type

[TokenModel](#)

## Example data

Content-Type: application/json

```
{
  "access_token" : "access_token",
  "scope" : "scope",
  "token_type" : "token_type",
  "expires_in" : 0
}
```

## Responses

200

OK [TokenModel](#)

# LogoutEndpoint

GET /logout

[Up](#)

logout (logoutUsingGET)

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK

GET /oauth/revoke-token

[Up](#)

logout (logoutUsingGET1)

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

OK

## OpenEndpoint

[Up](#)

POST /createnewuser

addNewUser (addNewUserUsingPOST)

### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

### Request body

newminuser [UserMinimum](#) (required)

*Body Parameter* — newminuser

### Query parameters

getaccess (optional)

*Query Parameter* — getaccess default: true

### Return type

Object

### Example data

Content-Type: application/json

```
"{}"
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

OK [Object](#)

## RolesEndpoints

[Up](#)

POST /roles/role

addNewRole (addNewRoleUsingPOST)

### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

## Request body

newRole [Role](#) (required)  
*Body Parameter* — newRole

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

[Up](#)

PUT /roles/role/{roleid}

addNewRole (addNewRoleUsingPUT)

## Path parameters

roleid (required)  
*Path Parameter* — roleid format: int64

## Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

## Request body

newRole [Role](#) (required)  
*Body Parameter* — newRole

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

[Up](#)

DELETE /roles/role/{id}



deleteRoleById (deleteRoleByIdUsingDELETE)

### Path parameters

id (required)

*Path Parameter* — id format: int64

### Return type

Object

### Example data

Content-Type:

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

### Responses

200

OK [Object](#)

[Up](#)

GET /roles/role/{roleId}

getRoleById (getRoleByIdUsingGET)

### Path parameters

roleId (required)

*Path Parameter* — roleId format: int64

### Return type

Object

### Example data

Content-Type: application/json

```
" {} "
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

OK [Object](#)

[Up](#)

GET /roles/role/name/{roleName}

getRoleByName (getRoleByNameUsingGET)

### Path parameters

roleName (required)

*Path Parameter* — roleName

**Return type**

Object

**Example data**

Content-Type: application/json

```
"{}"
```

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

**Responses**

200

OK [Object](#)

GET /roles/roles

[Up](#)

listRoles (listRolesUsingGET)

**Return type**

Object

**Example data**

Content-Type: application/json

```
"{}"
```

**Produces**

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

**Responses**

200

OK [Object](#)

## UserEndpoints

POST /users/user

[Up](#)

adds a user given in the request body (addNewUserUsingPOST1)

**Consumes**

This API call consumes the following media types via the Content-Type request header:

- application/json

**Request body**

newuser [User](#) (required)  
*Body Parameter* — newuser

**Return type**

[User](#)

**Example data**

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

User Found [User](#)

404

User Not Found [ErrorDetail](#)

[Up](#)

DELETE /users/user/{id}

deleteUserById (deleteUserByIdUsingDELETE)

## Path parameters

id (required)

*Path Parameter* — id format: int64

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

[Up](#)

DELETE /users/user/{userid}/role/{roleid}

deleteUserRoleByIds (deleteUserRoleByIdsUsingDELETE)

## Path parameters

roleid (required)

*Path Parameter* — roleid format: int64

userid (required)

*Path Parameter* — userid format: int64

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

[Up](#)

GET /users/getuserinfo

getCurrentUserInfo (getCurrentUserInfoUsingGET)

## Query parameters

authenticated (optional)

*Query Parameter* —

authorities[0].authority (optional)

*Query Parameter* —

credentials (optional)

*Query Parameter* —

details (optional)

*Query Parameter* —

principal (optional)

*Query Parameter* —

## Return type

Object

## Example data

Content-Type: application/json

```
" {} "
```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

OK [Object](#)

[Up](#)

GET /users/getusername

getCurrentUserName (getCurrentUserNameUsingGET)

## Query parameters

authenticated (optional)

*Query Parameter* —

authorities[0].authority (optional)

*Query Parameter* —

credentials (optional)

*Query Parameter* —

details (optional)

*Query Parameter* —  
principal (optional)  
*Query Parameter* —

### Return type

Object

### Example data

Content-Type: application/json

```
"{}"
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

OK [Object](#)

GET /users/user/email/count

[Up](#)

getNumUserEmails (getNumUserEmailsUsingGET)

### Return type

Object

### Example data

Content-Type: application/json

```
"{}"
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

OK [Object](#)

GET /users/user/{userId}

[Up](#)

Retrieve a user based off user id (getUserByIdUsingGET)

### Path parameters

userId (required)

*Path Parameter* — User id format: int64

### Return type

[User](#)

### Example data

Content-Type: application/json

```
{
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ],
  "password" : "password",
  "userroles" : [ null, null ],
  "primaryemail" : "primaryemail",
  "userid" : 1,
  "passwordNotEncrypt" : "passwordNotEncrypt",
  "username" : "Some Name"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

User Found [User](#)

404

User Not Found [ErrorDetail](#)

GET /users/user/name/{userName}

[Up](#)

returns the user based off of user name ([getUserByNameUsingGET](#))

### Path parameters

**userName (required)**

*Path Parameter* – Username

### Return type

[User](#)

### Example data

Content-Type: application/json

```
{
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ],
  "password" : "password",
  "userroles" : [ null, null ],
  "primaryemail" : "primaryemail",
  "userid" : 1,
  "passwordNotEncrypt" : "passwordNotEncrypt",
  "username" : "Some Name"
}
```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

User Found [User](#)

404

User Not Found [ErrorDetail](#)

[Up](#)

GET /users/user/name/like/{userName}

returns all Users whose name contains the given substring with paging and sorting ([getUserLikeNameUsingGET](#))

## Path parameters

userName (required)

*Path Parameter* – Username

## Query parameters

page (optional)

*Query Parameter* – Results page you want to retrieve (1..N)

size (optional)

*Query Parameter* – Number of records per page.

sort (optional)

*Query Parameter* – Sorting criteria in the format: property(,asc|desc). Default sort order is ascending.

Multiple sort criteria are supported.

## Return type

array[[User](#)]

## Example data

Content-Type: application/json

```
[ {
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ],
  "password" : "password",
  "userroles" : [ null, null ],
  "primaryemail" : "primaryemail",
  "userid" : 1,
  "passwordNotEncrypt" : "passwordNotEncrypt",
  "username" : "Some Name"
}, {
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ],
  "password" : "password",
  "userroles" : [ null, null ],
  "primaryemail" : "primaryemail",
  "userid" : 1,
  "passwordNotEncrypt" : "passwordNotEncrypt",
```

```
"username" : "Some Name"
} ]
```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

OK

[Up](#)

## GET /users/users

returns all Users with paging and sorting (`listAllUsersUsingGET`)

## Query parameters

page (optional)

*Query Parameter* — Results page you want to retrieve (1..N)

size (optional)

*Query Parameter* — Number of records per page.

sort (optional)

*Query Parameter* — Sorting criteria in the format: property(,asc|desc). Default sort order is ascending.

Multiple sort criteria are supported.

## Return type

array[[User](#)]

## Example data

Content-Type: application/json

```
[ {
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ],
  "password" : "password",
  "userroles" : [ null, null ],
  "primaryemail" : "primaryemail",
  "userid" : 1,
  "passwordNotEncrypt" : "passwordNotEncrypt",
  "username" : "Some Name"
}, {
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ],
  "password" : "password",
  "userroles" : [ null, null ],
  "primaryemail" : "primaryemail",
  "userid" : 1,
  "passwordNotEncrypt" : "passwordNotEncrypt",
  "username" : "Some Name"
} ]
```



## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

OK

[Up](#)

POST /users/user/{userid}/role/{roleid}

postUserRoleByIds (postUserRoleByIdsUsingPOST)

## Path parameters

**roleid (required)**

*Path Parameter* — roleid format: int64

**userid (required)**

*Path Parameter* — userid format: int64

## Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

[Up](#)

GET /users/allusers

returns all Users without paging or sorting (reallyListAllUsersUsingGET)

## Return type

array[[User](#)]

## Example data

Content-Type: application/json

```
[ {
  "useremails" : [ {
    "useremailid" : 0,
    "useremail" : "useremail"
  }, {
    "useremailid" : 0,
    "useremail" : "useremail"
  } ]
}
```

```

    } ],
    "password" : "password",
    "userroles" : [ null, null ],
    "primaryemail" : "primaryemail",
    "userid" : 1,
    "passwordNotEncrypt" : "passwordNotEncrypt",
    "username" : "Some Name"
  }, {
    "useremails" : [ {
      "useremailid" : 0,
      "useremail" : "useremail"
    }, {
      "useremailid" : 0,
      "useremail" : "useremail"
    } ],
    "password" : "password",
    "userroles" : [ null, null ],
    "primaryemail" : "primaryemail",
    "userid" : 1,
    "passwordNotEncrypt" : "passwordNotEncrypt",
    "username" : "Some Name"
  } ]

```

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

### Responses

200

OK

PUT /users/user/{id}

[Up](#)

updateUser (updateUserUsingPUT)

### Path parameters

id (required)

*Path Parameter* — id format: int64

### Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

### Request body

updateUser [User](#) (required)

*Body Parameter* — updateUser

### Return type

Object

### Example data

Content-Type:

### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

# UseremailEndpoints

[Up](#)

DELETE /useremails/useremail/{useremailid}

deleteUserEmailById (deleteUserEmailByIdUsingDELETE)

## Path parameters

useremailid (required)

*Path Parameter* — useremailid format: int64

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

[Up](#)

GET /useremails/username/{userName}

findUseremailByUsername (findUseremailByUsernameUsingGET)

## Path parameters

userName (required)

*Path Parameter* — userName

## Return type

Object

## Example data

Content-Type: application/json

```
" {} "
```

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

## Responses

200

OK [Object](#)

GET /useremails/useremail/{useremailId}

getUserEmailById (getUserEmailByIdUsingGET)

#### Path parameters

useremailId (required)

*Path Parameter* — useremailId format: int64

#### Return type

Object

#### Example data

Content-Type: application/json

```
"{}"
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK [Object](#)

GET /useremails/useremails

listAllUseremails (listAllUseremailsUsingGET)

#### Return type

Object

#### Example data

Content-Type: application/json

```
"{}"
```

#### Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

#### Responses

200

OK [Object](#)

PUT /useremails/useremail/{useremailid}/email/{emailaddress}

updateUserEmail (updateUserEmailUsingPUT)

#### Path parameters

emailaddress (required)

*Path Parameter* — emailaddress

useremailid (required)

*Path Parameter* — useremailid format: int64

## Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

## Return type

Object

## Example data

Content-Type:

## Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- \*/\*

## Responses

200

OK [Object](#)

## Models

[ [Jump to Methods](#) ]

### Table of Contents

1. [Author - Author](#)
2. [Book - Book](#)
3. [ErrorDetail - ErrorDetail](#)
4. [Role - Role](#)
5. [Section - Section](#)
6. [TokenModel - TokenModel](#)
7. [User - User](#)
8. [UserMinimum - UserMinimum](#)
9. [Useremail - Useremail](#)
10. [Userroles - Userroles](#)
11. [ValidationError - ValidationError](#)
12. [Wrote - Wrote](#)

### Author - Author

[Up](#)

authorid (optional)

[Long](#) format: int64

fname (optional)

[String](#)

lname (optional)

[String](#)

wrote (optional)

[array\[Wrote\]](#)

### Book - Book

[Up](#)

Book Entity

bookid (optional)

[Long](#) format: int64

copy (optional)  
[\*Integer\*](#) The year the book was published (copyright date) format: int32

isbn (optional)  
[\*String\*](#) The ISBN number of the book

section (optional)  
[\*Section\*](#)

title  
[\*String\*](#) The title of the book

wrote (optional)  
[\*array\[Wrote\]\*](#) List of book authors

## ErrorDetail - ErrorDetail

[Up](#)

Exceptions encountered when working with the REST API are reported to the client using this format

detail (optional)  
[\*String\*](#)

developerMessage (optional)  
[\*String\*](#)

errors (optional)  
[\*map\[String, array\[ValidationError\]\]\*](#)

status (optional)  
[\*Integer\*](#) format: int32

timestamp (optional)  
[\*Long\*](#) format: int64

title (optional)  
[\*String\*](#)

## Role - Role

[Up](#)

Determines what access the user has

name  
[\*String\*](#) The Name of the Role. This name is hard coded throughout the application so change it with reservations!

roleid  
[\*Long\*](#) primary key for Role format: int64

userroles (optional)  
[\*array\[Userroles\]\*](#) A List of the Users who are assigned this role

## Section - Section

[Up](#)

Section Entity

book (optional)  
[\*array\[Book\]\*](#)

name  
[\*String\*](#) Name of section

sectionid  
[\*Long\*](#) Primary key for section format: int64

## TokenModel - TokenModel

[Up](#)

access\_token (optional)

[\*String\*](#)

expires\_in (optional)

[\*Integer\*](#) format: int32

scope (optional)

[\*String\*](#)

token\_type (optional)

[\*String\*](#)

## **User - User**

[Up](#)

Yes, this is an actual user

password (optional)

[\*String\*](#)

passwordNotEncrypt (optional)

[\*String\*](#)

primaryemail (optional)

[\*String\*](#)

useremails (optional)

[\*array\[Useremail\]\*](#)

userid

[\*Long\*](#) primary key for User format: int64

username

[\*String\*](#) Actual user name for sign on

userroles (optional)

[\*array\[Userroles\]\*](#)

## **UserMinimum - UserMinimum**

[Up](#)

password (optional)

[\*String\*](#)

primaryemail (optional)

[\*String\*](#)

username (optional)

[\*String\*](#)

## **Useremail - Useremail**

[Up](#)

A list of secondary emails for the user

user (optional)

[\*User\*](#)

useremail (optional)

[\*String\*](#)

useremailid (optional)

[\*Long\*](#) format: int64

## **Userroles - Userroles**

[Up](#)

Which user has what roles

role (optional)

[\*Role\*](#)

user (optional)

[User](#)

**ValidationError - ValidationError**

[Up](#)

code (optional)

[String](#)

message (optional)

[String](#)

**Wrote - Wrote**

[Up](#)

author (optional)

[Author](#)

book (optional)

[Book](#)