



## INTRODUCTION

This is the version 1 of the Watch Tracker API. Below you will find a current list of the available methods on our API.

## INDEX

USERS	2
MOVIES	8
PROVIDERS	12
USER SUBSCRIPTIONS	14

## INFORMATION

In the response information if the boolean variable is strikethrough it means that the return doesn't return a boolean with that name, just a boolean, but in the API is written like this to clarify some method responses.

## USERS

### GET

URL: `api/Users/GetAllUsers`

Get all users basic information.

RESPONSE INFORMATION	
Key	Value
UserId	String
Email	String
Name	String
Surname	String
Birthday	DateTime
RegisterDate	DateTime
Active	Boolean

Returns *UserNoImgDTO* object.

URL: api/Users?userId={userId}

Check if a specific username exists in the database.

RESPONSE INFORMATION	
Key	Value
<del>Exists</del>	Boolean

Returns *true* if the username exists, *false* if not.

URL: api/Users?email={email}

Check if a specific email exists in the database.

RESPONSE INFORMATION	
Key	Value
<del>Exists</del>	Boolean

Returns *true* if the email exists, *false* if not.

URL: api/Users?userId={userId}&password={password}

Check if the username and password are correct.

RESPONSE INFORMATION	
Key	Value
<del>Correct</del>	Boolean

Returns *true* if are correct, *false* if not.

URL: api/Users/UserStatusCount?userId={userId}

Get how many movies the user is watching, is planning to watch and has watched.

RESPONSE INFORMATION	
Key	Value
<del>Watching</del>	Integer
<del>Watched</del>	Integer
<del>Plan to Watch</del>	Integer

Returns *int[]*.

URL: `api/Users/UserData?userId={userId}`

Get the information about a specified user.

RESPONSE INFORMATION	
Key	Value
UserId	String
Email	String
Name	String
Surname	String
Birthday	DateTime
Image	Byte[]
RegisterDate	DateTime
Active	Boolean

Returns *UserDTO* object.

URL: `api/Users/Registers`

Get all the registration dates and how many times is repeated each one.

RESPONSE INFORMATION	
Key	Value
<del>RegisterDate</del>	String
<del>Count</del>	String

Returns *List<String[]>* in this format [ ["2021-02-02","2"],["2021-02-06","1"] ].

# POST

URL: api/Users

Create a new user in the database.

BODY PARAMETER (User object)		
Key	Value	Required
UserId	String	Yes
Password	String	Yes
Email	String	Yes
Name	String	Yes
Surname	String	Yes
Birthday	DateTime	Yes
Image	Byte[]	No

Returns true if the user was successfully created.

# PUT

URL: api/Users?userId={userId}&oldPassword={oldPassword}&newPassword={newPassword}

Change the user password.

RESPONSE INFORMATION	
Key	Value
Changed	Boolean

Returns true if password was successfully changed.

URL: api/Users/ChangeData?userId={userId}

Change the name and surname of a user.

BODY PARAMETERS	
Key	Value
User	User

Returns true if the data was successfully changed.

URL: api/Users/ChangeUserActive?userId={userId}

Change the active status of a user, if the user is active deactivates the user status, and if it is deactivated, activates the user status.

RESPONSE INFORMATION	
Key	Value
Changed	Boolean

Returns true if the change was successful.

URL: api/Users/GeneratePassword?userId={userId}

Generate a new password for the user and send a mail to him with the new generated password.

RESPONSE INFORMATION	
Key	Value
Changed	Boolean

Returns true if the password was successfully generated.

URL: api/Users?userId={userId}

Generate a new password for the user and send a mail to him with the new generated password.

BODY PARAMETERS	
Key	Value
User	User

\* You have to have a value in the User.Image attribute to change the image.

Returns true if the change was successful.

# DELETE

URL: `api/Users?userId={userId}`

Delete a user.

RESPONSE INFORMATION	
Key	Value
<del>Deleted</del>	Boolean

Returns true if the user was successfully deleted.

## MOVIES

### GET

URL: `api/Movies?movieId={movieId}&userId={userId}`

Get all the details about a specific movie and the user selected values for the movie.

RESPONSE INFORMATION	
Key	Value
MovieId	Integer
Poster	String
Title	String
VoteAverage	Double
ReleaseDate	String
Overview	String
Popularity	Double
VoteCount	Integer
Runtime	Integer
Providers	List<String>
UserSatus	String
UserData	DateTime
UserVote	Integer
UserId	String

Returns Movie object.



URL: `api/Movies?movieName={movieName}`

Get a list of movies that match with the name provided.

RESPONSE INFORMATION	
Key	Value
Movied	Integer
Poster	String
Title	String
VoteAverage	Double
Overview	String

Returns List<MovieDTO> object.

URL: `api/Movies/UserMovies?userId={userId}`

Get a list of movies that the user has in his list.

RESPONSE INFORMATION	
Key	Value
Movied	Integer
Poster	String
Title	String
VoteAverage	Double
Overview	String

Returns List<MovieDTO> object.

URL: `api/Movies/Recent`

Get a list of movies in theaters.

RESPONSE INFORMATION	
Key	Value
Movied	Integer
Poster	String
Title	String

Returns List<MoviePosterDTO> object.

URL: api/Movies/Upcoming

Get a list of upcoming movies in theaters.

RESPONSE INFORMATION	
Key	Value
Movied	Integer
Poster	String
Title	String

Returns List<MoviePosterDTO> object.

URL: api/Movies/TopRated

Get the top rated movies on TMDb.

RESPONSE INFORMATION	
Key	Value
Movied	Integer
Poster	String
Title	String

Returns List<MoviePosterDTO> object.

URL: api/Movies/MostListed

Get the top five movies added by users to their lists.

RESPONSE INFORMATION	
Key	Value
Title	String
Count	String

Returns List<String[]> in this format [ ["Iron-Man","2"],["The Lord of Rings","1"] ].

# POST

URL: api/Movies

Create a new movie in the user's list in the database.

BODY PARAMETER (UserMovies object)		
Key	Value	Required
MovieId	Integer	Yes
UserSatus	String	Yes
UserData	DateTime	Yes
UserVote	Integer	Yes
UserId	String	Yes

No response.

# PUT

URL: api/Movies?movieId={movieId}&userId={userId}&userSatus={userSatus}&userDate={userData}&userVote={userVote}

Change the user values for a movie added to his list.

No response.

# DELETE

URL: api/Movies?movieId={movieId}&userId={userId}

Delete a movie from the user's list.

No response.

## PROVIDERS

### GET

URL: `api/Providers`

Get all the providers in the database.

RESPONSE INFORMATION	
Key	Value
ProviderId	Integer
ProviderName	String
ProviderLogo	Byte[]

Returns *Provider* object.

# POST

URL: api/Providers

Create a new provider.

BODY PARAMETER (Provider object)		
Key	Value	Required
ProviderId	Integer	Yes
ProviderName	String	Yes
ProviderLogo	Byte[]	No

Returns true if the Provider was successfully created.

# PUT

URL: api/Providers?providerId={providerId}

Change the data of a specific provider.

BODY PARAMETER (Provider object)		
Key	Value	Required
ProviderId	Integer	Yes
ProviderName	String	Yes
ProviderLogo	Byte[]	No

Returns true if the Provider data was successfully changed.

# DELETE

URL: api/Providers?provierId={provierId}

Delete a provider.

Returns true if the Provider was successfully deleted.

## USER SUBSCRIPTIONS

### GET

URL: `api/UsersSubscriptions?userId={userId}`

Get all subscriptions that the user has.

RESPONSE INFORMATION	
Key	Value
UserSubscriptionsId	Integer
ProviderLogo	Byte[]
ProviderName	String
PaymentDate	DateTime
BillingPeriod	String
Price	Double
UserId	String
ProviderId	Integer

Returns *UserSubscriptionsDTO* object.

# POST

URL: api/UsersSubscriptions

Create a new subscription for a user.

BODY PARAMETER (UserSubscriptions object)		
Key	Value	Required
UserSubscriptionsId	Integer	Yes
ProviderName	String	Yes
PaymentDate	DateTime	Yes
BillingPeriod	String	Yes
Price	Double	Yes
UserId	String	Yes
ProviderId	Integer	Yes

Returns *true* if the user subscription was successfully created.

# PUT

URL: api/UsersSubscriptions/{id}?paymentDate={paymentDate}&billingPeriod={billingPeriod}&price={price}

Change the data of a specific user subscription.

Returns *true* if the data was successfully changed.

# DELETE

URL: api/UsersSubscriptions?userId={userId}&providerId={providerId}

Delete a user subscription.

Returns *true* if the user subscription was successfully deleted.