# User REST service API description

## Get Users collection

**URL**: '/api/users'

**Method**: GET

**Success response example:**

Format JSON

[{  
 id: String,  
 firstName: String,  
 lastName: String,  
 company: String,  
 position: String,  
 email: String,  
 phoneNumber: String  
}, …]

Note: response will contain an empty array in case if there is no any record in database.

**Error responses**:

If any server error it’ll return HTTP 500

## Get User

**URL**: '/api/users/{ID}'

**Method**: GET

**Parameters description:**

{ID} is part of URL. Service matches model by this ID.

**Success response example:**

Format JSON

{  
 id: String,  
 firstName: String,  
 lastName: String,  
 company: String,  
 position: String,  
 email: String,  
 phoneNumber: String  
}

**Error responses**:

If any server error it’ll return HTTP 500

## Create User

**URL**: '/api/users'

**Method**: POST

**Request payload:**

Format JSON. All fields are optional.

{  
 "firstName": "Alex",  
 "lastName": "Hall",  
 "company": "OBS",  
 "position": "Postman",  
 "email": "ahall@obs.com",  
 "phoneNumber": "+46577894563"  
}

**Success response example:**

Format JSON

After service creates model in DB, model get new field **id**. After successful creation service sends updated model in response.

{  
 id: String,  
 firstName: String,  
 lastName: String,  
 company: String,  
 position: String,  
 email: String,  
 phoneNumber: String  
}

**Error responses**:

If any server error it’ll return HTTP 500

## Update User

**URL**: '/api/users/{ID}'

**Method**: PUT

**Parameters description:**

{ID} is part of URL. Service matches model by this ID and then updates it with data from request payload.

**Request payload:**

Format JSON.

{  
 "id": "9879s8d7f67df67",

"firstName": "Alex",  
 "lastName": "Hall",  
 "company": "OBS",  
 "position": "Postman",  
 "email": "ahall@obs.com",  
 "phoneNumber": "+46577894563"  
}

**Success response example:**

Format JSON

After successful creation service sends updated model in response.

{  
 id: String,  
 firstName: String,  
 lastName: String,  
 company: String,  
 position: String,  
 email: String,  
 phoneNumber: String  
}

**Error responses**:

If any server error it’ll return HTTP 500

## Delete User

**URL**: '/api/users/{ID}'

**Method**: DELETE

**Parameters description:**

{ID} is part of URL. Service matches model by this ID and then remove it from DB.

**Success response example:**

HTTP 200

**Error responses**:

If any server error it’ll return HTTP 500