Web service specification v15

Contents

[Current User Points 4](#_Toc468798132)

[Description 4](#_Toc468798133)

[Parameters 4](#_Toc468798134)

[Request example 4](#_Toc468798135)

[Structure 4](#_Toc468798136)

[Videos filtered by tags 5](#_Toc468798137)

[Description 5](#_Toc468798138)

[Parameters 5](#_Toc468798139)

[Request example 5](#_Toc468798140)

[Structure 5](#_Toc468798141)

[Video comments 7](#_Toc468798142)

[Description 7](#_Toc468798143)

[Parameters 7](#_Toc468798144)

[Request example 7](#_Toc468798145)

[Structure 7](#_Toc468798146)

[Add point and/or view 8](#_Toc468798147)

[Description 8](#_Toc468798148)

[Parameters 8](#_Toc468798149)

[Request example 8](#_Toc468798150)

[Structure 8](#_Toc468798151)

[Add comment to video 9](#_Toc468798152)

[Description 9](#_Toc468798153)

[Parameters 9](#_Toc468798154)

[Request example 9](#_Toc468798155)

[Structure 9](#_Toc468798156)

[Manage video like 10](#_Toc468798157)

[Description 10](#_Toc468798158)

[Parameters 10](#_Toc468798159)

[Request example 10](#_Toc468798160)

[Structure 10](#_Toc468798161)

[Update user information 11](#_Toc468798162)

[Description 11](#_Toc468798163)

[Parameters 11](#_Toc468798164)

[Request example 11](#_Toc468798165)

[Structure 11](#_Toc468798166)

[Log in user 12](#_Toc468798167)

[Description 12](#_Toc468798168)

[Parameters 12](#_Toc468798169)

[Request example 12](#_Toc468798170)

[Structure 12](#_Toc468798171)

[Login succeeded 12](#_Toc468798172)

[Login falied – login information incorrect, deleted user, blocked user, deactivated user (check each one) 12](#_Toc468798173)

[All geographic units 14](#_Toc468798174)

[Description 14](#_Toc468798175)

[Parameters 14](#_Toc468798176)

[Request example 14](#_Toc468798177)

[Structure 14](#_Toc468798178)

[Register user to app 15](#_Toc468798179)

[Description 15](#_Toc468798180)

[Parameters 15](#_Toc468798181)

[Request example 15](#_Toc468798182)

[Structure 15](#_Toc468798183)

[Change password 16](#_Toc468798184)

[Description 16](#_Toc468798185)

[Parameters 16](#_Toc468798186)

[Request example 16](#_Toc468798187)

[Structure 16](#_Toc468798188)

[Deactivate account 17](#_Toc468798189)

[Description 17](#_Toc468798190)

[Parameters 17](#_Toc468798191)

[Request example 17](#_Toc468798192)

[Structure 17](#_Toc468798193)

[Activate account (not meant to be web service!!!! Change to web page for production... ) 18](#_Toc468798194)

[Description 18](#_Toc468798195)

[Parameters 18](#_Toc468798196)

[Request example 18](#_Toc468798197)

[Structure 18](#_Toc468798198)

[Change telecommunication operator 19](#_Toc468798199)

[Description 19](#_Toc468798200)

[Parameters 19](#_Toc468798201)

[Request example 19](#_Toc468798202)

[Structure 19](#_Toc468798203)

[All telecomm operators 20](#_Toc468798204)

[Description 20](#_Toc468798205)

[Parameters 20](#_Toc468798206)

[Request example 20](#_Toc468798207)

[Structure 20](#_Toc468798208)

[Add telecomm user connection 21](#_Toc468798209)

[Description 21](#_Toc468798210)

[Parameters 21](#_Toc468798211)

[Request example 21](#_Toc468798212)

[Structure 21](#_Toc468798213)

[Check phone number 22](#_Toc468798214)

[Description 22](#_Toc468798215)

[Parameters 22](#_Toc468798216)

[Request example 22](#_Toc468798217)

[Structure 22](#_Toc468798218)

[Get points stats per month 23](#_Toc468798219)

[Description 23](#_Toc468798220)

[Parameters 23](#_Toc468798221)

[Request example 23](#_Toc468798222)

[Structure 23](#_Toc468798223)

# Current User Points

## Description

Web service returns current user points for telecommunication operator to use that information for discount and benefits for user in their webshop.

## Parameters

user\_id – active user's id currently logged in

telecomm\_operator – active telecomm operator for logged-in user, its code is used instead of id

## Request example

GET localhost/webservice/current\_user\_points?user\_id=1&telecomm\_operator=111

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"user\_id\_remote":"<user\_id\_remote>",

"points": ##

}

}

# Videos filtered by tags

## Description

This webservice returns videos and likes for user to choose which video he wants to watch. Those videos are filtered by tags. Inside json, It is determined if video was watched by user with parameter seen. Seen = 1 means video was seen and seen = 0 means it was not.

user\_like says which like current user used, if 1 it means user liked video, if 0 it means there is no like under that video and -1 means user disliked it.

## Parameters

id – logged-in user for whom are videos filtered

## Request example

GET localhost/webservice/videos?id=3

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

[

{

"id":###,

"title":"<title>",

"link":"<link>",

"thumbnail\_url":"<thumbnail\_url>",

"likes":###,

"dislikes":###,

"points":###,

"seen":###,

"user\_like":###,

"created\_at":"<created\_at>"

},

{

"id":###,

"title":"<title>",

"link":"<link>",

"thumbnail\_url":"<thumbnail\_url>",

"likes":###,

"dislikes":###,

"points":###,

"seen":###,

"user\_like":###,

"created\_at":"<created\_at>"

}

]

}

}

# Video comments

## Description

Web service which returns all comments for video that is watched at least one time.

## Parameters

video – watched video's id

## Request example

GET localhost/webservice/video\_comments?video=1

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

[

{

"id":###,

"author":{

"id":"<id>",

"name":"<name>",

"surname":"<surname>"

},

"content":"<content>",

"created\_at":"date and time of creation"

},

{

"id":###,

"author":{

"id":"<id>",

"name":"<name>",

"surname":"<surname>"

},

"content":"<content>",

"created\_at":"date and time of creation"

}

]

}

}

# Add point and/or view

## Description

Web service adds point and/or view. If user hasn't seen the video yet, points are added to user and video view too. If video has one view, extra view is added to user's video views.

## Parameters

video – video's id that is seen

user – user's id that is logged-in currenty

## Request example

GET localhost/webservice/view\_video?video=1&user=3

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Add comment to video

## Description

Web service adds new comment to video that is currently open.

## Parameters

video – video's id that is currently open

author – user's id that wrote comment

content – comment's content that shall be added

## Request example

GET localhost/webservice/comment\_video?video=1&author=1&content=This is my comment

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Manage video like

## Description

Web service that updates user's like status.

## Parameters

video – video's id that is seen

user – user's id that is logged-in currenty

like - current like value

## Request example

GET localhost/webservice/videolike?video=1&user=1&like=-1

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Update user information

## Description

Web service to update user information

If password is not changed at all, do not send password parameter. If you do send it with old password, it will result with error that password cannot be changed into old one. Be aware of that.

## Parameters

id – user's id that is logged-in currenty

name - user's name

surname – user's surname

birthday – user's birthday

gender – user's gender

email – user's email

username – user's username

geo\_unit – geographic unit's id

mobile\_number – user's mobile phone number

password – user's NEW password (DO NOT SEND old one) Send it only if it's changed.

## Request example

POST localhost/webservice/updateuser with all parameters written above

NOTICE: use GET method to test web service , POST IS REQUIRED

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Log in user

## Description

Web service that checks login info for user and returns its information.

## Parameters

username – username for user to be logged in

password - hashed password for user to be logged in

## Request example

POST localhost/webservice/login?username=username&password=password

## Structure

### Login succeeded

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message":"<message>",

"user":{

"id":##,

"name": "<name>",

"surname": "<surname>",

"birthday": "<birthday>",

"gender": "<gender>",

"email": "<email>",

"username": "<username>",

"password": "<password>",

"password\_changed\_at": "<password\_changed\_at>",

"mobile\_number": "<mobile\_number>",

"geographic\_unit": {

"id": ###,

"name": "<name>"

}

}

}

}

### Login falied – login information incorrect, deleted user, blocked user, deactivated user (check each one)

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# All geographic units

## Description

Web service returns all geographic units.

## Parameters

None

## Request example

GET localhost/webservice/geographic\_units

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":[

{

"id":###,

"name": "<name>"

},

{

"id":###,

"name": "<name>"

}

]

}

# Register user to app

## Description

Web service for registering users to application.

If service is not successful it returns null as id, otherwise it returns integer id.

## Parameters

name - user's name

surname – user's surname

birthday – user's birthday

gender – user's gender

email – user's email

username – user's username

password – user's password

geo\_unit – geographic unit's id

mobile\_number – user's mobile phone number (format: 038598…)

## Request example

GET localhost/webservice/registration with all parameters specified

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>",

"id": „null“ or ###

}

}

# Change password (to DELETE - not used)

## Description

Web service for changing user password. The web service returns false if password is equal to old one.

## Parameters

password – new user's password

id – user who wants to change password

## Request example

GET localhost/webservice/change\_password?id=1&password=nekalozinka

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Deactivate account

## Description

Web service for deactivating user account. If user is blocked, deactivated or deleted, web service returns false .

## Parameters

id – user id

## Request example

GET localhost/webservice/deactivate?id=1

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Activate account (not meant to be web service!!!! Change to web page for production... )

## Description

Web service for activating user account. If user is blocked, activated or deleted, web service returns false .

## Parameters

id – user id

## Request example

GET localhost/webservice/activate\_user?id=1

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Change telecommunication operator

## Description

Web service for changing active telecommunication operator. If new one is the same as old one it returns error.

## Parameters

id – user id

telecomm – telecommunication operator code

## Request example

GET localhost/webservice/change\_telecomm?id=1&telecomm=2322

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# All telecomm operators

## Description

Web service returns all telecomm operators.

## Parameters

None

## Request example

GET localhost/webservice/telecomm\_operators

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":[

{

"id":###,

"name": "<name>"

},

{

"id":###,

"name": "<name>"

}

]

}

# Add telecomm user connection

## Description

Web service adds new telecomm user connection.

## Parameters

user – user id

telecomm – telecomm operator code

## Request example

GET localhost/webservice/add\_telecomm\_user?user=1&telecomm=222

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Check phone number

## Description

Web service checks if phone number exists and returns true if it does.

Number format in database: +385987654321

Number format in webservice parameter: 0385987654321

+385 is number for country croatia.

## Parameters

phone\_number – phone number in format : 038598...

## Request example

GET localhost/webservice/check\_phone\_number?phone\_number=0385987654321

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"success":##,

"message": "<message>"

}

}

# Get points stats

## Description

Web service that returns spent and earned points.

Web service will take active teleoperator and find all information needed for this web service.

## Parameters

user – id of currently logged user

## Request example

GET localhost/webservice/points\_stats?user=1

## Structure

{

"status":100,

"requested\_at":"<date and time>",

"data":{

"earned":##,

"spent":####,

}

}