Research of

Spika for Business

<https://github.com/neotim/Spika/>

Neo Tim

Spika for business is full customisable chat platform for small to big team. This document is made for developers to help them to customise Spika.

If you have any request about this document please send email to **info@clover.studio**.

Document tasks

Here are contents planned to add to this document.

1. Usage of API v3
2. Performance Test Result
3. Here is perfonrmance test result we did in June, 2017.

| **App Sever Spec** | **MongoDB Serve Spec** | **Redis Server Spec** | **Concurrent Users** | **Messages sent per sec** |
| --- | --- | --- | --- | --- |
| Xeon 2.4Ghz 2cores, 7GB Mem, 2 servers | Xeon 2.4Ghz 2cores, 7GB Mem, 2 servers | Xeon 2.4Ghz 1core, 3.5GB Mem | 2000 | 20 |
| Xeon 2.4Ghz 2cores, 7GB Mem, 4 servers | Xeon 2.4Ghz 4cores, 8GB Mem, 2 servers | Xeon 2.4Ghz 1core, 3.5GB Mem | 3800 | 40 |



# Setting Up Server

Here is step by step instruction to setup new backend and frontend. Before doing this all our clients have to buy license and we provice access permission to our github private repository.

### **Minimum server requirements.**

This tutorial is done using t2.micro instance in Amazon Web Service using Ubunti 16.04. t2.micro instance's spec is following.

|  |  |
| --- | --- |
| CPU | 1core 1GHz |
| Memory | 1GB |
| Storage | 8GB |

**Please be careful it is minimum server requirements, you will need more powerful server for production or even for development.**

### **Setting up backend.**

#### **Install softwares**

$ curl -sL https://deb.nodesource.com/setup\_6.x | sudo -E bash -

$ sudo apt-get install -y nodejs

$ sudo apt-get install -y git imagemagick build-essential libfuse-dev libcurl4-openssl-dev libxml2-dev mime-support automake libtool mongodb redis-server turnserver python

#### **Clone from repository**

**You will have your own repository. So please change this url part.**

$ git clone https://github.com/cloverstudio/SFB\_Server.git

#### **Setup Backend**

$ cd SFB\_Server

$ npm install

#### **Start servers**

$ sudo /etc/init.d/mongodb start

$ sudo /etc/init.d/redis-server start

#### **Edit init.js**

$ cp src/server/lib/init-sample.js src/server/lib/init.js

$ vim src/server/lib/init.js

Probably you have to change following configuration to start server.

Config.port = 8080;

**Please be sure the port is opened so you can access to the server from browser.**

#### **Create directory**

$ mkdir public/uploads

#### **Start Spika Backend**

$ sudo node src/server/main

If you see this Spika is correctly started.

Connecting mongoDB mongodb://localhost/spikaenterprise

Server listening on port 8080!

##### **In some cases following error could occure.**

Error: Cannot find module '../build/Debug/iconv.node'

at Function.Module.\_resolveFilename (module.js:469:15)

If you encount this error please just do this.

$ npm i iconv

### **What's next ?**

Please open service owner admin console like this

http://54.215.140.61:8080/owner

Put Username and Password which is in init.js

# Setting Turn/Stun server

Spika uses standard turn-stun protocol for WebRTC communication.

We recommend to user this server to host Turn/Stun by your self. https://github.com/coturn/coturn

### **Install library**

$ wget https://github.com/downloads/libevent/libevent/libevent-2.0.21-stable.tar.gz

$ tar xvfz libevent-2.0.21-stable.tar.gz

$ cd libevent-2.0.21-stable

$ ./configure

$ make

$ sudo make install

### **Coturn installation**

Find latest code from here. https://github.com/coturn/coturn/wiki/Downloads

Build

$ tar xvfz turnserver-<...>.tar.gz

$ ./configure

$ make

$ sudo make install

Start server

$ turnserver -L <public\_ip\_address> -o -a -f -r <realm-name>

You have detaild instructions here. https://github.com/coturn/coturn/wiki/CoturnConfig

# Configuration

Here is explanation of init.js.

**Config.host**: Keep this "localhost"

**Config.port**: Runnning port.

**Config.urlPrefix**: Change this if you want change baseURL of API.

**Config.dbCollectionPrefix**: Im most case keep this empty string.

**Config.databaseUrl**: Connection string to mongoDB

**Config.supportUserId**: If you set userId here, the user appears to all users member list.

**Config.forceLogoutAllDevice**: Set true, if you want users to auto logout from all device when user logout from one device.

**Config.redis**: Connection info for Redis

**Config.AESPassword**: Password for encryption text message.

**Config.publicPath**: Where you put index.html and all sets for frontend part.

**Config.uploadPath**: Where saved files which users upload.

**Config.socketNameSpace**: Keep this as it is.

**Config.defaultAvatar**: Default avatar for users.

**Config.defaultAvatarGroup**: Default avatar for groups.

**Config.hashSalt**: Hash salt for generating hash from password.

**Config.username**: Service owner username.

**Config.password**: Service owner password.

**Config.signinBackDoorSecret**: This is used only for debugging purpose.

**Config.apnsCertificates**: Certificates for push notification.

**Config.gcmAPIKey**: This is depricated.

**Config.fcmServerKey**: FCM settings.

**Config.webRTCConfig**: WebRTC settings.

**Config.email**: **Config.smtp**: It is used for send email when user signup.

**Config.protocol**: If you use builtin ssl change here to "https://"

**Config.twilio**: It is used when you want allow users signup by telephone number.

**Config.useVoipPush**: Keep here true when you want use VoIP push for APN.

**Config.useCluster**: This is experimented parameter. We will announce when it wors correctly.

**Config.robotUserId**: Create user who you want use for webhooks.

# Spika Enterprise

## **Spika Enterprise API's reference**

**0.1.0**

# WebAPI

# WebAPI

**0.0.0**

Returns list of messages. Direction should be string 'old' or 'new'

/api/v2/message/list/:roomId/:lastMessageId/:direction

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-MessageList-0_0_0-0)

{

"code": 1,

"time": 1499868072495,

"data": {

"messages": [{

"\_\_v": 0,

"\_id": "594abd2a17a3077a5c0381dc",

"created": 1498070314307,

"localID": "culkpvtyEs1G6OUqQ2RkkjZXKBjZCFC7",

"message": "0301dc83cfffa5a368ade6872e397f7595c7cee27261a8cd0282c16caf0a41b3e86555ce92acf0c7ef00796fb6ff0540cf4df36a7e270cc840e84fbc3f293276bc243894f6e31a0ad81af10d9339fd1615aada49b4476f99c967a472437ffaa31636dafa8643eee8853b8ed7305111756625",

"remoteIpAddress": "109.60.110.82",

"roomID": "3-58fdd63c5b9736991af8ca13",

"type": 1,

"user": {

"\_id": "56e0088062a63ebf55eef39a",

"description": "Samo jakooooo!!!",

"name": "Jurica Blazevic",

"organizationId": "56e005b1695213295419f5df",

"userid": "jurica.blazevic",

"avatar": {

"thumbnail": {

"originalName": "417020\_10150655766113706\_100243839\_n.jpg",

"size": 28617,

"mimeType": "image/png",

"nameOnServer": "Nk51aWGNBmpP5im8x3ecJwAKizI7ANJe"

},

"picture": {

"originalName": "417020\_10150655766113706\_100243839\_n.jpg",

"size": 28617,

"mimeType": "image/png",

"nameOnServer": "1QKB80SOm3waYooa3enPxPBIbiENj5Y6"

}

}

},

"userID": "56e0088062a63ebf55eef39a",

"seenBy": [{

"version": 2,

"at": 1498623027378,

"user": "57f315081d9cabd56e905080"

}..

],

"isFavorite": 0

}, ...

}]

}

}

# WebAPI - Add users to room

**0.0.0**

Update profile of conversation

/api/v2/room/users/add

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-Token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| roomId | string | roomId |
| users | Array | Array of user ids |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Add_users_to_room-0_0_0-0)

{

"code": 1,

"time": 1457082886197,

"data": {

"room": {

"\_id": "56d94c88bf06a1f30ad6091d",

"owner" : "56c32acd331dd81f8134f200"

"ownerModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457082869691,

"token": "\*\*\*\*\*",

"departments": [],

"groups": ["56cf0a60ed51d2905e28a848"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"name": "ああああああ",

"created": 1457081480907,

"\_\_v": 0,

"description": "いいいいいddd",

"modified": 1457082886193,

"avatar": {

"thumbnail": {

"originalName": "2014-06-03 17.23.39.jpg",

"size": 1504586,

"mimeType": "image/png",

"nameOnServer": "ut4G1A3Jq9LbfeDxXUh8jibgDB4wPGV1"

},

"picture": {

"originalName": "2014-06-03 17.23.39.jpg",

"size": 1504586,

"mimeType": "image/png",

"nameOnServer": "egStPNb3ysJKhUGtyeFzcwKCPgmp5Cnj"

}

},

"users": ["56c32acd331dd81f8134f200"]

}

}

}

# WebAPI - AddNewInboundHook

**0.0.0**

Returns nothing

/api/v2/hook/in/add

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| targetType | Number | 1:user 2:group 3:room |
| targetId | String |  |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-AddNewInboundHook-0_0_0-0)

{

code: 1,

time: 1458827045780,

data: {

hook: {

\_\_v: 0,

userId: '56f3ef23a8769830b480f52a',

identifier: 'TcXAofnpRrEn',

targetType: 1,

targetId: '56f3ef23a8769830b480f52a',

hookType: 1,

url: '',

created: 1458827045779,

\_id: '56f3ef25a8769830b480f536'

}

}

}

# WebAPI - AddNewOutgoingHook

**0.0.0**

Returns nothing

/api/v2/hook/out/add

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| targetType | Number | 1:user 2:group 3:room |
| targetId | String |  |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-AddNewOutgoingHook-0_0_0-0)

{

code: 1,

time: 1458827160294,

data: {

hook: {

\_\_v: 0,

userId: '56f3ef96ab5396acb5346830',

identifier: '',

hookType: 2,

url: 'http://www.clover-studio.com',

created: 1458827160292,

\_id: '56f3ef98ab5396acb534683f'

}

}

}

# WebAPI - Block/Unblock user

**0.0.0**

Block/Unblock user

/api/v2/user/block

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| action | string | action |
| target | string | target |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Block-0_0_0-0)

{}

# WebAPI - BlockList

**0.0.0**

Returns nothing

/api/v2/user/blocklist

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-BlockList-0_0_0-0)

{}

# WebAPI - FavoriteList

**0.0.0**

Returns list of favorites

/api/v2/message/favorite/list/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-FavoriteList-0_0_0-0)

{

"code": 1,

"time": 1457364665523,

"data": {

"favorites": [{

"\_id": "56dd9b986aefa87d526cdf98",

"userId": "56c32acd331dd81f8134f200",

"messageId": "56cc6d880675f1d00532628f",

"created": 1457363864235,

"\_\_v": 0,

"messageModel": {

"\_id": "56cc6d880675f1d00532628f",

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "1-56c32acd331dd81f8134f200-56c32bf6db88293409a20299",

"message": "aa",

"localID": "\_fOpZHNDZiuPl8lN0btUfPbS54dQ6Vui8",

"type": 1,

"created": 1456237960593,

"\_\_v": 0,

"seenBy": [],

"userModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362952304,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"userModelTarget": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362225645,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

}

}

}, {

"\_id": "56dd9b659714d72f5212b904",

"userId": "56c32acd331dd81f8134f200",

"messageId": "56dd9522f3e5890d4700b9e5",

"created": 1457363813665,

"\_\_v": 0,

"messageModel": {

"\_id": "56dd9522f3e5890d4700b9e5",

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "2-56cf082ded51d2905e28a6d4",

"message": "zzz",

"localID": "\_z4a5fHmI2J55yOcbtmtt4MReZBiiXtD1",

"type": 1,

"created": 1457362210967,

"\_\_v": 0,

"seenBy": [],

"userModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362952304,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"groupModel": {

"\_id": "56cf082ded51d2905e28a6d4",

"organizationId": "56ab7b9061b760d9eb6feba3",

"name": "BlogXS",

"sortName": "BlogXS",

"description": "lorem quisque ut erat curabitur gravida nisi at nibh in hac habitasse platea dictumst aliquam augue",

"created": 1456408621251,

"\_\_v": 0,

"users": ["56c32bdfdb88293409a20234", "56c32b79db88293409a2008b", "56c32c02db88293409a202cb", "56c32c02db88293409a202c9"],

"avatar": {

"thumbnail": {

"originalName": "vAnLjyXV9FSnQo6TRDdtOUqBlMQATMXi",

"size": 1,

"mimeType": "image/png",

"nameOnServer": "vAnLjyXV9FSnQo6TRDdtOUqBlMQATMXi"

},

"picture": {

"originalName": "EU0IS6053tCts0Ar8sPQYPrcxtfqZPwX",

"size": 1,

"mimeType": "image/png",

"nameOnServer": "EU0IS6053tCts0Ar8sPQYPrcxtfqZPwX"

}

}

}

}

}, {

"\_id": "56dd9977ee7b28114fa651e6",

"userId": "56c32acd331dd81f8134f200",

"messageId": "56dd9527f3e5890d4700b9e6",

"created": 1457363319710,

"\_\_v": 0,

"messageModel": {

"\_id": "56dd9527f3e5890d4700b9e6",

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "3-56d9b199cdcc4b69755a5431",

"message": "zzz",

"localID": "\_ScfrTjXRdxjNNa3laUtclNpIKuGZggP4",

"type": 1,

"created": 1457362215637,

"\_\_v": 0,

"seenBy": [],

"userModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362952304,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726", "56cf08cded51d2905e28a754", "56cf092fed51d2905e28a783"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"roomModel": {

"\_id": "56d9b199cdcc4b69755a5431",

"owner": "56c32acd331dd81f8134f200",

"name": "Notification Test",

"created": 1457107353959,

"\_\_v": 0,

"avatar": {

"thumbnail": {

"originalName": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe",

"size": 118868,

"mimeType": "image/png",

"nameOnServer": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe"

},

"picture": {

"originalName": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe",

"size": 118868,

"mimeType": "image/png",

"nameOnServer": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe"

}

},

"users": ["56c32acd331dd81f8134f200", "56d375b7b92c46fc0403a8eb"]

}

}

}],

"count": 3

}

}

# WebAPI - Forward Message

**0.0.0**

Returns forward message

/api/v2/message/forward

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| messageId | string | messageId |
| roomId | string | roomId of Spika ( exp: 1-56cd9c45b84f5a1b393abb8f-56cdd6a358501da66d73a5dc) |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-ForwardMessage-0_0_0-0)

{

"code": 1,

"time": 1458051079100,

"data": {

"message": {

"\_\_v": 0,

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "3-56e7e9e138a97a86db83f3cf",

"message": "",

"localID": "\_roIF6lUllHg4RboTEr70iJHntIAuPLCc",

"type": 2,

"created": 1458051079096,

"\_id": "56e818070f2ba9df173e6289",

"seenBy": [],

"file": {

"thumb": {

"id": "56e8152b31906da71481643b",

"name": "thumb\_2014-06-03 17.23.38.jpg",

"size": 32200.000000000004,

"mimeType": "image/jpeg"

},

"file": {

"id": "56e8152b31906da71481643a",

"name": "2014-06-03 17.23.38.jpg",

"size": 1438407,

"mimeType": "image/jpeg"

}

}

}

}

}

# WebAPI - Get Online Status

**0.0.0**

Get users online status

/api/v2/user/online

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| users | Array | array of userid |

* [Request-Example:](http://document.spika.chat/docs/API/#parameter-examples-WebAPI-Get_Online_Status-0_0_0-0)

{

users: [

"56d36f91906bfe8a1046709b",

"56d36f91906bfe8a1046709c",

"56d36f91906bfe8a1046709d"

]

}

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Get_Online_Status-0_0_0-0)

{

code: 1,

time: 1456697234843,

data: [{

userId: '56d36f91906bfe8a1046709b',

onlineStatus: 1

}, {

userId: '56d36f91906bfe8a1046709c',

onlineStatus: 1

}, {

userId: '56d36f91906bfe8a1046709d',

onlineStatus: 0

}]

}

# WebAPI - Get list of users who've seen the message

**0.0.0**

Returns array of user model

/api/v2/message/seenby/messageid

# WebAPI - GroupAvatar

**0.0.0**

Returns image of group avatar

/api/v2/avatar/group/:fileID

# WebAPI - GroupDetail

**0.0.0**

Returns group detail

/api/v2/group/detail/:groupId

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-GroupDetail-0_0_0-0)

{

code: 1,

time: 1457091042485,

data: {

group: {

\_id: '56d971e027e87d9634468b13',

name: 'GROUP 1',

sortName: 'group 1',

description: 'GROUP 1 DESCRIPTION',

created: 1457091040863,

organizationId: '56d971e027e87d9634468b0e',

\_\_v: 0,

users: ['56d971e027e87d9634468b0f',

'56d971e027e87d9634468b10',

'56d971e027e87d9634468b11',

'56d971e027e87d9634468b12'

],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 46100,

mimeType: 'jpeg',

nameOnServer: 'PWxLHOeV2cNUtrBsCXOE0bRQ4Osz6CTp'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/jpeg',

nameOnServer: 'hncQaxqwl4jAQSBDfNeJtLNxvsyaSI39'

}

},

userModels: [{

\_id: '56d971e027e87d9634468b0f',

name: 'test',

userid: 'userid1Zbpz8',

password: '\*\*\*\*\*',

organizationId: '56d971e027e87d9634468b0e',

created: 1457091040829,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457091041276,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'wSstyxEpeSAhJIZPNTXlx2frF41Dk5Er'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'ThpaQcWLuaChaYMQbGBUYxtQeE01Lgj3'

}

}

}, {

\_id: '56d971e027e87d9634468b10',

name: 'User2',

userid: 'userid2zqX36',

password: '\*\*\*\*\*',

organizationId: '56d971e027e87d9634468b0e',

created: 1457091040840,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457091041271,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'BQG4BcxBAqHs3Y9ryx2wixJ1O6tfDgbA'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'RF3JINx0cBEKdkKZuuSIJM9X7VjEXoEV'

}

}

}, {

\_id: '56d971e027e87d9634468b11',

name: 'User3',

userid: 'userid3D1x0u',

password: '\*\*\*\*\*',

organizationId: '56d971e027e87d9634468b0e',

created: 1457091040844,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457091041272,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'Ih6Nod1NcnTmmkh5gioxeMYGf4VchKbJ'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'N3DLqQEZCko7PDqp4DLpEbyCdLB1Wz3s'

}

}

}, {

\_id: '56d971e027e87d9634468b12',

name: 'User4',

userid: 'userid43Ow7D',

password: '\*\*\*\*\*',

organizationId: '56d971e027e87d9634468b0e',

created: 1457091040848,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457091041276,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'p4BSqz4wscdiIA3bk6E5To3i9gui1ufG'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'jTxTQMgY0DbrIrF88ekrpJa3vlkukCad'

}

}

}]

}

}

}

# WebAPI - GroupUserList

**0.0.0**

Returns users of the group

/api/v2/group/users/:groupId/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-GroupUserList-0_0_0-0)

{ code: 1,

time: 1457096396842,

data:

{ count: 4,

list:

[ { \_id: '56d986cbfc6914974deb7676',

name: 'test',

userid: 'userid17RH5J',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395204,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395644,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'dDCn5wYHP7d8RkkwvoaTZxdckv8IRqDX' },

picture:

{ originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'RChBqbqMVD7ScoaTUcc2iSvVRTuoJybp' } } },

{ \_id: '56d986cbfc6914974deb7677',

name: 'User2',

userid: 'userid22eVJu',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395215,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395639,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'l4GFBU2lzNa42q7Rymx31q96u3XpgPbV' },

picture:

{ originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'ThremyfFhXrP6YlqxOAzva52Jpx7ulfx' } } },

{ \_id: '56d986cbfc6914974deb7678',

name: 'User3',

userid: 'userid3RP0zE',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395221,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395640,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'pILnZ13qS1PldslPmuGwPKC3FkwZrTP1' },

picture:

{ originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'Tqy2TWTJO9qblAixb578ZhiFOthENw7F' } } },

{ \_id: '56d986cbfc6914974deb7679',

name: 'User4',

userid: 'userid4tIcLG',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395228,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395644,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'Pit9pVfi3jVuWG2HbvlCoZzPOg8xQJwg' },

picture:

{ originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'Kvay7NvUPMvMiYkjPJvOogcVNLJZgejO' } } } ] } }

# WebAPI - Guest Signin

**0.0.0**

Returns new token for guest user.

/api/v2/user/signin/guest

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| organizationid | String | organizationid |
| userid | String | userid |
| pushToken | String | pushToken for push notification service |
| voipPushToken | String | pushToken for push notification service |
| secret | String | secret |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Guest_Signin-0_0_0-0)

{

code: 1,

time: 1454417582385,

data: {

newToken: 'DOqsolWe6zt3EFn0',

user: {

\_id: '56b0a6ae6753ea416ad58ea9',

name: 'user1',

userid: 'userid1ixfQJ',

password: '\*\*\*\*\*',

created: 1454417582354,

\_\_v: 0,

token: '\*\*\*\*\*',

tokenGeneratedAt: 1454417582384

},

organization: {

\_id: '56b0a6ae6753ea416ad58ea8',

organizationId: 'clover',

name: 'user1',

created: 1454417582336,

status: 1,

\_\_v: 0

}

}

}

# WebAPI - HooksList

**0.0.0**

Returns list of hooks of the caller user

/api/v2/user/hooks

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-HooksList-0_0_0-0)

{

code: 1,

time: 1458894561251,

data: {

hooks: [{

\_id: '56f4f6e14a8e10185fca7049',

userId: '56f4f6df4a8e10185fca703c',

identifier: 'I4lq8pTlOSDv',

targetType: 2,

targetId: '56f4f6df4a8e10185fca7040',

hookType: 1,

url: '',

created: 1458894561098,

\_\_v: 0,

groupModel: {

\_id: '56f4f6df4a8e10185fca7040',

name: 'GROUP 1',

sortName: 'group 1',

description: 'GROUP 1 DESCRIPTION',

created: 1458894559075,

organizationId: '56f4f6de4a8e10185fca703b',

\_\_v: 0,

users: ['56f4f6df4a8e10185fca703c',

'56f4f6df4a8e10185fca703d',

'56f4f6df4a8e10185fca703e',

'56f4f6df4a8e10185fca703f'

],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 11800,

mimeType: 'jpeg',

nameOnServer: '3KcOVW9HHXHyMrPRqNrJHUFUSBiUcQ4f'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/jpeg',

nameOnServer: 'J9aAM3LhnmUkUvg6LX9eLJXyyBTX8uNo'

}

}

}

}, {

\_id: '56f4f6e14a8e10185fca704a',

userId: '56f4f6df4a8e10185fca703c',

identifier: 'mdtNpPVLuVbb',

targetType: 3,

targetId: '56f4f6e04a8e10185fca7044',

hookType: 1,

url: '',

created: 1458894561107,

\_\_v: 0,

roomModel: {

\_id: '56f4f6e04a8e10185fca7044',

description: 'description',

owner: '56f4f6df4a8e10185fca703c',

name: 'Room1Changeg',

created: 1458894560293,

\_\_v: 0,

modified: 1458894560613,

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'p6fwL5coh17AcvkxZkS67i2nyGoFqDZQ'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'D0NsZIxcxnjWSvGuwXHKz4Fy5GYGl44c'

}

},

users: ['56f4f6df4a8e10185fca703c',

'56f4f6df4a8e10185fca703d',

'56f4f6df4a8e10185fca703e'

]

}

}, {

\_id: '56f4f6e14a8e10185fca704b',

userId: '56f4f6df4a8e10185fca703c',

identifier: '',

hookType: 2,

url: 'http://www.clover-studio.com',

targetType: 1,

targetId: '56f4f6df4a8e10185fca703c',

created: 1458894561160,

\_\_v: 0,

userModel: {

\_id: '56f4f6df4a8e10185fca703c',

name: 'test',

userid: 'userid1wAySm',

password: '\*\*\*\*\*',

organizationId: '56f4f6de4a8e10185fca703b',

created: 1458894559028,

status: 1,

\_\_v: 0,

description: null,

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'YrHMzm60C5aLlE0h95yfSp4rGDKUI00h'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'B1Uda2BmtPaDWbk9Sw3JTJL74bxZG188'

}

},

pushToken: ['pushtoken'],

token: [{

generateAt: 1458894559375,

token: '\*\*\*\*\*'

}]

}

}, {

\_id: '56f4f6e14a8e10185fca704c',

userId: '56f4f6df4a8e10185fca703c',

identifier: '',

hookType: 2,

url: 'http://www.clover-studio.com',

targetType: 2,

targetId: '56f4f6df4a8e10185fca7040',

created: 1458894561171,

\_\_v: 0,

groupModel: {

\_id: '56f4f6df4a8e10185fca7040',

name: 'GROUP 1',

sortName: 'group 1',

description: 'GROUP 1 DESCRIPTION',

created: 1458894559075,

organizationId: '56f4f6de4a8e10185fca703b',

\_\_v: 0,

users: ['56f4f6df4a8e10185fca703c',

'56f4f6df4a8e10185fca703d',

'56f4f6df4a8e10185fca703e',

'56f4f6df4a8e10185fca703f'

],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 11800,

mimeType: 'jpeg',

nameOnServer: '3KcOVW9HHXHyMrPRqNrJHUFUSBiUcQ4f'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/jpeg',

nameOnServer: 'J9aAM3LhnmUkUvg6LX9eLJXyyBTX8uNo'

}

}

}

}]

}

}

# WebAPI - Leave from room

**0.0.0**

Leave from joined room.

/api/v2/room/leave

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-Token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| roomId | string | roomId |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Leave_from_room-0_0_0-0)

{

success: 1,

data: {

}

}

# WebAPI - List user's room

**0.0.0**

Returns all rooms which caller user is owner

/api/v2/room/list/mine

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-List_user_s_room-0_0_0-0)

{

code: 1,

time: 1458829974637,

data: {

rooms: [{

\_id: '56f3fa956994704ec1b7fba5',

description: 'description',

owner: '56f3fa946994704ec1b7fb9d',

name: 'Room1Changeg',

created: 1458829973564,

\_\_v: 0,

modified: 1458829973909,

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: '1quUqnJh2dfSzgaMXoSpMCaQguL4mMgO'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'KCcOhqigP4tKYyZlB0IuCCvLZztg26fZ'

}

},

users: ['56f3fa946994704ec1b7fb9d',

'56f3fa946994704ec1b7fb9e',

'56f3fa946994704ec1b7fb9f'

]

}]

}

}

# WebAPI - MarkAll

**0.0.0**

Mark all as read

/api/v2/user/history/markchat

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-MarkAll-0_0_0-0)

{}

# WebAPI - MuteList

**0.0.0**

Returns nothing

/api/v2/user/mutelist

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-MuteList-0_0_0-0)

{}

# WebAPI - New Room

**0.0.0**

Create new conversation

/api/v2/room/new

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| name | name | of room. |
| description | string | Description |
| file | file | file |
| users | string | comma separated user ids. |

* [Request-Example:](http://document.spika.chat/docs/API/#parameter-examples-WebAPI-Create_New_room-0_0_0-0)

{

name : "name of conversation ", // if empty generates by users

useOld: false, // put true if use old conversation for same users

users: [

"563a0cc46cb168c8e9c4071d",

"563a0cc46cb168c8e9c4071a",

"563a0cc46cb168c8e9c4071b"

]

}

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Create_New_room-0_0_0-0)

{

code: 1,

time: 1455785008104,

data: {

room: {

\_\_v: 0,

owner: '56c5842faea1bfac4a657bbd',

name: 'room1',

created: 1455785008046,

\_id: '56c58430aea1bfac4a657bc1',

avatar: {

thumbnail: {

originalName: 'dfhSvtAQyCGMALxs4AZXzuZVOrXSmbfg',

size: 136825,

mimeType: 'image/png',

nameOnServer: 'dfhSvtAQyCGMALxs4AZXzuZVOrXSmbfg'

},

picture: {

originalName: 'dfhSvtAQyCGMALxs4AZXzuZVOrXSmbfg',

size: 136825,

mimeType: 'image/png',

nameOnServer: 'dfhSvtAQyCGMALxs4AZXzuZVOrXSmbfg'

}

},

users: ['56c5842faea1bfac4a657bbd',

'56c5842faea1bfac4a657bbe',

'56c5842faea1bfac4a657bbf',

'56c5842faea1bfac4a657bc0'

]

}

}

}

# WebAPI - Remove users from room

**0.0.0**

Update profile of conversation

/api/v2/room/users/remove

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-Token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| roomId | string | roomId |
| users | Array | Array of user ids |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Remove_users_from_room-0_0_0-0)

{

"code": 1,

"time": 1457082886197,

"data": {

"room": {

"\_id": "56d94c88bf06a1f30ad6091d",

"owner" : "56c32acd331dd81f8134f200"

"ownerModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457082869691,

"token": "\*\*\*\*\*",

"departments": [],

"groups": ["56cf0a60ed51d2905e28a848"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"name": "ああああああ",

"created": 1457081480907,

"\_\_v": 0,

"description": "いいいいいddd",

"modified": 1457082886193,

"avatar": {

"thumbnail": {

"originalName": "2014-06-03 17.23.39.jpg",

"size": 1504586,

"mimeType": "image/png",

"nameOnServer": "ut4G1A3Jq9LbfeDxXUh8jibgDB4wPGV1"

},

"picture": {

"originalName": "2014-06-03 17.23.39.jpg",

"size": 1504586,

"mimeType": "image/png",

"nameOnServer": "egStPNb3ysJKhUGtyeFzcwKCPgmp5Cnj"

}

},

"users": ["56c32acd331dd81f8134f200"]

}

}

}

# WebAPI - RemoveFromFavorite

**0.0.0**

Remove from favorite

/api/v2/message/favorite/remove

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| messageId | string | messageId |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-RemoveFromFavorite-0_0_0-0)

{

"code": 1,

"time": 1457363319718

}

# WebAPI - RemoveInboundHook

**0.0.0**

Remove from favorite

/api/v2/hook/in/remove

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| hookId | string | hookId |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-RemoveInboundHook-0_0_0-0)

{

"code": 1,

"time": 1457363319718

}

# WebAPI - RoomAvatar

**0.0.0**

Returns image of room avatar

/api/v2/avatar/room/:fileID

# WebAPI - RoomDetail

**0.0.0**

Returns room detail

/api/v2/room/detail/:roomId

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-RoomDetail-0_0_0-0)

{

code: 1,

time: 1457092969492,

data: {

room: {

\_id: '56d979690c0ce24041304b43',

description: 'description',

owner: '56d979670c0ce24041304b3b',

name: 'Room1Changeg',

created: 1457092969060,

\_\_v: 0,

modified: 1457092969368,

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'p9qj4omrwLGRJ2HY0ycDGRBnWZ8sSdNC'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'm5K6IcDbFPLA7Rb802JyLgdUENnp065j'

}

},

users: ['56d979670c0ce24041304b3b',

'56d979670c0ce24041304b3c',

'56d979670c0ce24041304b3d'

],

userModels: [{

\_id: '56d979670c0ce24041304b3b',

name: 'test',

userid: 'userid1VLoM0',

password: '\*\*\*\*\*',

organizationId: '56d979670c0ce24041304b3a',

created: 1457092967847,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457092968300,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'If9mccjOJGIDezaV9s8B3wtLLvIL1QcB'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'CXOAffrBqgWfXzr5ytywezBZh4KVUbJV'

}

}

}, {

\_id: '56d979670c0ce24041304b3c',

name: 'User2',

userid: 'userid2AztOz',

password: '\*\*\*\*\*',

organizationId: '56d979670c0ce24041304b3a',

created: 1457092967858,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457092968295,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'g7AmtxYWVfZxFWzoc5LFytnVd9LhLiUu'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'cOObVhMSjSxTHiw4UWBdy28A2bvSaevn'

}

}

}, {

\_id: '56d979670c0ce24041304b3d',

name: 'User3',

userid: 'userid3xPYBs',

password: '\*\*\*\*\*',

organizationId: '56d979670c0ce24041304b3a',

created: 1457092967864,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457092968297,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'AVa6NQdeKc83NPYvy9Y7G5RJdDKN6Fpw'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'druLJc3iituP5BAiHx7sNMmNuKkjd3rP'

}

}

}]

}

}

}

# WebAPI - RoomUserList

**0.0.0**

Returns users of the room. If send "all" text as page param, it returns all user.

/api/v2/room/users/:roomId/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-RoomUserList-0_0_0-0)

{ code: 1,

time: 1457096396842,

data:

{ count: 4,

list:

[ { \_id: '56d986cbfc6914974deb7676',

name: 'test',

userid: 'userid17RH5J',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395204,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395644,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'dDCn5wYHP7d8RkkwvoaTZxdckv8IRqDX' },

picture:

{ originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'RChBqbqMVD7ScoaTUcc2iSvVRTuoJybp' } } },

{ \_id: '56d986cbfc6914974deb7677',

name: 'User2',

userid: 'userid22eVJu',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395215,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395639,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'l4GFBU2lzNa42q7Rymx31q96u3XpgPbV' },

picture:

{ originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'ThremyfFhXrP6YlqxOAzva52Jpx7ulfx' } } },

{ \_id: '56d986cbfc6914974deb7678',

name: 'User3',

userid: 'userid3RP0zE',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395221,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395640,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'pILnZ13qS1PldslPmuGwPKC3FkwZrTP1' },

picture:

{ originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'Tqy2TWTJO9qblAixb578ZhiFOthENw7F' } } },

{ \_id: '56d986cbfc6914974deb7679',

name: 'User4',

userid: 'userid4tIcLG',

password: '\*\*\*\*\*',

organizationId: '56d986cbfc6914974deb7675',

created: 1457096395228,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457096395644,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar:

{ thumbnail:

{ originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'Pit9pVfi3jVuWG2HbvlCoZzPOg8xQJwg' },

picture:

{ originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'Kvay7NvUPMvMiYkjPJvOogcVNLJZgejO' } } } ] } }

# WebAPI - Save Notes for chat

**0.0.0**

Add to callers favorite

/api/v2/notes

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| chatId | string | chatId |
| note | string | note |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-AddToFavorite-0_0_0-0)

{

"code": 1,

"time": 1457363319718,

"data": {

"note": {

"\_\_v": 0,

"chatId": "1-56c32acd331dd81f8134f200-56c32acd331dd81f8134f200",

"note": "text",

"created": 1457363319710,

"modified": "1457363319710"

}

}

}

# WebAPI - SavePushToken

**0.0.0**

Save push token to user model

/api/v2/user/savepushtoken

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| pushToken | String | pushToken for push notification service |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-SavePushToken-0_0_0-0)

{

code: 1,

time: 1454417582385,

data: {

user: {

\_id: '56b0a6ae6753ea416ad58ea9',

name: 'user1',

userid: 'userid1ixfQJ',

password: '\*\*\*\*\*',

created: 1454417582354,

\_\_v: 0,

token: '\*\*\*\*\*',

tokenGeneratedAt: 1454417582384,

pushToken : ["test","test1"]

}

}

}

# WebAPI - SaveVoipPushToken

**0.0.0**

Save Voip push token to user model

/api/v2/user/savevoippushtoken

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| pushToken | String | pushToken for push notification service |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-SaveVoipPushToken-0_0_0-0)

{

code: 1,

time: 1454417582385,

data: {

user: {

\_id: '56b0a6ae6753ea416ad58ea9',

name: 'user1',

userid: 'userid1ixfQJ',

password: '\*\*\*\*\*',

created: 1454417582354,

\_\_v: 0,

token: '\*\*\*\*\*',

tokenGeneratedAt: 1454417582384,

pushToken : ["test","test1"]

}

}

}

# WebAPI - Search Message

**0.0.0**

Returns messages. [!NOTICE] This API doesn't returns same row count for each page. Stop paging when it returns 0 result. Match target is text message and file name.

/api/v2/message/search/:keyword/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| keyword | String | URL encoded keyword |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Search_Message-0_0_0-0)

{

"code": 1,

"time": 1457362232696,

"data": {

"messages": [{

"\_id": "56dd952ef3e5890d4700b9e7",

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "1-56c32acd331dd81f8134f200-56d963dc4c7b0c89201acde7",

"message": "zzz",

"localID": "\_QTuVRdVXmmnYL4bM2cOGtWXnYqVQgkUZ",

"type": 1,

"created": 1457362222564,

"\_\_v": 0,

"seenBy": [],

"userModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362225645,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"userModelTarget": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362225645,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

}

}, {

"\_id": "56dd9522f3e5890d4700b9e5",

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "2-56cf082ded51d2905e28a6d4",

"message": "zzz",

"localID": "\_z4a5fHmI2J55yOcbtmtt4MReZBiiXtD1",

"type": 1,

"created": 1457362210967,

"\_\_v": 0,

"seenBy": [],

"userModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362225645,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726", "56cf08cded51d2905e28a754"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"groupModel": {

"\_id": "56cf082ded51d2905e28a6d4",

"organizationId": "56ab7b9061b760d9eb6feba3",

"name": "BlogXS",

"sortName": "BlogXS",

"description": "lorem quisque ut erat curabitur gravida nisi at nibh in hac habitasse platea dictumst aliquam augue",

"created": 1456408621251,

"\_\_v": 0,

"users": ["56c32bdfdb88293409a20234", "56c32b79db88293409a2008b", "56c32c02db88293409a202cb"],

"avatar": {

"thumbnail": {

"originalName": "vAnLjyXV9FSnQo6TRDdtOUqBlMQATMXi",

"size": 1,

"mimeType": "image/png",

"nameOnServer": "vAnLjyXV9FSnQo6TRDdtOUqBlMQATMXi"

},

"picture": {

"originalName": "EU0IS6053tCts0Ar8sPQYPrcxtfqZPwX",

"size": 1,

"mimeType": "image/png",

"nameOnServer": "EU0IS6053tCts0Ar8sPQYPrcxtfqZPwX"

}

}

}

}, {

"\_id": "56dd9527f3e5890d4700b9e6",

"user": "56c32ae5331dd81f8134f201",

"userID": "56c32acd331dd81f8134f200",

"roomID": "3-56d9b199cdcc4b69755a5431",

"message": "zzz",

"localID": "\_ScfrTjXRdxjNNa3laUtclNpIKuGZggP4",

"type": 1,

"created": 1457362215637,

"\_\_v": 0,

"seenBy": [],

"userModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457362225645,

"token": "\*\*\*\*\*",

"groups": ["56cf0a60ed51d2905e28a848", "56cf0897ed51d2905e28a726", "56cf08cded51d2905e28a754"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"roomModel": {

"\_id": "56d9b199cdcc4b69755a5431",

"owner": "56c32acd331dd81f8134f200",

"name": "Notification Test",

"created": 1457107353959,

"\_\_v": 0,

"avatar": {

"thumbnail": {

"originalName": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe",

"size": 118868,

"mimeType": "image/png",

"nameOnServer": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe"

},

"picture": {

"originalName": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe",

"size": 118868,

"mimeType": "image/png",

"nameOnServer": "ZrylJtaVOkhHoCiz7R1kPcuTU8mChIZe"

}

},

"users": ["56c32acd331dd81f8134f200", "56d375b7b92c46fc0403a8eb"]

}

}],

"count": 3

}

}

# WebAPI - Search Room

**0.0.0**

Returns rooms matches to the keyword

/api/v2/room/search/:page?keyword=keyword

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Search_Room-0_0_0-0)

{

"code": 1,

"time": 1458048275869,

"data": {

"list": [

{

"\_id": "56e19ec20dc358a4c4c1bab3",

"owner": "56c32acd331dd81f8134f200",

"name": "Ken",

"created": 1457626818892,

"\_\_v": 0,

"description": "test",

"modified": 1457962642422,

"avatar": {

"thumbnail": {

"originalName": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1",

"size": 112325,

"mimeType": "image/png",

"nameOnServer": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1"

},

"picture": {

"originalName": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1",

"size": 112325,

"mimeType": "image/png",

"nameOnServer": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1"

}

},

"users": ["56c32acd331dd81f8134f200"],

"type": 3

}

],

"count": 1

}

}

# WebAPI - Search user/group/room

**0.0.0**

Returns users groups rooms matches to the keyword

/api/v2/search/all/:page?keyword=keyword

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-SearchAll-0_0_0-0)

{

"code": 1,

"time": 1458048275869,

"data": {

"list": [{

"\_id": "56e19ec20dc358a4c4c1bab3",

"owner": "56c32acd331dd81f8134f200",

"name": "Ken",

"created": 1457626818892,

"\_\_v": 0,

"description": "test",

"modified": 1457962642422,

"avatar": {

"thumbnail": {

"originalName": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1",

"size": 112325,

"mimeType": "image/png",

"nameOnServer": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1"

},

"picture": {

"originalName": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1",

"size": 112325,

"mimeType": "image/png",

"nameOnServer": "wAO6OLzQqrdRaT9xP15pFmcCJGcKSuo1"

}

},

"users": ["56c32acd331dd81f8134f200"],

"type": 3

}, {

"\_id": "56e6b6ec06552124125cdb88",

"owner": "56c32acd331dd81f8134f200",

"name": "Ken Test 1",

"created": 1457960684510,

"\_\_v": 0,

"description": "ddddd",

"modified": 1457968944671,

"avatar": {

"thumbnail": {

"originalName": "2014-06-03 17.23.38.jpg",

"size": 1438407,

"mimeType": "image/png",

"nameOnServer": "D5SXoOQKhDnHZOJWTgla2jV00uIhaElp"

},

"picture": {

"originalName": "2014-06-03 17.23.38.jpg",

"size": 1438407,

"mimeType": "image/png",

"nameOnServer": "BKg0DwlNPvI7tyUKZ7I1C2aMmusMzK9R"

}

},

"users": ["56c32acd331dd81f8134f200", "56e6b6d206552124125cdb86", "56e6b73d06552124125cdb92"],

"type": 3

}, {

"\_id": "56cdd4268c2de60868f7d7cc",

"owner": "56c32acd331dd81f8134f200",

"name": "Test Room",

"created": 1456329766450,

"\_\_v": 0,

"avatar": {

"thumbnail": {

"originalName": "ar12oQdbhh2QFOR9FzMVmi9ZRoc9j7KR",

"size": 127310,

"mimeType": "image/png",

"nameOnServer": "ar12oQdbhh2QFOR9FzMVmi9ZRoc9j7KR"

},

"picture": {

"originalName": "ar12oQdbhh2QFOR9FzMVmi9ZRoc9j7KR",

"size": 127310,

"mimeType": "image/png",

"nameOnServer": "ar12oQdbhh2QFOR9FzMVmi9ZRoc9j7KR"

}

},

"users": ["56c32acd331dd81f8134f200", "56c32b35db88293409a1ff82", "56c32b53db88293409a1ffff", "56c32b59db88293409a20019", "56c32b5edb88293409a2002f", "56c32b66db88293409a2004a", "56c32b7bdb88293409a20091", "56c32b8bdb88293409a200cf", "56c32bb0db88293409a20170", "56c32bb4db88293409a2017e", "56c32bd6db88293409a2020e", "56c32be8db88293409a2025e", "56c32bf3db88293409a2028c", "56c32bf6db88293409a20299", "56c32bf7db88293409a2029c", "56c32c21db88293409a2034e"],

"type": 3

}, {

"\_id": "56d95be8a852b06d19b0d537",

"description": "test",

"owner": "56c32acd331dd81f8134f200",

"name": "Test Room",

"created": 1457085416115,

"\_\_v": 0,

"avatar": {

"thumbnail": {

"originalName": "Sxm8Hjk7M6XmHdeGLOjuHZeVH8veSLq4",

"size": 113213,

"mimeType": "image/png",

"nameOnServer": "Sxm8Hjk7M6XmHdeGLOjuHZeVH8veSLq4"

},

"picture": {

"originalName": "Sxm8Hjk7M6XmHdeGLOjuHZeVH8veSLq4",

"size": 113213,

"mimeType": "image/png",

"nameOnServer": "Sxm8Hjk7M6XmHdeGLOjuHZeVH8veSLq4"

}

},

"users": ["56c32acd331dd81f8134f200", "56c32b5edb88293409a2002f", "56c32baddb88293409a20162"],

"type": 3

}, {

"\_id": "56d95d27437835f2199ebbe3",

"description": "Test Room",

"owner": "56c32acd331dd81f8134f200",

"name": "Test Room1",

"created": 1457085735694,

"\_\_v": 0,

"avatar": {

"thumbnail": {

"originalName": "tL3TxBiNX04SBYAjjmIbgT7Um42L2i6b",

"size": 115851,

"mimeType": "image/png",

"nameOnServer": "tL3TxBiNX04SBYAjjmIbgT7Um42L2i6b"

},

"picture": {

"originalName": "tL3TxBiNX04SBYAjjmIbgT7Um42L2i6b",

"size": 115851,

"mimeType": "image/png",

"nameOnServer": "tL3TxBiNX04SBYAjjmIbgT7Um42L2i6b"

}

},

"users": ["56c32acd331dd81f8134f200", "56c32b5edb88293409a2002f", "56c32b66db88293409a2004a", "56c32bf7db88293409a2029c"],

"type": 3

}, {

"\_id": "56e7ef842500caf2eda5d5fb",

"name": "atest",

"sortName": "atest",

"description": "",

"created": 1458040708283,

"organizationId": "56ab7b9061b760d9eb6feba3",

"type": 2,

"\_\_v": 0,

"users": ["56c32acd331dd81f8134f200"],

"usersCount": 1,

"userModels": [{

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458048265940,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

}]

}, {

"\_id": "56e7da1671be6b38d3beb967",

"name": "test",

"sortName": "test",

"description": "test",

"created": 1458035222590,

"organizationId": "56ab7b9061b760d9eb6feba3",

"type": 2,

"\_\_v": 0,

"users": ["56c32acd331dd81f8134f200"],

"usersCount": 1,

"userModels": [{

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458048265940,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

}]

}],

"count": 7

}

}

# WebAPI - Send Message

**0.0.0**

Returns messageobj

/api/v2/message/send

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| roomID | String | roomID |
| message | String | message |
| type | String | message type |
| file | String | file( if eixsts) |
| thumb | String | thumb ( if picture ) |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-SendMessage-0_0_0-0)

{

"code": 1,

"time": 1462805270534,

"data": {

"message": {

"\_\_v": 0,

"user": "56ec127aa4718ef424641693",

"userID": "56ec126ca4718ef424641692",

"roomID": "1-56ec126ca4718ef424641692-572b3fdd52ae03995757478e",

"message": "test",

"type": 1,

"created": 1462805270529,

"\_id": "5730a316cce6a28d3afd8eee",

"seenBy": []

}

}

}

## 發送樣品申請

Top of Form

> 網址

### Headers

#### Header

access-token

>  String

### Parameters

#### Parameter

roomID

>  String

message

>  String

type

>  String

file.file

>  String

file.thumb

>  String

發送

Bottom of Form

# WebAPI - ShowSticker

**0.0.0**

Returns image of the sticker

/api/v2/sticker/:fileID

# WebAPI - SignOut

**0.0.0**

nothing

/api/v2/user/signout

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Signin-0_0_0-0)

{

code: 1,

time: 1454417582385

}

# WebAPI - Signup Finish

**0.0.0**

Updates a user

/api/v2/user/signup/finish

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| name | String | display name |
| password | String | hashed password |
| secret | String | secret |
| file | String | avatar file |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Signup_Finish-0_0_0-0)

{

code: 1,

time: 1454417582385,

data: {

"user": {

"\_id": "588b4f191f8eae1954e9a37c",

"organizationId": "test",

"created": 1485524761592,

"phoneNumber": "+385989057351",

"userid": "+385989057351",

"activationCode": "815443",

"status": 1,

"\_\_v": 1,

"UUID": [],

"devices": [],

"blocked": [],

"muted": [],

"groups": [],

"voipPushToken": [],

"pushToken": [],

"token": [

{

"generateAt": 1485524904386,

"token": "\*\*\*\*\*"

}

]

}

}

}

# WebAPI - Signup Send Sms

**0.0.0**

Send SMS verification code to user

/api/v2/user/signup/sendSms

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| organizationId | String | organizationId |
| phoneNumber | String | phoneNumber |
| isWeb | Number | if api is called from web (1), otherwise (0)  允許值: 0, 1 |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Signup_Send_Sms-0_0_0-0)

{

code: 1,

time: 1454417582385,

data: {}

}

# WebAPI - Signup Verify Sms

**0.0.0**

Returns new token for the user.

/api/v2/user/signup/verify

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| uuid | String | uuid |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| activationCode | String | Activation code received in SMS |
| pushToken | String | pushToken |
| voipPushToken | String | voipPushToken |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Signup_Verify_Sms-0_0_0-0)

{

code: 1,

time: 1454417582385,

data: {

"newToken": "uSKuT4qC9LQDRHJv",

"user": {

"\_id": "588b4f191f8eae1954e9a37c",

"organizationId": "test",

"created": 1485524761592,

"phoneNumber": "+385989057351",

"userid": "+385989057351",

"activationCode": "815443",

"status": 1,

"\_\_v": 1,

"UUID": [],

"devices": [],

"blocked": [],

"muted": [],

"groups": [],

"voipPushToken": [],

"pushToken": [],

"token": [

{

"generateAt": 1485524904386,

"token": "\*\*\*\*\*"

}

]

}

}

}

# WebAPI - Sticker List

**0.0.0**

Returns list of stickers

/api/v2/stickers

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-StickerList-0_0_0-0)

{

"code": 1,

"time": 1458594799210,

"data": {

"stickers": [{

"name": "sticker1",

"list": [{

"fullPic": "/api/v2/sticker/M7mlnDGKvZwbpTdtUYJfAQkx87Sqq2jk",

"smallPic": "/api/v2/sticker/RQId55OV5rAA5PDSXehKkmfvu1yBVtfm"

}, {

"fullPic": "/api/v2/sticker/IAy3Fsm5Ldp0iUYpnhalFZE96bzCG89C",

"smallPic": "/api/v2/sticker/y1tAUeqrSnHbLiEIxzOnY5L9ycX2pb4J"

}, {

"fullPic": "/api/v2/sticker/MT2ZS0yfcD4ANykJGmgZsCvbk4r3GavX",

"smallPic": "/api/v2/sticker/eoVFhVw00wgV7tlZBD1qWCtCJltTyHI9"

}],

"mainTitlePic": "/api/v2/sticker/y1tAUeqrSnHbLiEIxzOnY5L9ycX2pb4J"

}, {

"name": "sticker2",

"list": [{

"fullPic": "/api/v2/sticker/crbsy6L64dMH2sqQ2Dwajowc7iVXd4S4",

"smallPic": "/api/v2/sticker/bGIzNJ3618UmpKtFKM3HoYA43VM4afTQ"

}, {

"fullPic": "/api/v2/sticker/faSBTStGxeiSR8ojFXxbK9mZIiE4nFhO",

"smallPic": "/api/v2/sticker/9A98VuyQavssHEFAY872efM9JHFqfm9L"

}, {

"fullPic": "/api/v2/sticker/eAmxxxQVuxWANxHnfs9vht0wqLJgX7na",

"smallPic": "/api/v2/sticker/Yxv63CSDwHTK6lSOqdPMztQWePdxCt2j"

}],

"mainTitlePic": "/api/v2/sticker/bGIzNJ3618UmpKtFKM3HoYA43VM4afTQ"

}]

}

}

# WebAPI - Updaet Password

**0.0.0**

Update password of request user

/api/v2/user/updatepassword

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| currentPassword | string | hashed current password |
| newPassword | string | send raw string here |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Updaet_Password-0_0_0-0)

{

success: 1,

data: {

}

}

# WebAPI - Update Message Delivered To

**0.0.0**

@apiHeader {String} access-token Users unique access-token.

/api/v2/message/deliver

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| messageId | String | messageId |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Update_Message_Delivered_To-0_0_0-0)

{

"code": 1,

"time": 1457363319718,

"data": {}

}

# WebAPI - Update Profile

**0.0.0**

Update profile of request user

/api/v2/user/update

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| name | string | Name to display |
| description | string | Description |
| file | file | avatar file |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Update_Profile-0_0_0-0)

{

code: 1,

time: 1455627869209,

data: {

user: {

\_id: '56c31e5c8402867d76843c7b',

name: 'test',

userid: 'userid13fXan',

password: '\*\*\*\*\*',

organizationId: '56c31e5c8402867d76843c7a',

created: 1455627868973,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1455627869008,

token: '\*\*\*\*\*',

description: null,

groups: [],

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'te0z9HsG9qOfxVwNawqS4CSnX1AovEMq'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'P5AuA6bZ8qwdkf71cIWl32K4KmvMETyp'

}

}

}

}

# WebAPI - Update Room Profile

**0.0.0**

Update profile of conversation

/api/v2/room/update

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| roomId | string | roomId |
| name | string | Name to display |
| description | string | Description |
| file | file | avatar file |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Update_Room_Profile-0_0_0-0)

{

"code": 1,

"time": 1457082886197,

"data": {

"room": {

"\_id": "56d94c88bf06a1f30ad6091d",

"owner" : "56c32acd331dd81f8134f200"

"ownerModel": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1457082869691,

"token": "\*\*\*\*\*",

"departments": [],

"groups": ["56cf0a60ed51d2905e28a848"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"name": "ああああああ",

"created": 1457081480907,

"\_\_v": 0,

"description": "いいいいいddd",

"modified": 1457082886193,

"avatar": {

"thumbnail": {

"originalName": "2014-06-03 17.23.39.jpg",

"size": 1504586,

"mimeType": "image/png",

"nameOnServer": "ut4G1A3Jq9LbfeDxXUh8jibgDB4wPGV1"

},

"picture": {

"originalName": "2014-06-03 17.23.39.jpg",

"size": 1504586,

"mimeType": "image/png",

"nameOnServer": "egStPNb3ysJKhUGtyeFzcwKCPgmp5Cnj"

}

},

"users": ["56c32acd331dd81f8134f200"]

}

}

}

# WebAPI - UpdateInboundHook

**0.0.0**

Returns hook

/api/v2/hook/in/update

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| hookId | String |  |
| targetType | Number | 1:user 2:group 3:room |
| targetId | String |  |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-UpdateInboundHook-0_0_0-0)

{

code: 1,

time: 1458827082658,

data: {

hook: {

\_id: '56f3ef4a716487a4b40ee081',

userId: '56f3ef48716487a4b40ee075',

identifier: 'WMJbh5HtdBiO',

targetType: 3,

targetId: '56f3ef49716487a4b40ee07d',

hookType: 1,

url: '',

created: 1458827082625,

\_\_v: 0

}

}

}

# WebAPI - UpdateOutgoingHook

**0.0.0**

Returns hook

/api/v2/hook/out/update

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| hookId | String |  |
| targetType | Number | 1:user 2:group 3:room |
| targetId | String |  |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-UpdateOutgoingHook-0_0_0-0)

{

code: 1,

time: 1458827160294,

data: {

hook: {

\_\_v: 0,

userId: '56f3ef96ab5396acb5346830',

identifier: '',

hookType: 2,

url: 'http://www.clover-studio.com',

created: 1458827160292,

\_id: '56f3ef98ab5396acb534683f'

}

}

}

# WebAPI - User Delete Contact

**0.0.0**

User Delete Contact

/api/v2/user/deleteContact/:contactId

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-User_Delete_Contact-0_0_0-0)

{

"code": 1,

"time": 1507293117920,

"data": {}

}

# WebAPI - User Get Contacts

**0.0.0**

User Get Contacts

/api/v2/user/getContacts

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-User_Get_Contacts-0_0_0-0)

{

"code": 1,

"time": 1507293117920,

"data": {

"users": [

{

"\_id": "59df1bde5f62d1632b3ba381",

"created": 1507793886231,

"phoneNumber": "+385989057351",

"userid": "+385989057351",

"name": "Jura",

"description": "jura opis",

"avatar": {

"thumbnail": {

"originalName": "images.jpg",

"size": 7897,

"mimeType": "jpeg",

"nameOnServer": "pYasg5hAJG1YFmPRFQjFMZ2XeBR021EY"

},

"picture": {

"originalName": "images.jpg",

"size": 3269,

"mimeType": "image/jpeg",

"nameOnServer": "WKYwoIcIG4PVykzFudK4Z1jjQQGzyYpp"

}

},

"userContactName": "jura mobile"

}

]

}

}

# WebAPI - User Sync Contacts

**0.0.0**

User Sync Contacts

/api/v2/user/sync

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| organization | String | organization name |
| phoneNumbers | String | should receive with all numbers like this: "+385987654324,+385998765456,+385916342536" |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-User_Sync_Contacts-0_0_0-0)

{

"code": 1,

"time": 1507293117920,

"data": {

"users": [

{

"\_id": "59d5f9dfc4282cb82e229519",

"phoneNumber": "+385976376676",

"userid": "+385976376676",

"name": "ivo2345",

"description": ""

},

{

"\_id": "59d607f4ad8f7e8b23c10af2",

"phoneNumber": "+385989057351",

"userid": "+385989057351",

"name": "Jura",

"description": "debeli jura",

"avatar": {

"thumbnail": {

"originalName": "images.jpg",

"size": 7897,

"mimeType": "jpeg",

"nameOnServer": "udbipys2F14jrRC5a1L19HSGKL8MpIaC"

},

"picture": {

"originalName": "images.jpg",

"size": 3269,

"mimeType": "image/jpeg",

"nameOnServer": "bq1EeoYyK5yOsVRqSywPXk80nYdv8jK3"

}

}

}

]

}

}

# WebAPI - User Update Contacts

**0.0.0**

User Update Contacts

/api/v2/user/updateContacts

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| contacts | Object[] | object array of contacts |
| id | String |  |
| name | String |  |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-User_Update_Contacts-0_0_0-0)

{

"code": 1,

"time": 1507293117920,

"data": {}

}

# WebAPI - UserAvatar

**0.0.0**

Returns image of user avatar

/api/v2/avatar/user/:fileID

# WebAPI - UserDetail

**0.0.0**

Returns user detail

/api/v2/user/detail/:userId

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-UserDetail-0_0_0-0)

{

"code": 1,

"time": 1457090503752,

"data": {

"user": {

"\_id": "56c32b53db88293409a1ffff",

"organizationId": "56ab7b9061b760d9eb6feba3",

"name": "Adam Perez",

"description": "aperez40@wix.com",

"userid": "aperez40",

"password": "\*\*\*\*\*",

"status": 1,

"created": 1455631187121,

"\_\_v": 0,

"departments": [],

"groups": ["56cefbf39543eb2026ba8d5e", "56cefc189543eb2026ba8daa", "56cefc279543eb2026ba8dcc", "56cefc739543eb2026ba8e56", "56cefcd79543eb2026ba8f15", "56cefce19543eb2026ba8f27", "56cf005fff6fef543ccd5a27", "56cf00ea9016d2133ebc01a9", "56cf01db68b9ddba3e63756f", "56cf02b568b9ddba3e637676", "56cf02cb68b9ddba3e637696", "56cf032668b9ddba3e6376fe", "56cf0491235996c850d0b99e", "56cf049b235996c850d0b9a9", "56cf05e6235996c850d0bad9", "56cf0651235996c850d0bb3e", "56cf079bed51d2905e28a642", "56cf0839ed51d2905e28a6df", "56cf085fed51d2905e28a700", "56cf0927ed51d2905e28a780", "56cf0b31ed51d2905e28a8f4", "56cf0b8aed51d2905e28a938", "56cf0bf8ed51d2905e28a98f", "56cf0bfced51d2905e28a991"],

"avatar": {

"thumbnail": {

"originalName": "ljHWzRLBrOFFOrccltYPc8F8MbQgPoKN",

"size": 1,

"mimeType": "image/png",

"nameOnServer": "ljHWzRLBrOFFOrccltYPc8F8MbQgPoKN"

},

"picture": {

"originalName": "Sczhz0ogE7ofXMkp2et39EfS712oR9A4",

"size": 1,

"mimeType": "image/png",

"nameOnServer": "Sczhz0ogE7ofXMkp2et39EfS712oR9A4"

}

},

"groupModels": []

}

}

}

# WebAPI - WebHookReceiver

**0.0.0**

Returns messageId

/api/v2/hook/r/:hookIdentifier

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-WebHookReceiver-0_0_0-0)

{

"code": 1,

"time": 1457090503752,

"data": {

"messageId": 121212

}

}

# WebAPI - download file

**0.0.0**

Returns file binary

/api/v2/file/fileId

# WebAPI - get dictionary

**0.0.0**

get dictionary basend on cookie saved lang

/api/v2/lang/get

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Get_Dictionary-0_0_0-0)

{}

# WebAPI - grouplist

**0.0.0**

Returns all groups

/api/v2/group/list/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-List_Group-0_0_0-0)

{

code: 1,

time: 1457086011856,

data: {

list: [{

\_id: '56d95e3a999a71831bfdea65',

name: 'GROUP 1',

sortName: 'group 1',

description: 'GROUP 1 DESCRIPTION',

created: 1457086010379,

organizationId: '56d95e3a999a71831bfdea60',

\_\_v: 0,

users: ['56d95e3a999a71831bfdea61',

'56d95e3a999a71831bfdea62',

'56d95e3a999a71831bfdea63',

'56d95e3a999a71831bfdea64'

],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 46100,

mimeType: 'jpeg',

nameOnServer: '6CWKh3oDPZZOECEA6vzUknFMy5U1gRrT'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/jpeg',

nameOnServer: 'eNJdtFNW4bu9IZdQbt4oJwms7WbRelUC'

}

},

usersCount: 4,

userModels: [{

\_id: '56d95e3a999a71831bfdea61',

name: 'test',

userid: 'userid1OlegK',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010345,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'QLBDbupbTqPwPKQAXbq9gRop1OT7IHkc'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'DyFSaCRcPddPCQKJyAXnbCL6icqgkyIt'

}

}

}, {

\_id: '56d95e3a999a71831bfdea62',

name: 'User2',

userid: 'userid2On3Kv',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010355,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010780,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'NTKiNzll6pCsEZUqba3LTD5MFk8i7Uk2'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'ZKQgT2fJ6RPOSz34qKyX7FDPKiqJfeWo'

}

}

}, {

\_id: '56d95e3a999a71831bfdea63',

name: 'User3',

userid: 'userid3FUG8n',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010360,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010782,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'l7IXqnsOIMeivPZqy9k6EJH24rNDPcfs'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'zXJ93yHzpVFxOk7PQf8TI0WaedqaftsG'

}

}

}, {

\_id: '56d95e3a999a71831bfdea64',

name: 'User4',

userid: 'userid4A3ZDS',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010364,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'oH2CJguUIe4FlOVtbnKnQghLPkf4ojE1'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'JNOtph4podh3DZjMRq5eT4iOaQEdIN8v'

}

}

}]

}, {

\_id: '56d95e3a999a71831bfdea66',

name: 'GROUP 2',

sortName: 'group 2',

description: 'GROUP 2 DESCRIPTION',

created: 1457086010482,

organizationId: '56d95e3a999a71831bfdea60',

\_\_v: 0,

users: ['56d95e3a999a71831bfdea61',

'56d95e3a999a71831bfdea62',

'56d95e3a999a71831bfdea63',

'56d95e3a999a71831bfdea64'

],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 67200,

mimeType: 'jpeg',

nameOnServer: 'gLZGiiZe6WG0U6fbqYgN4rQH4oVmO25w'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/jpeg',

nameOnServer: 'RPnndRW8tjWJhD7W766dt0k6fy1wDavb'

}

},

usersCount: 4,

userModels: [{

\_id: '56d95e3a999a71831bfdea61',

name: 'test',

userid: 'userid1OlegK',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010345,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'QLBDbupbTqPwPKQAXbq9gRop1OT7IHkc'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'DyFSaCRcPddPCQKJyAXnbCL6icqgkyIt'

}

}

}, {

\_id: '56d95e3a999a71831bfdea62',

name: 'User2',

userid: 'userid2On3Kv',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010355,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010780,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'NTKiNzll6pCsEZUqba3LTD5MFk8i7Uk2'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'ZKQgT2fJ6RPOSz34qKyX7FDPKiqJfeWo'

}

}

}, {

\_id: '56d95e3a999a71831bfdea63',

name: 'User3',

userid: 'userid3FUG8n',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010360,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010782,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'l7IXqnsOIMeivPZqy9k6EJH24rNDPcfs'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'zXJ93yHzpVFxOk7PQf8TI0WaedqaftsG'

}

}

}, {

\_id: '56d95e3a999a71831bfdea64',

name: 'User4',

userid: 'userid4A3ZDS',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010364,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'oH2CJguUIe4FlOVtbnKnQghLPkf4ojE1'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'JNOtph4podh3DZjMRq5eT4iOaQEdIN8v'

}

}

}]

}, {

\_id: '56d95e3a999a71831bfdea67',

name: 'GROUP 3',

sortName: 'group 3',

description: 'GROUP 3 DESCRIPTION',

created: 1457086010575,

organizationId: '56d95e3a999a71831bfdea60',

\_\_v: 0,

users: ['56d95e3a999a71831bfdea61',

'56d95e3a999a71831bfdea62',

'56d95e3a999a71831bfdea63',

'56d95e3a999a71831bfdea64'

],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 68800,

mimeType: 'jpeg',

nameOnServer: 'HaxQADVSLlp8Vqw1wcjP6CjBsCnagk9J'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'rDtRL9FauSCadmLJFOz4wox9MgaKOQGp'

}

},

usersCount: 4,

userModels: [{

\_id: '56d95e3a999a71831bfdea61',

name: 'test',

userid: 'userid1OlegK',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010345,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'QLBDbupbTqPwPKQAXbq9gRop1OT7IHkc'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'DyFSaCRcPddPCQKJyAXnbCL6icqgkyIt'

}

}

}, {

\_id: '56d95e3a999a71831bfdea62',

name: 'User2',

userid: 'userid2On3Kv',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010355,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010780,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'NTKiNzll6pCsEZUqba3LTD5MFk8i7Uk2'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'ZKQgT2fJ6RPOSz34qKyX7FDPKiqJfeWo'

}

}

}, {

\_id: '56d95e3a999a71831bfdea63',

name: 'User3',

userid: 'userid3FUG8n',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010360,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010782,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'l7IXqnsOIMeivPZqy9k6EJH24rNDPcfs'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'zXJ93yHzpVFxOk7PQf8TI0WaedqaftsG'

}

}

}, {

\_id: '56d95e3a999a71831bfdea64',

name: 'User4',

userid: 'userid4A3ZDS',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010364,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'oH2CJguUIe4FlOVtbnKnQghLPkf4ojE1'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'JNOtph4podh3DZjMRq5eT4iOaQEdIN8v'

}

}

}]

}, {

\_id: '56d95e3a999a71831bfdea68',

name: 'GROUP 4',

sortName: 'group 4',

description: 'GROUP 4 DESCRIPTION',

created: 1457086010666,

organizationId: '56d95e3a999a71831bfdea60',

\_\_v: 0,

users: ['56d95e3a999a71831bfdea61',

'56d95e3a999a71831bfdea62',

'56d95e3a999a71831bfdea63',

'56d95e3a999a71831bfdea64'

],

avatar: {

thumbnail: {

originalName: 'user4.png',

size: 52600,

mimeType: 'jpeg',

nameOnServer: 'djG4hAWoR22Butr3CuOHazf4fCyyQSEc'

},

picture: {

originalName: 'user4.png',

size: 41968,

mimeType: 'image/png',

nameOnServer: 'HZFKevSQcuJX61DBzUAEoQPHzsgTyakg'

}

},

usersCount: 4,

userModels: [{

\_id: '56d95e3a999a71831bfdea61',

name: 'test',

userid: 'userid1OlegK',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010345,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'QLBDbupbTqPwPKQAXbq9gRop1OT7IHkc'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'DyFSaCRcPddPCQKJyAXnbCL6icqgkyIt'

}

}

}, {

\_id: '56d95e3a999a71831bfdea62',

name: 'User2',

userid: 'userid2On3Kv',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010355,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010780,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'NTKiNzll6pCsEZUqba3LTD5MFk8i7Uk2'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'ZKQgT2fJ6RPOSz34qKyX7FDPKiqJfeWo'

}

}

}, {

\_id: '56d95e3a999a71831bfdea63',

name: 'User3',

userid: 'userid3FUG8n',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010360,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010782,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'l7IXqnsOIMeivPZqy9k6EJH24rNDPcfs'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'zXJ93yHzpVFxOk7PQf8TI0WaedqaftsG'

}

}

}, {

\_id: '56d95e3a999a71831bfdea64',

name: 'User4',

userid: 'userid4A3ZDS',

password: '\*\*\*\*\*',

organizationId: '56d95e3a999a71831bfdea60',

created: 1457086010364,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086010786,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'oH2CJguUIe4FlOVtbnKnQghLPkf4ojE1'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'JNOtph4podh3DZjMRq5eT4iOaQEdIN8v'

}

}

}]

}],

count: 4

}

}

# WebAPI - historylist for update

**0.0.0**

Returns user history which is updated from lastUpdate param

/api/v2/user/history/diff/:lastUpdate

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-History_List_for_update-0_0_0-0)

same as history list

# WebAPI - historylist

**0.0.0**

Returns user histoy

/api/v2/user/history/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-History_List-0_0_0-0)

{

"code": 1,

"time": 1458058376999,

"data": {

"list": [{

"\_id": "56e83466e804ad952eb49863",

"userId": "56e6b6d206552124125cdb86",

"chatId": "56dee3091f9b351bcd4e546a",

"chatType": 2,

"lastUpdate": 1458058343838,

"lastUpdateUser": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058317218,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"lastMessage": {

"type": 1,

"created": 1458058343831,

"message": "4"

},

"unreadCount": 4,

"\_\_v": 0,

"group": {

"\_id": "56dee3091f9b351bcd4e546a",

"name": "Developers",

"sortName": "developers",

"description": "",

"created": 1457447689335,

"organizationId": "56ab7b9061b760d9eb6feba3",

"parentId": "56dee2fd1f9b351bcd4e5469",

"type": 2,

"\_\_v": 0,

"users": ["56e6b6d706552124125cdb87", "56e6b71106552124125cdb8c", "56e6b71306552124125cdb8d", "56e6b71f06552124125cdb8e", "56e6b72806552124125cdb8f", "56e6b73206552124125cdb90", "56e6b73706552124125cdb91", "56e6b73d06552124125cdb92", "56e6b6d206552124125cdb86"],

"usersCount": 9,

"userModels": [{

"\_id": "56e6b6d206552124125cdb86",

"name": "user1",

"sortName": "user1",

"description": "user1",

"userid": "testuser1",

"password": "\*\*\*\*\*",

"created": 1457960658915,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058365344,

"token": "\*\*\*\*\*",

"groups": ["56e7da1671be6b38d3beb967", "56dee3091f9b351bcd4e546a"]

}, {

"\_id": "56e6b6d706552124125cdb87",

"name": "user2",

"sortName": "user2",

"description": "user1",

"userid": "testuser2",

"password": "\*\*\*\*\*",

"created": 1457960663961,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"groups": ["56dee3091f9b351bcd4e546a"]

}, {

"\_id": "56e6b71106552124125cdb8c",

"name": "user3",

"sortName": "user3",

"description": "",

"userid": "testuser3",

"password": "\*\*\*\*\*",

"created": 1457960721257,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"groups": ["56dee3091f9b351bcd4e546a"]

}, {

"\_id": "56e6b71306552124125cdb8d",

"name": "user4",

"sortName": "user4",

"description": "",

"userid": "testuser4",

"password": "\*\*\*\*\*",

"created": 1457960723968,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"groups": ["56dee3091f9b351bcd4e546a"]

}, {

"\_id": "56e6b71f06552124125cdb8e",

"name": "user6",

"sortName": "user6",

"description": "",

"userid": "testuser6",

"password": "\*\*\*\*\*",

"created": 1457960735226,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"groups": ["56dee3091f9b351bcd4e546a"]

}]

}

}, {

"\_id": "56e83451e804ad952eb49857",

"userId": "56e6b6d206552124125cdb86",

"chatId": "56e7da1671be6b38d3beb967",

"chatType": 2,

"lastUpdate": 1458058322044,

"lastUpdateUser": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058317218,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"lastMessage": {

"type": 1,

"created": 1458058322036,

"message": "d"

},

"unreadCount": 3,

"\_\_v": 0,

"group": {

"\_id": "56e7da1671be6b38d3beb967",

"name": "test",

"sortName": "test",

"description": "test",

"created": 1458035222590,

"organizationId": "56ab7b9061b760d9eb6feba3",

"type": 1,

"\_\_v": 0,

"users": ["56c32acd331dd81f8134f200", "56e6b6d206552124125cdb86"],

"usersCount": 2,

"userModels": [{

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058317218,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

}, {

"\_id": "56e6b6d206552124125cdb86",

"name": "user1",

"sortName": "user1",

"description": "user1",

"userid": "testuser1",

"password": "\*\*\*\*\*",

"created": 1457960658915,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058365344,

"token": "\*\*\*\*\*",

"groups": ["56e7da1671be6b38d3beb967", "56dee3091f9b351bcd4e546a"]

}]

}

}, {

"\_id": "56e83277fabab6dd2c0e1860",

"userId": "56e6b6d206552124125cdb86",

"chatId": "56c32acd331dd81f8134f200",

"chatType": 1,

"lastUpdate": 1458058176955,

"isUnread": 1,

"lastUpdateUser": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058165516,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

}

},

"lastMessage": {

"type": 1,

"created": 1458058176946,

"message": "ss"

},

"unreadCount": 7,

"\_\_v": 0,

"user": {

"\_id": "56c32acd331dd81f8134f200",

"name": "Ken",

"sortName": "ken yasue",

"description": "ああああ",

"userid": "kenyasue",

"password": "\*\*\*\*\*",

"created": 1455631053660,

"status": 1,

"organizationId": "56ab7b9061b760d9eb6feba3",

"\_\_v": 0,

"tokenGeneratedAt": 1458058317218,

"token": "\*\*\*\*\*",

"groups": ["56e7ef842500caf2eda5d5fb", "56e7da1671be6b38d3beb967", "56dee2fd1f9b351bcd4e5469"],

"avatar": {

"thumbnail": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "c2gQT5IMJAYqx89eo8gwFrKJSRxlFYFU"

},

"picture": {

"originalName": "2015-01-11 21.30.05 HDR.jpg",

"size": 1551587,

"mimeType": "image/png",

"nameOnServer": "jf7mBTsU6CVfFPLsnY4Ijqcuo6vYTKAs"

}

},

"onlineStatus": 1

}

}],

"count": 3,

"totalUnreadCount": 20

}

}

# WebAPI - just test

**0.0.0**

Returns text "test"

/api/v2/test

# WebAPI - mute / unmute

**0.0.0**

mute / unmute

/api/v2/user/mute

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| action | String | action |
| target | String | target |
| type | String | type |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-mute___unmute-0_0_0-0)

{}

# WebAPI - search group

**0.0.0**

Returns groups matched to keyword

/api/v2/group/search/:page?keyword=\*\*\*\*\*\*\*\*

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Seach_Group-0_0_0-0)

{

code: 1,

time: 1457086227601,

data: {

list: [{

\_id: '56d95f128da73e481e25b4c6',

name: 'GROUP 1',

sortName: 'group 1',

description: 'GROUP 1 DESCRIPTION',

created: 1457086226073,

organizationId: '56d95f128da73e481e25b4c1',

\_\_v: 0,

users: ['56d95f128da73e481e25b4c2',

'56d95f128da73e481e25b4c3',

'56d95f128da73e481e25b4c4',

'56d95f128da73e481e25b4c5'

],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 46100,

mimeType: 'jpeg',

nameOnServer: 'wvz4ezsEtgqTlQ6ec740gCKg48OB6df1'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/jpeg',

nameOnServer: 'hLWVa6YetMzzw8gNZoR2zHKnz1WYuyVq'

}

},

usersCount: 4,

userModels: [{

\_id: '56d95f128da73e481e25b4c2',

name: 'test',

userid: 'userid1RgQus',

password: '\*\*\*\*\*',

organizationId: '56d95f128da73e481e25b4c1',

created: 1457086226039,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086226476,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: 'Bd8PfEfryfFKm87GPPgK7vCFjavFsGRT'

},

picture: {

originalName: 'max.jpg',

size: 64914,

mimeType: 'image/png',

nameOnServer: '28PCuc4iljhbxuPHDYGXlBUZHU1wjP7y'

}

}

}, {

\_id: '56d95f128da73e481e25b4c3',

name: 'User2',

userid: 'userid2KHQyH',

password: '\*\*\*\*\*',

organizationId: '56d95f128da73e481e25b4c1',

created: 1457086226050,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086226471,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'YNuPww9hx1QZkzXrrxxn7OjTRSTrqmDk'

},

picture: {

originalName: 'user1.jpg',

size: 36023,

mimeType: 'image/png',

nameOnServer: 'aYdcKWAtOWBZbUWQMiaInmkz7esHuYpY'

}

}

}, {

\_id: '56d95f128da73e481e25b4c4',

name: 'User3',

userid: 'userid3NmVQ6',

password: '\*\*\*\*\*',

organizationId: '56d95f128da73e481e25b4c1',

created: 1457086226055,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086226472,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: 'uOkWAaLBQaITjgA7NVtUz0NC1Q5e9wOC'

},

picture: {

originalName: 'user2.jpg',

size: 53586,

mimeType: 'image/png',

nameOnServer: '3OYwlwUojGskBF6NXdb2LK1sWuEaaeht'

}

}

}, {

\_id: '56d95f128da73e481e25b4c5',

name: 'User4',

userid: 'userid4eZJ83',

password: '\*\*\*\*\*',

organizationId: '56d95f128da73e481e25b4c1',

created: 1457086226058,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1457086226476,

token: '\*\*\*\*\*',

description: null,

departments: [],

groups: [],

avatar: {

thumbnail: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'Jxl4Q0IkuGhPFk4AGLPzlyV19ACWlKyl'

},

picture: {

originalName: 'user3.png',

size: 54101,

mimeType: 'image/png',

nameOnServer: 'iMpc4pbIqnazrSXhucTn8gTCfE4WJp0u'

}

}

}]

}],

count: 1

}

}

# WebAPI - search user

**0.0.0**

Returns users matched to keyword

/api/v2/user/search/:page?keyword=\*\*\*\*\*\*\*\*

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-Seach_User-0_0_0-0)

{

code: 1,

time: 1456698684520,

data: {

list: [{

\_id: '56d3753c294ed9761afe489c',

name: 'user1',

userid: 'userid1ilmaN',

password: '\*\*\*\*\*',

organizationId: '56d3753b294ed9761afe489b',

created: 1456698684006,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1456698684452,

token: '\*\*\*\*\*',

departments: [],

groups: [],

onelineStatus: 0

}],

count: 1

}

}

# WebAPI - upload file

**0.0.0**

Returns fileId

/api/v2/file/upload

# WebAPI - userlist

**0.0.0**

Returns all users

/api/v2/user/list/:page

## Header

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| access-token | String | Users unique access-token. |

* [Success-Response:](http://document.spika.chat/docs/API/#success-examples-WebAPI-List_User-0_0_0-0)

{

code: 1,

time: 1456698548213,

data: {

list: [{

\_id: '56d374b30daaf268195201bb',

name: 'user1',

userid: 'userid1mW84z',

password: '\*\*\*\*\*',

organizationId: '56d374b30daaf268195201ba',

created: 1456698547727,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1456698548170,

token: '\*\*\*\*\*',

departments: [],

groups: [],

onelineStatus: 0

}, {

\_id: '56d374b30daaf268195201bc',

name: 'user2',

userid: 'userid2EqTJ7',

password: '\*\*\*\*\*',

organizationId: '56d374b30daaf268195201ba',

created: 1456698547737,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1456698548166,

token: '\*\*\*\*\*',

departments: [],

groups: [],

onelineStatus: 0

}, {

\_id: '56d374b30daaf268195201bd',

name: 'user3',

userid: 'userid3V2S36',

password: '\*\*\*\*\*',

organizationId: '56d374b30daaf268195201ba',

created: 1456698547743,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1456698548167,

token: '\*\*\*\*\*',

departments: [],

groups: [],

onelineStatus: 0

}, {

\_id: '56d374b30daaf268195201be',

name: 'user4',

userid: 'userid4qXXvb',

password: '\*\*\*\*\*',

organizationId: '56d374b30daaf268195201ba',

created: 1456698547747,

status: 1,

\_\_v: 0,

tokenGeneratedAt: 1456698548170,

token: '\*\*\*\*\*',

departments: [],

groups: [],

onelineStatus: 0

}],

count: 4

}

}

# Spika Enterprise

## **Spika Enterprise API's reference**

**0.1.0**

# Socket

# Socket - Delete Message

**0.0.0**

Delete Message

"deleteMessage"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userID | string | User ID |
| messageID | string | Message ID |

# Socket - Deliver Message

**0.0.0**

Get delivered message

"deliverMessage"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userID | string | User ID |
| messageID | string | Message ID |

# Socket - Disconnect

**0.0.0**

Disconnect from socket

"disconnect"

# Socket - Login to socket

**0.0.0**

Login to socket

"login"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| token | string | user token |
| processId | string | identifier for device |

# Socket - Send New Message

**0.0.0**

Send new message by socket

"sendMessage"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| roomID | string | Room ID |
| userID | string | User ID |
| type | string | Message Type. 1:Text 2:File 3:Location |
| message | string | Message if type == 1 |
| fileID | string | File ID if type == 2 |
| location | object | lat and lng if type == 3 |

# Socket - Send typing notification

**0.0.0**

Send typing notification

"sendTyping"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userID | string | User ID |
| roomID | string | Room ID |
| type | string | 0: Remove typing notificaiton 1: Show typing notification |

# Socket - Update Message

**0.0.0**

Delete Message

"updateMessage"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userID | string | User ID |
| messageID | string | Message ID |
| message | string | Mesage |

# Socket - answer to call request

**0.0.0**

answer to call request

"call\_answer"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | user id |

# Socket - cancel call request

**0.0.0**

cancel call request

"call\_cancel"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | user id |

# Socket - finish call

**0.0.0**

finish call

"call\_close"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | user id |

# Socket - keepalive

**0.0.0**

set user online

"keepalive"

# Socket - make call request

**0.0.0**

Make call request

"call\_request"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | user id |
| mediaType | string | 1: audio 2: video |

# Socket - open unread message

**0.0.0**

set user online

"openMessage"

# Socket - ping ok

**0.0.0**

answer for ping request

"pingok"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | userId |

# Socket - reject call request

**0.0.0**

reject call request

"call\_reject"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | user id |
| rejectType | string | 2: user busy 3: user declined |

# Socket - send call\_received to caller

**0.0.0**

send call\_received to caller

"call\_received"

## 参数

| **字段** | **类型** | **描述** |
| --- | --- | --- |
| userId | string | user id |

Frontend

# Build instructions

Here is step by step instruction to setup development environment of frontend. Before doing this all our clients have to buy license and we provice access permission to our github private repository.

#### **Clone from repository**

**You will have your own repository. So please change this url part.**

$ git clone https://github.com/cloverstudio/SFB\_Server.git

#### **Install libraries**

$ cd SFB\_Serve

$ npm install

#### **Edit config file**

$ cp src/client/js/lib/init-sample.js src/client2/js/lib/config.js

$ vim src/client/js/lib/config.js

**In production you will need only change these configurations.**

hashSalt: "",

AESPassword: ""

#### **Debug frontend**

$ npm start

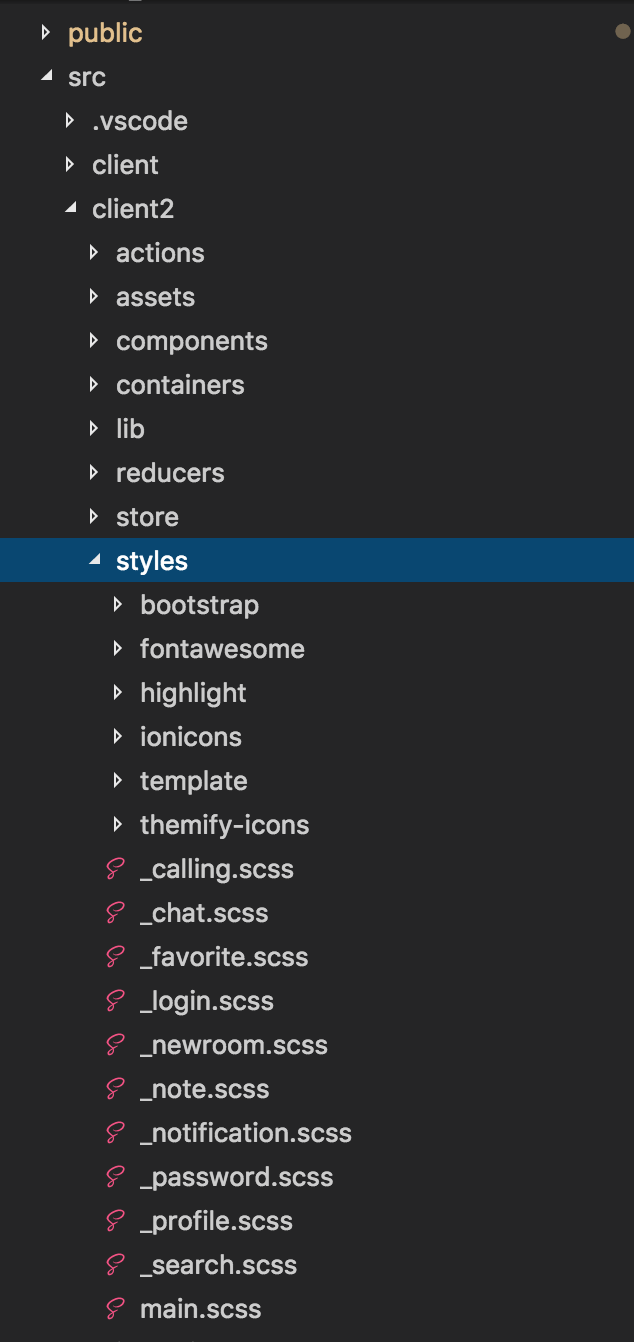
Then you can debug frontend by opening http://localhost:3000

#### **Build source code**

$ npm run build

# Change designs

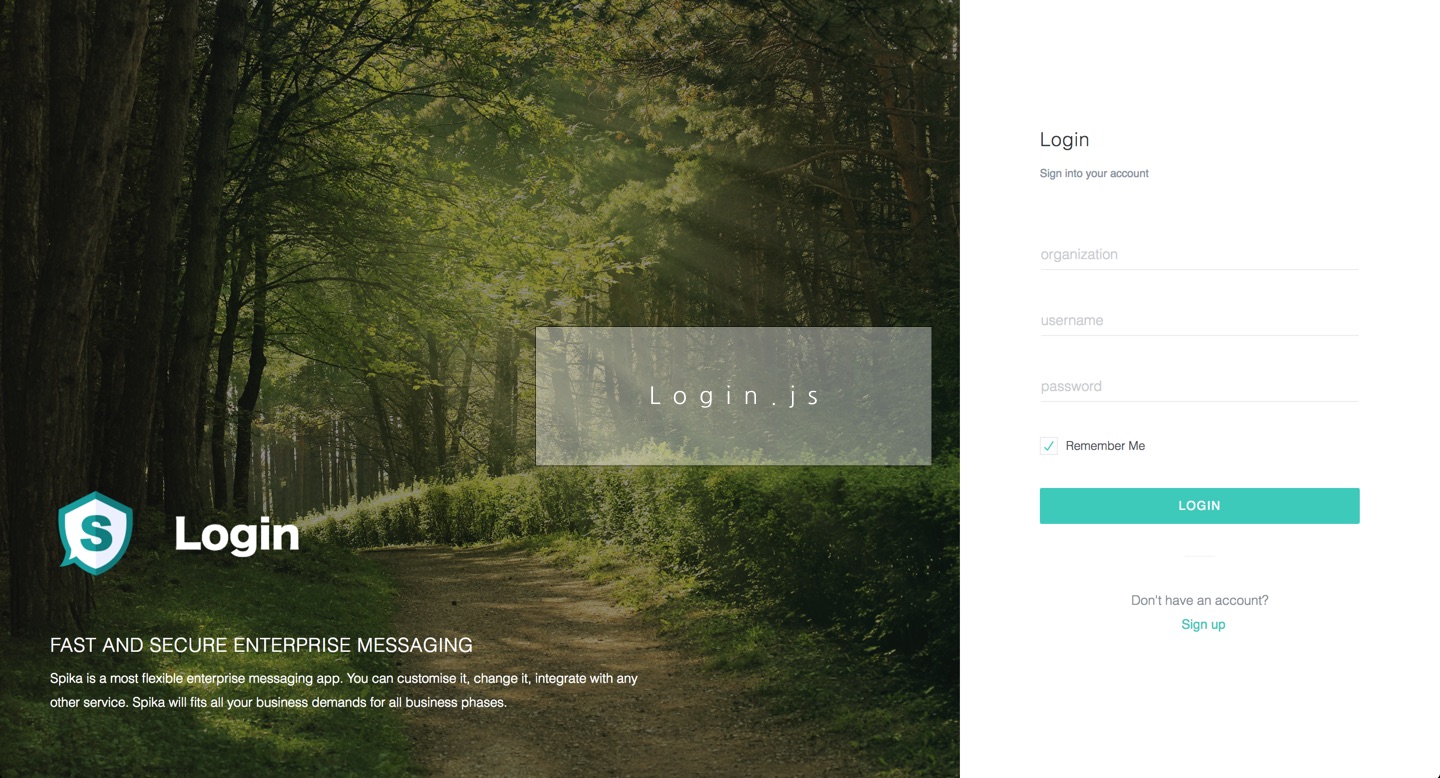
## **Update style sheet**

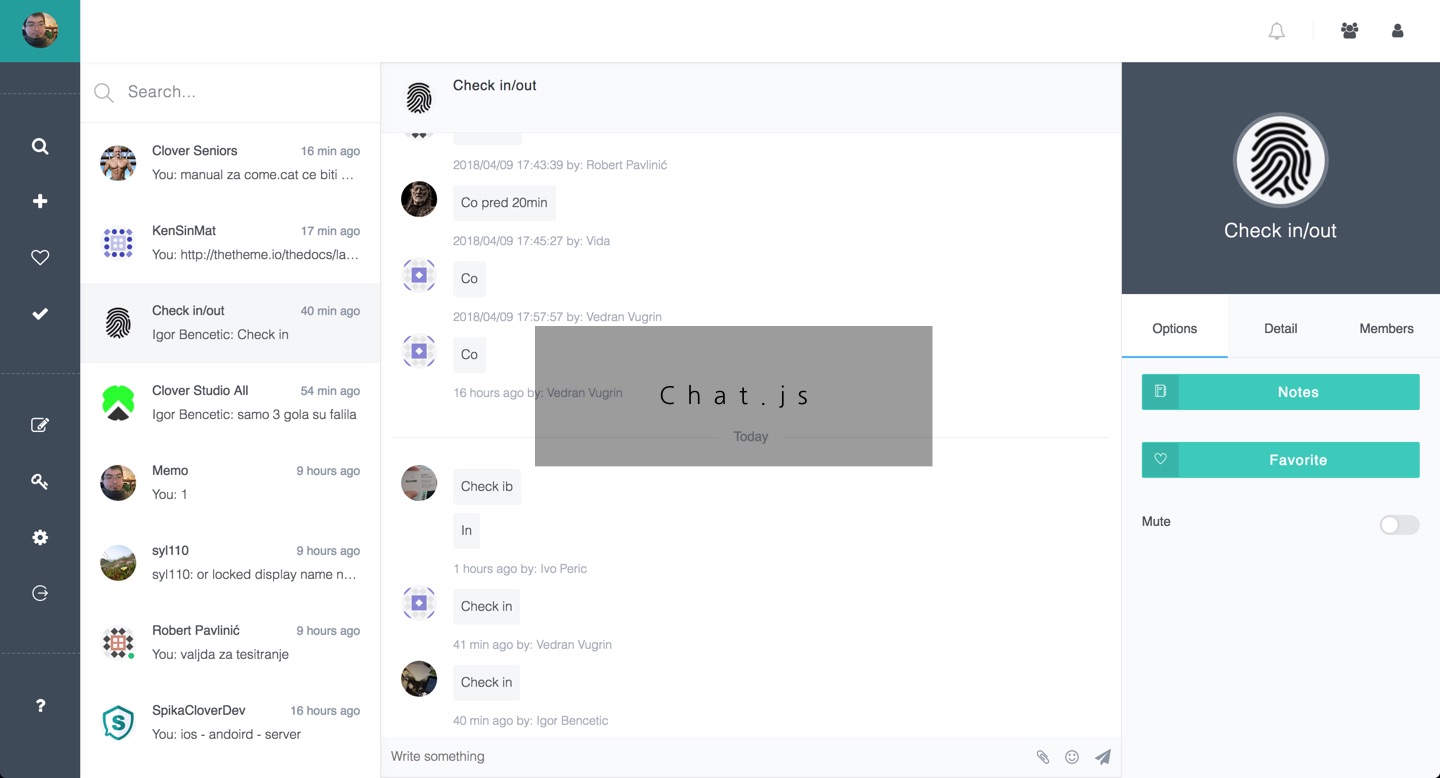
Spika uses SCSS to define styles. All style files are under src/client2/styles.

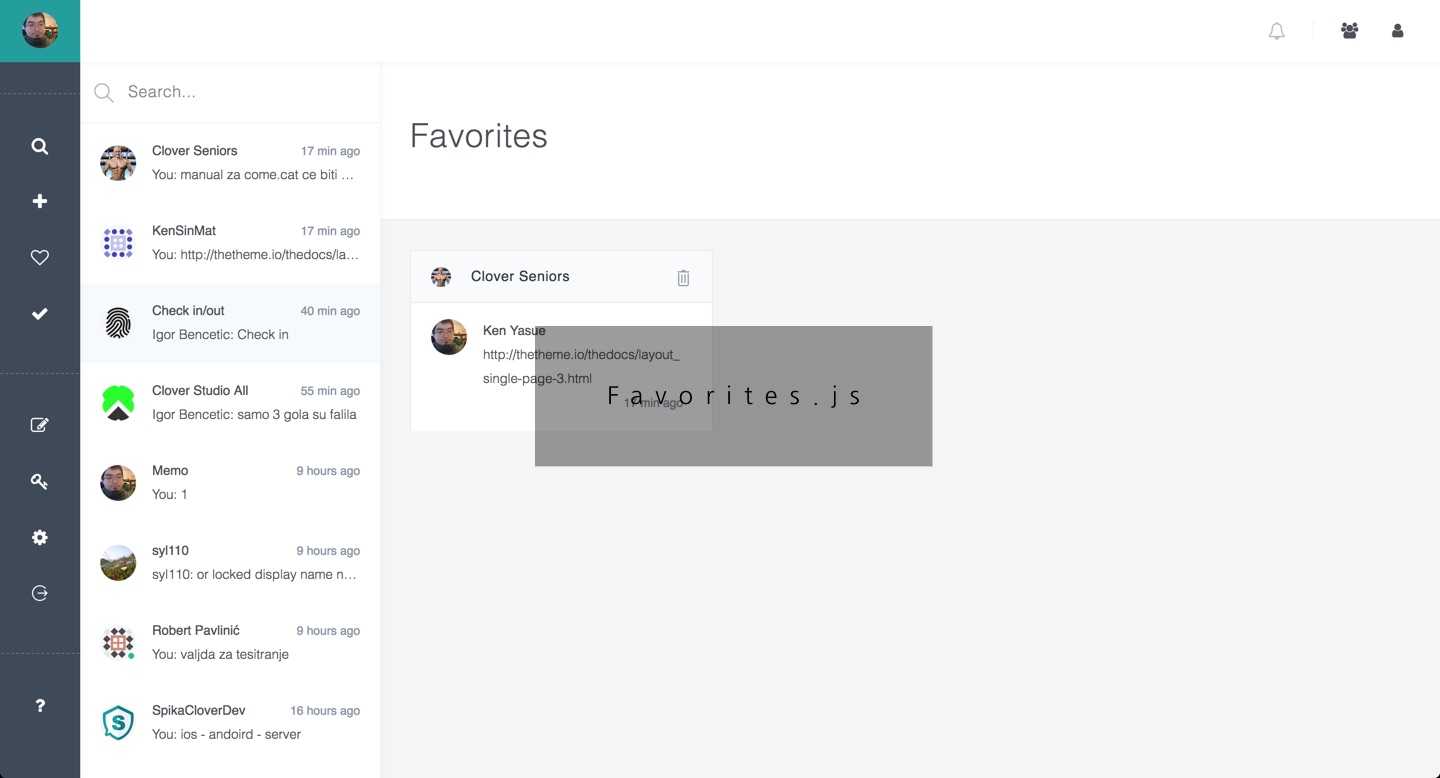
After changing .scss please execute following command to compile scss.

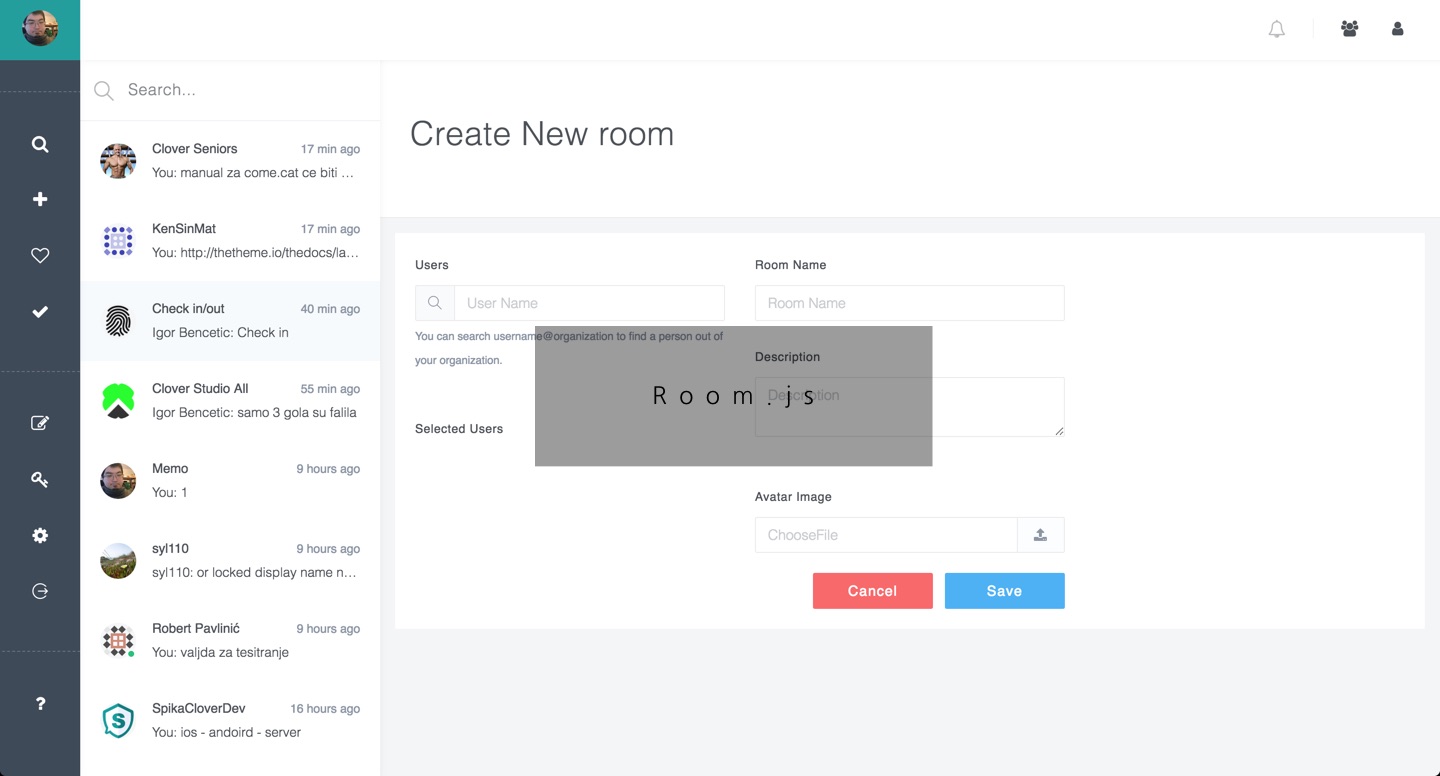
$ npm run build

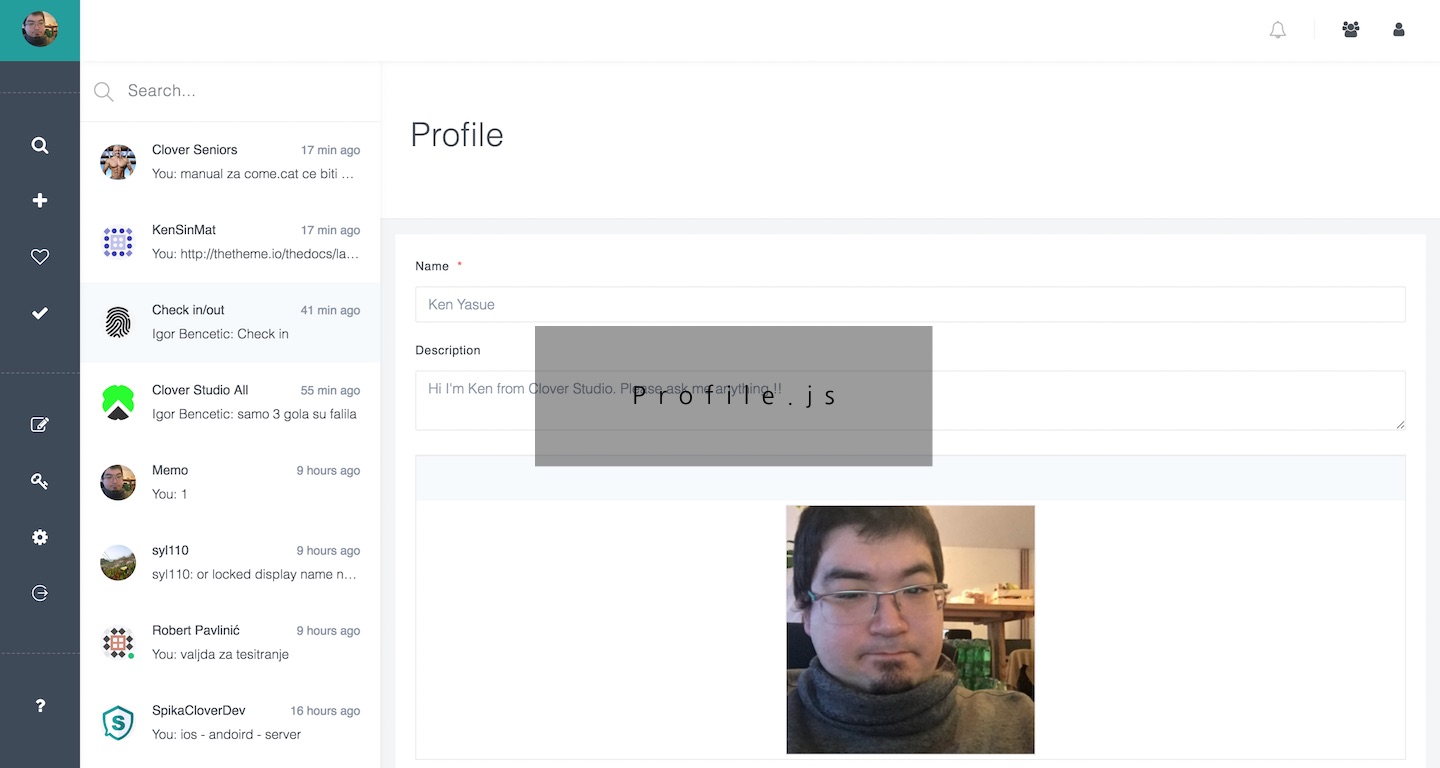
# Screens

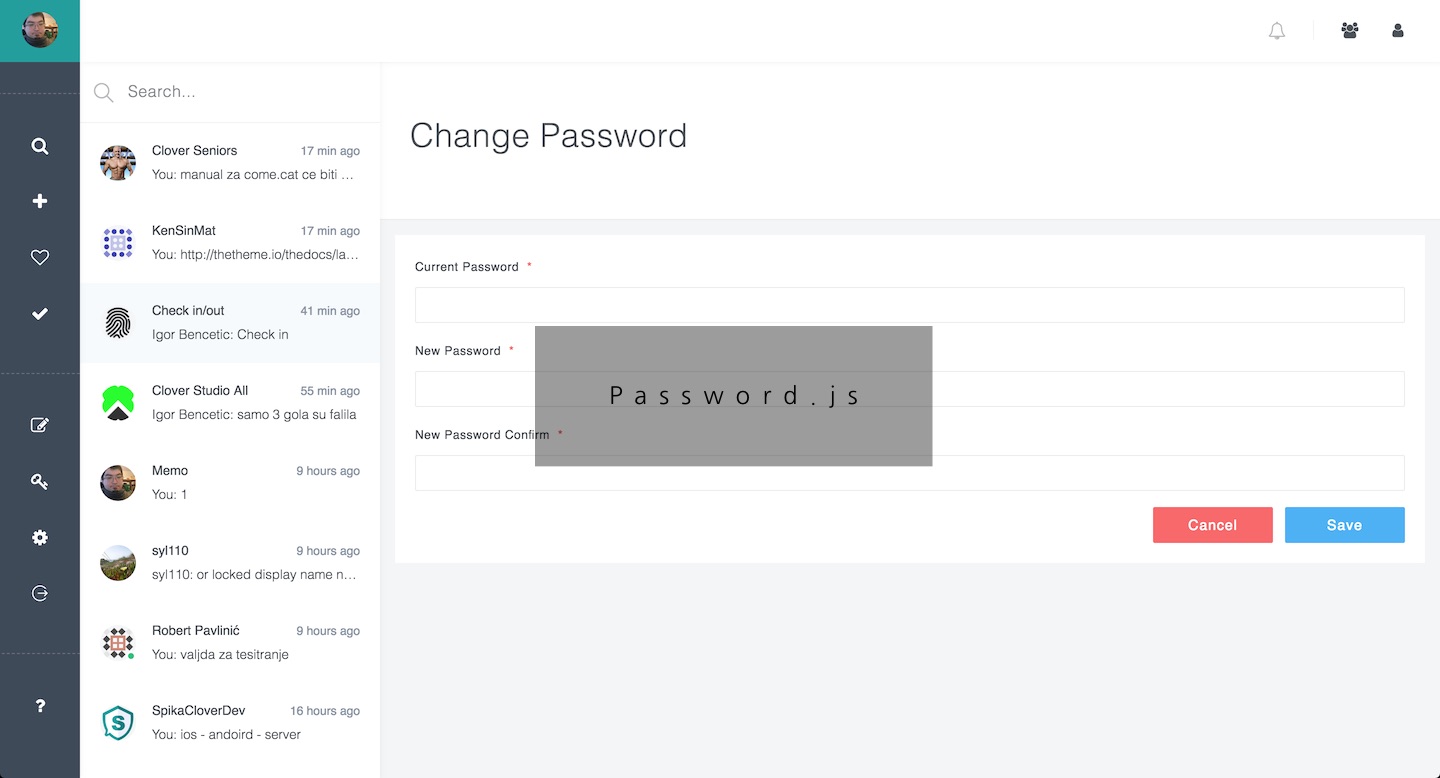


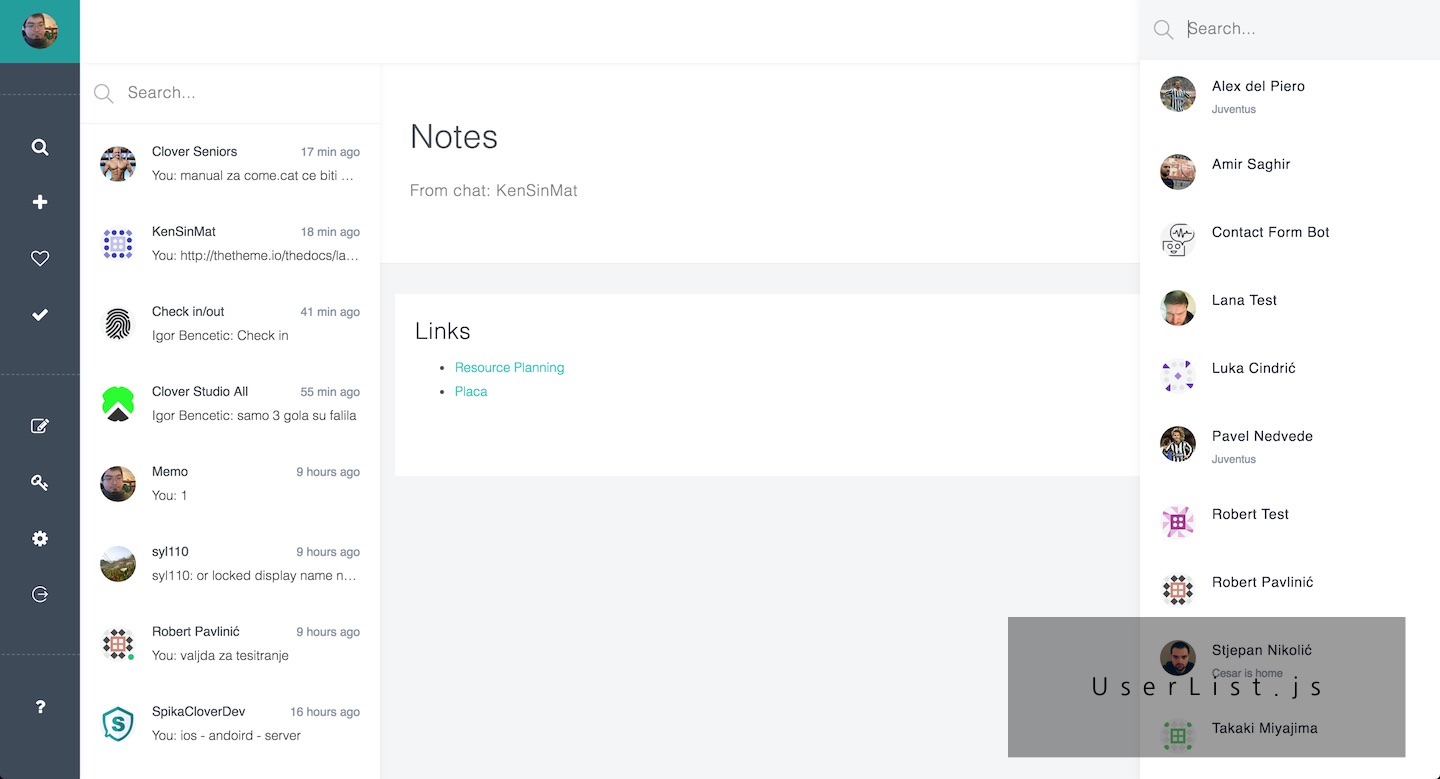


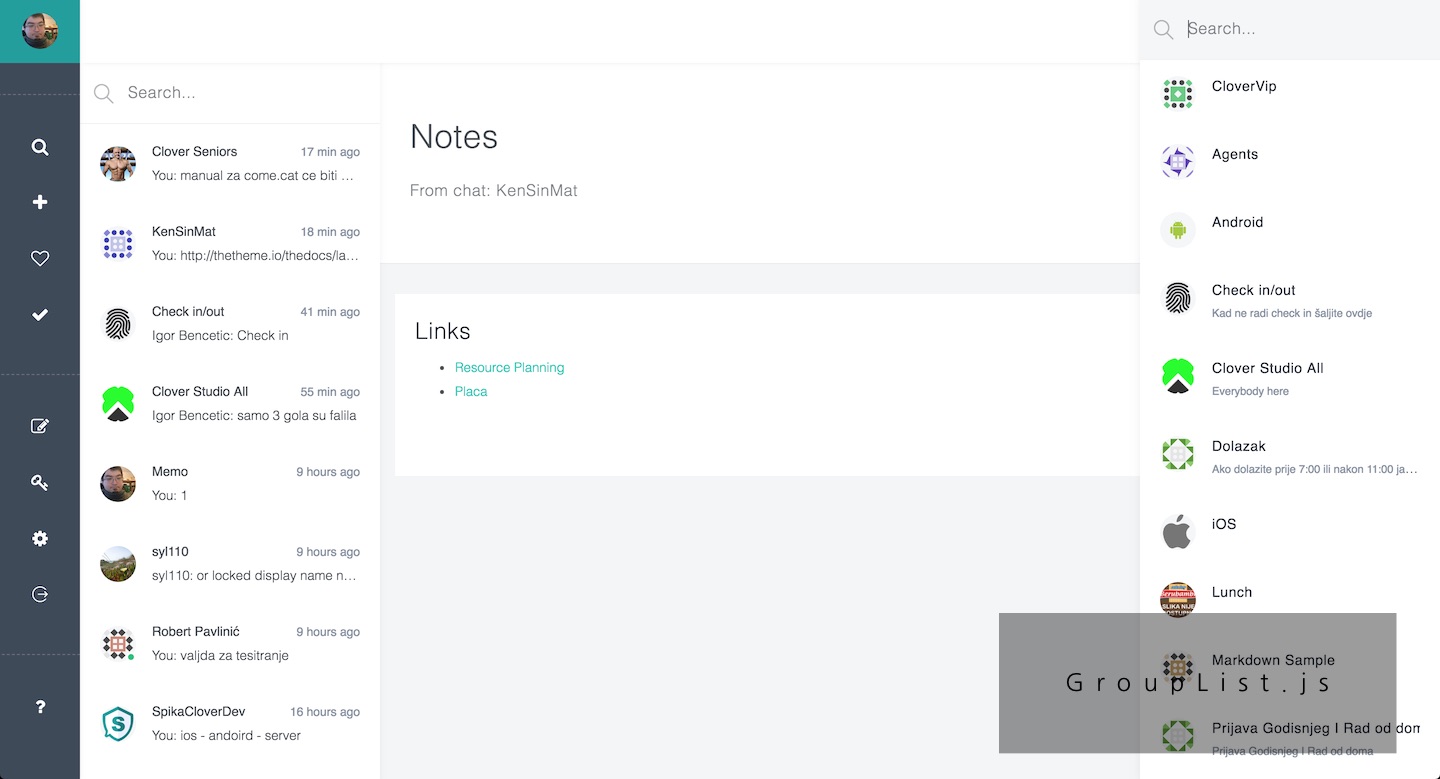












# How to login to Backend

All API call needs "access-token" parameter in request header. Backend system detect which user is sending the request by the parameter. So "login to backend" means getting the new accesss token for calling API.

[Here is API spec](http://document.spika.chat/docs/API/#api-WebAPI-Signin) for signin API.

* organizationid : Organization name to login.
* userid : Username to login.
* password : Hashed password of the user.
* pushToken : Pushtoken if it is iOS or Android app.
* secret : Secret is explained in following part.
* UUID : UUID is created in client side.

### **How to hash password.**

sha1(plainTextPassword + Config.hashSalt);

### **How to generate secret.**

First you have to call "/test" API to get server's time stamp. Here is response of Test API.

{"code":1,"time":1499866799001,"data":"test"}

Then secret is generate in this way.

var tenSec = Math.floor(time / 1000 / 10) + '';

var key = Config.hashSalt + tenSec;

var secret = sha1(key);

return secret;

With this secret Spika secure from data sniffing becasse the secret is valid only 10 second.

After calling /signin API, it returns following response.

{

code: 1,

time: 1454417582385,

data: {

newToken: 'DOqsolWe6zt3EFn0',

user: {

\_id: '56b0a6ae6753ea416ad58ea9',

name: 'user1',

userid: 'userid1ixfQJ',

password: '\*\*\*\*\*',

created: 1454417582354,

\_\_v: 0,

token: '\*\*\*\*\*',

tokenGeneratedAt: 1454417582384

},

organization: {

\_id: '56b0a6ae6753ea416ad58ea8',

organizationId: 'clover',

name: 'user1',

created: 1454417582336,

status: 1,

\_\_v: 0

}

}

}

Client app has to save "newToken" parameter for calling all other API.

# Get Message List

Here is how to get list of messages for certain chat.

Call [/message/list](http://document.spika.chat/docs/API/#api-WebAPI) API.

* roomId : RoomID is unique identifier for chat.
* lastMessageId : Using for paging. Send 0 to read from first.
* direction : To get older message send "old", to get newer massge send "new"

Here is sample response of /message/list API.

{

"code": 1,

"time": 1499868072495,

"data": {

"messages": [{

"\_\_v": 0,

"\_id": "594abd2a17a3077a5c0381dc",

"created": 1498070314307,

"localID": "culkpvtyEs1G6OUqQ2RkkjZXKBjZCFC7",

"message": "0301dc83cfffa5a368ade6872e397f7595c7cee27261a8cd0282c16caf0a41b3e86555ce92acf0c7ef00796fb6ff0540cf4df36a7e270cc840e84fbc3f293276bc243894f6e31a0ad81af10d9339fd1615aada49b4476f99c967a472437ffaa31636dafa8643eee8853b8ed7305111756625",

"remoteIpAddress": "109.60.110.82",

"roomID": "3-58fdd63c5b9736991af8ca13",

"type": 1,

"user": {

"\_id": "56e0088062a63ebf55eef39a",

"description": "Samo jakooooo!!!",

"name": "Jurica Blazevic",

"organizationId": "56e005b1695213295419f5df",

"userid": "jurica.blazevic",

"avatar": {

"thumbnail": {

"originalName": "417020\_10150655766113706\_100243839\_n.jpg",

"size": 28617,

"mimeType": "image/png",

"nameOnServer": "Nk51aWGNBmpP5im8x3ecJwAKizI7ANJe"

},

"picture": {

"originalName": "417020\_10150655766113706\_100243839\_n.jpg",

"size": 28617,

"mimeType": "image/png",

"nameOnServer": "1QKB80SOm3waYooa3enPxPBIbiENj5Y6"

}

}

},

"userID": "56e0088062a63ebf55eef39a",

"seenBy": [{

"version": 2,

"at": 1498623027378,

"user": "57f315081d9cabd56e905080"

}..

],

"isFavorite": 0

}, ...

}]

}

}

### **roomId parameter**

roomId is unique parameter for each chat.

1. Private chat : 1-"userid"-"userid" First userid is who has lower "created" timestamp.
2. Group chat : 2-"groupid"
3. Room chat : 3-"roomid"

In Spika group is generated by organization admin, room can create all users.

Send Message

Spika uses WebSocket for following tasks.

* Send message
* Typing signal
* Calling
* Delete message
* Send seen signal

Here is sample code to send message.

socketIOManager.emit('sendMessage',{

message: filteredMessage,

roomID: this.currentRoomId,

userID: loginUserManager.user.\_id,

type: Const.messageTypeText,

localID: tempID,

attributes:{

useclient: "web"

},

user: loginUserManager.user

});

# Using SDK Part.1

We released JS SDK so our clients use Spika from their service easily. Here is simple tutorial how to use the SDK.

You can play API's here with our demo server.

<https://spika.chat/js/sdk/>

### **Generate API Key**

First of all, admin of organization must generate API Key from admin console. Please login to admin console and select "API Keys" from side menu.

### **Load**

<script type="text/javascript" src="[Your URL]/js/sdk/spika.js"></script>

### **Init**

SpikaSDK.init(baseURL,apiKey);

* BaseURL is endpoint of your Spika installation. Most of cases it will pe **http(s)://[your server]/api/v3**
* apiKey is the key you generated in Admin Console.
* This method doesn't call any API, just set these values to properties. So when the value is wrong you will get error when you call apu.

### **Signin**

SpikaSDK.signin(organization,username,password,cb)

callBack(statusCode,body);

CallBack is called when API returns response. When statusCode is 200, body is object.

### **SendMessage**

SpikaSDK.sendMessage(targetType,target,messageType,message,file,cb)

callBack(statusCode,body);

* Target type is following. 1: User 2: Group 3:Room
* MessageType is following. 1: Text 2: file
* If messageType is 1 (TextMessage) file argument is ignored.
* If messageType is 2 (File) message argument is ignored.

### **UploadFile**

SpikaSDK.uploadFile(file,cb)

callBack(statusCode,body);

Here is sample code to send file.

var file = $('#st4File')[0].files[0];

if(!file){

alert('Please select a file.');

return;

}

SpikaSDK.uploadFile(file,

function(statusCode,body){

if(body.file.fileId){

$('#st5File').val(body.file.fileId);

}

$('#result4 pre').text('Status : ' + statusCode + "\n" + "result : " + JSON.stringify(body, null, 4));

$('#result4').show();

});

### **Download File**

SpikaSDK.downloadFile(fileID,cb)

callBack(statusCode,blob);

### **Get Messages**

SpikaSDK.messageList(roomID,lastMessageID,direction,cb)

callBack(statusCode,body);

* roomID is unique identifier for any chat.
* lastMessageID is used for paging, if you want read from first positon send "0".
* Direction can be "old" or "new".

# Build instructions

Clone source code from github.

$ git clone https://github.com/cloverstudio/SFB\_iOS.git

Navigate to downloaded folder and open workspace file **ios-v2-spika-enterprise.xcworkspace** to include all 3th party libraries via cocoapods.

For full functionality you will have to setup provisioning profile with push and voip push services.

# Customisation - colors, urls...

### **Change base url**

Base url is located in **SFBCustomConfig** singleton class as a NSString constant **default\_url**. Once you log in, base url is saved to shared preferances as you can change base url from app on log in screen. So changing base url in **SFBCustomConfig** may not be enough to change it. You will need to make clean istall of applicaton to change it, or change it in app interface as well.

### **Colors**

All colors are defined as macros in **SE2Config.h** file For example there is primary color for app which is defined as:

#define kColorPrimary(\_\_ALPHA\_\_) [UIColor colorWithRed:19/255. green:159/255. blue:158/255. alpha:\_\_ALPHA\_\_]

There is also alpha parameter for transparency, so for solid primary color you would use it in code as:

UIColor \*color = kColorPrimary(1);

Primary color is also used for chat bubbles for my messages, to set color for other chat bubble messages use: **kAppBubbleLeftColor**

There are colors for text defined as: **kSE2ColorGrayText** **kAppMessageFontColor**

Also there are colors for buttons and alerts, defined as blue and red color as: **kSE2ColorBlue** **kSE2ColorRed**

# Login procedure

Login procedure is implemented in **SFBLoginApi** class.

+(void) login:(NSString \*)username passwordSHA:(NSString \*)sha1String organization:(NSString\*)organization responseBlock:(SFBAPIResponseBlock)responseBlock;

Parameters needed for login are **username**, **password** and **organization**.  
Password is encryped with sha-1 before sending. There are two steps for log in procedure.  
First we get timestamp from server and generate secret key from it with sha-1. It is used to prevent data sniffing as this secret is valid only for limited time.  
Then we are calling real login api, with all parameters: **username**, **password sha-1**, **secret sha-1**, and **organisation**.

After login procedure is finished we get result from server in response block. Most important parameter in response is token which is called **newToken**. We save it in shared preferances and use it in all other api calls to authenticate user. In all api calls it will be sent in header as **access-token** parameter.

Also after we are logged in, we connect to socket for real time messaging with **SFBSocketController**singleton class. After we are connected we emit login message with **token** received earlier, **UUID** which is uniqe to current device and **processId** which is randomly generated string.

Get messages

For getting messages with api from server we use method defined in **SFBChatApi** class.

+(void)getMessageApi:(NSString \*)roomdId lastMessageId:(NSString \*)lastMessageId direction:(NSString\*) direction responseBlock:(SFBAPIResponseBlock)responseBlock;

Parameters needed are:

* roomId - unique identifier for chat
* lastMessageId - message id (can be zero for fetching last 50 messages when using old)
* direction - constant for defining direction for fetching messages in relation to message id.
  + old - gets 50 messages older than the message with messageId
  + new - gets messages newer than the message with messageId
  + allto - gets all messages newer than the message with messageId, including that message

When in application messages are received in real time via socket.  
**SFBSocketController** is class that handles all socket communication. New messages are received as **newmessage** message and payload contains message model.

# Send message

Sending messages is done via socket which is defined in **SFBSocketController** class.  
We are emiting **sendMessage** message and as a payload we send message parameters which are different than our standard message model.

[[SFBSocketController sharedController] emit:kAppSocketSendMessage args:array];

Message parameters for socket communication are generated in **SFBEmitJsonCreator** class in createEmitSendMessage method.

+(NSDictionary\*) createEmitSendMessage:(SFBMessageModel\*) message andUser: (CSUserModel\*) user andMessage:(NSString\*) messageTxt andFile:(SE2FileWrapperModel \*)file andLocation:(SE2LocationModel\*) locationModel;

If socket is currently unavailable message will be sent with rest api defined in **SFBSendMessageAPI** class.

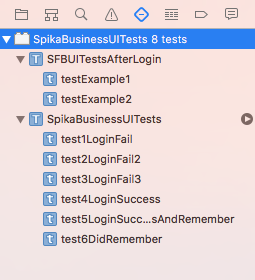
+(void)sendMessage:(NSString \*)message toRoom:(NSString \*)roomID responseBlock:(SFBAPIResponseBlock)responseBlock;

+(void)sendMessage:(NSString \*)message withType:(int) type withLocation:(SE2LocationModel\*) location withFileModel:(SE2FileWrapperModel\*) file andLocalId:(NSString\*) localId toRoom:(NSString \*)roomID responseBlock:(SFBAPIResponseBlock)responseBlock;

# UITests

UI tests are located in SpikaBusinessUITests folder.  
It consists from two scenarios. Logging in procedure and sending messages.  
**SpikaBusinessUITests** class is checking if the output is correct for logging in with both correct and incorrect data.  
**SFBUITestsAfterLogin** class is for testing basic funcionallity in app while you are logged in.

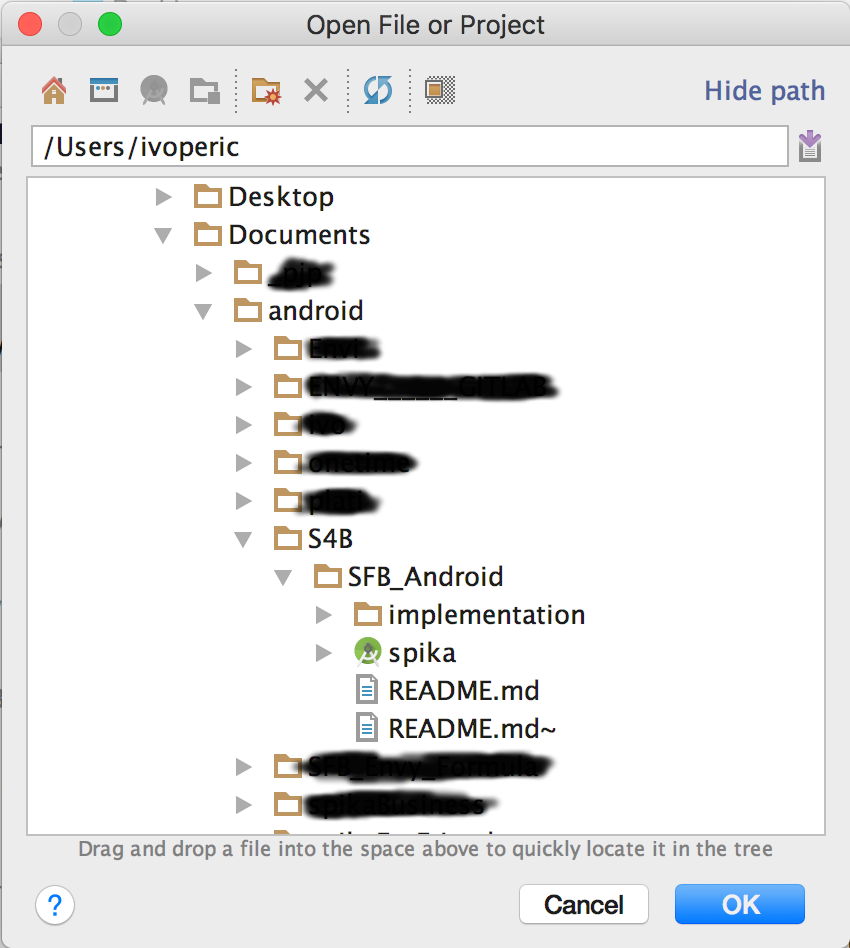
To run test simply select testing tab in project winow and run desired test or group of tests.



Build instructions

After download or clone source code from git, open android studio and navigate to S4B folder and open project. It is recomended to have latest version of android studio, sdk and gradle. After every android studio, gradle or sdk update we will change code to work on latest version. We always use latest android technologies.

* Version of sdk used in current application: 24
* Version of build tools in current application: 25.0.0
* Version of gradle build in current application: com.android.tools.build:gradle:2.3.3

It isn't recommend to use lower version then version above.  
Image 1 shows how to open S4B project.  
After that S4B is ready, just run project.  
For customisation (eg. change url or base color), open [this link](http://document.spika.chat/android/customisation-colors-urls).

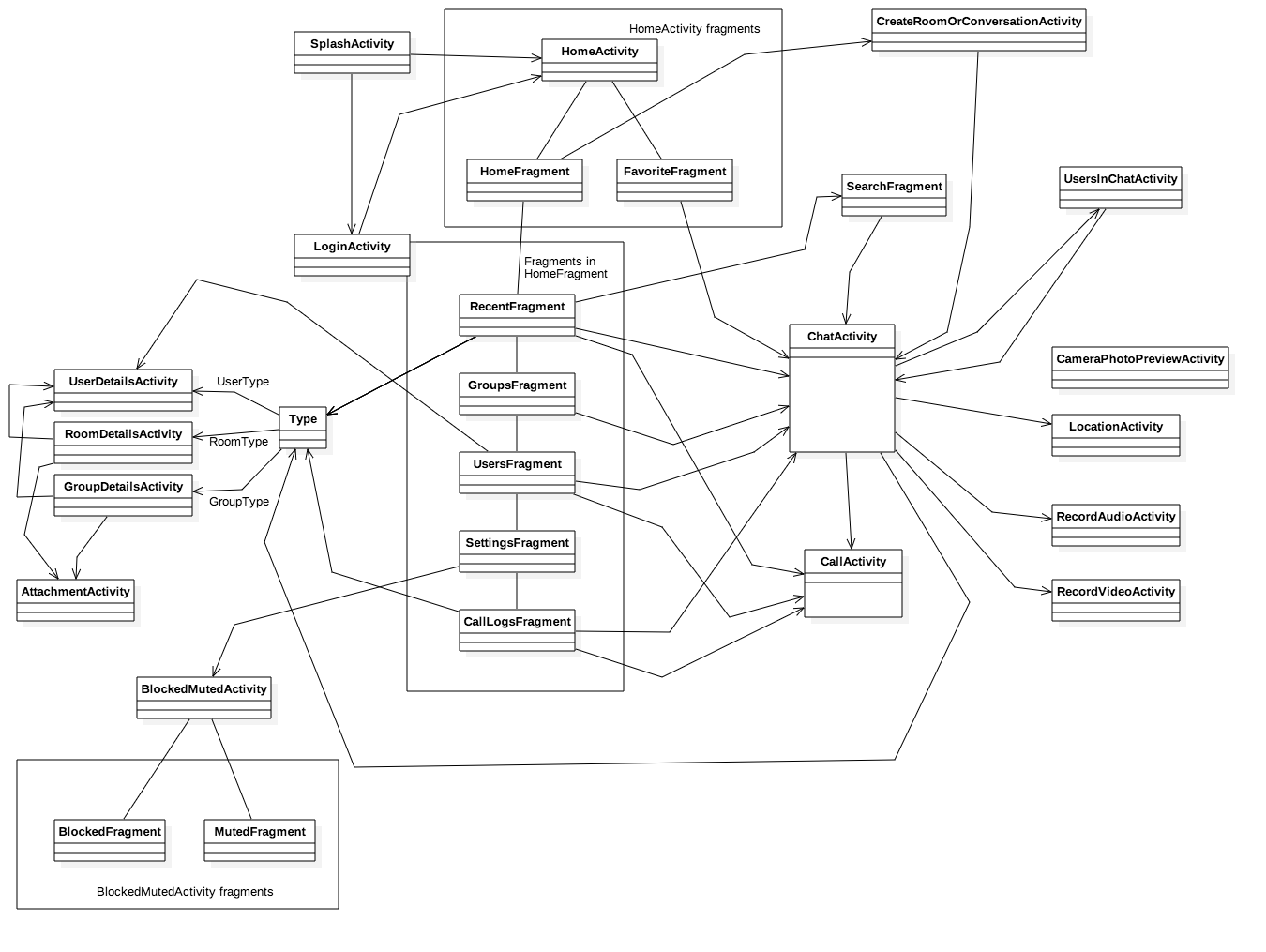
* [1. General about Spika](http://document.spika.chat/general)
* [2. Backend](http://document.spika.chat/backend)
* [3. Frontend](http://document.spika.chat/frontend)
* [4. iOS](http://document.spika.chat/ios)
* [5. Android](http://document.spika.chat/android)
  + [Build instructions](http://document.spika.chat/android/build-instructions)
  + [Screen diagram](http://document.spika.chat/android/screen-diagram)
  + [Customisation - colors, urls,...](http://document.spika.chat/android/customisation-colors-urls)
  + [Login procedure ( Example )](http://document.spika.chat/android/login-procedure-example)
  + [Get messages (Example)](http://document.spika.chat/android/get-messages-example)
  + [Send message ( Socket )](http://document.spika.chat/android/send-message-socket)
  + [JavaDoc](http://document.spika.chat/android/javadoc)
  + [UITests](http://document.spika.chat/android/uitests)
  + [Android SDK](http://document.spika.chat/android/android-sdk)
    - [Part 1 - import SDK](http://document.spika.chat/android/android-sdk/part-1-import-sdk)
    - [Part 2 - use SDK](http://document.spika.chat/android/android-sdk/part-2-use-sdk)

[Clear History](http://document.spika.chat/android/screen-diagram)

Built with [Grav](http://getgrav.org/) - The Modern Flat File CMS

[edit this page](https://github.com/getgrav/grav-skeleton-rtfm-site/blob/develop/pages/06.Android/02.screen-diagram/docs.md)

Screen diagram



# Customisation - colors, urls,...

### **Change base url of application**

For changing base url of application, we need to open Const class. Const class is in Utils package (folder). Just set BASE\_URL to new server url (withou / on end).

public static final String BASE\_URL = "";

Do not change BASE\_URL in Server class.

### **Change base color of application**

Open colors.xml file. It is in res folder. Primary color of application is

<color name="default\_color">#009E9F</color>

and change that variable into new base color. Secundary base color is

<color name="dark\_default\_color">#019191</color>

This value should be similar to primary color, just some darker or lighter (use it for click state or something similar). These three colors

<color name="best\_gray">#666666</color>

<color name="between\_best\_grays">#707070</color>

<color name="second\_best\_gray">#999999</color>

are using for default text color. Best gray is default, second best gray is secondary (lighter then default) and between best grasy is some shade of gray between best and second best gray color.  
These four colors below are related with buble background and text color in chat activity:

<color name="my\_message\_background\_color">@color/default\_color</color>

<color name="you\_message\_background\_color">#EBEBEB</color>

<color name="my\_message\_text\_color">#FFFFFF</color>

<color name="you\_message\_text\_color">#4C4C4C</color>

# Login procedure ( Example )

For network connection S4B uses Retrofit. More information about Retrofit can be found at [this link](http://square.github.io/retrofit/).  
All Retrofit files using for connection are in api.retofit and api.retrofit\_file packages. There are two listeners. One is with auto relogin, and other is without auto reglogin. With auto relogin listener, application will try to relogin user when user token is invalid or expired. Next important file for login api is LoginRetroApiInterface. That is Retrofit interface with methods related to login procedure. There are two methods in login interface

@GET(Const.Server.TEST\_API)

Call<TestApiModel> testApi(@Header(Const.HeadersParams.UUID) String uuid);

@POST(Const.Server.SINGIN\_API)

Call<SignInDataModel> signIn(@Body PostSignInModel signIn, @Header(Const.HeadersParams.UUID) String uuid);

First api is for getting exact server time. It is GET api which return JSON, which retrofit with GSON transform to TestApiModel. Exact server time is used for crypting secret.

LoginRetroApiInterface retroApiInterface = getRetrofit().create(LoginRetroApiInterface.class);

Call<TestApiModel> call = retroApiInterface.testApi(UserSingleton.getUUID(getActivity()));

call.enqueue(new CustomResponseNoNewToken<TestApiModel>(getActivity(), true, true){

@Override

public void onCustomSuccess(Call<TestApiModel> call, Response<TestApiModel> response) {

super.onCustomSuccess(call, response);

TestApiModel temp = response.body();

long time = temp.time;

time = time / 10000;

String timePlusSalt = Const.Secrets.STATIC\_SALT + time;

try {

String sha1Secret = SecretGeneratorUtils.SHA1(timePlusSalt);

loginToServer(sha1Password, sha1Secret);

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

}

}

@Override

public void onTryAgain(Call<TestApiModel> call, Response<TestApiModel> response) {

super.onTryAgain(call, response);

showProgress();

getTimeForSecret(sha1Password);

}

});

After getting exact server time with api, application generated secret key, which is need to login api. S4B send data to server with JSON encoding. To use that encoding, post model with sending data needs to be created. Than that model is passed to Retrofit, and Retrofit with GSON generate json and send it to server.

PostSignInModel postModel = new PostSignInModel(etUsername.getText().toString(), sha1Password, sha1Secret, SpikaApp.getEnterpriseSharedPreferences().getPushToken(), etOrganization.getText().toString(), UserSingleton.getUUID(getActivity()));

LoginRetroApiInterface retroApiInterface = getRetrofit().create(LoginRetroApiInterface.class);

Call<SignInDataModel> call = retroApiInterface.signIn(postModel, UserSingleton.getUUID(getActivity()));

call.enqueue(new CustomResponseNoNewToken<SignInDataModel>(getActivity(), true, true) {

@Override

public void onCustomSuccess(Call<SignInDataModel> call, Response<SignInDataModel> response) {

super.onCustomSuccess(call, response);

SignInDataModel model = response.body();

if(model.data.newToken != null){

UserSingleton.getInstance().updateToken(model.data.newToken);

}

if(model.data.user != null){

UserSingleton.getInstance().updateUser(model.data.user);

}

SpikaApp.getEnterpriseSharedPreferences().setCustomBoolean(Const.PreferencesKeys.REMEMBER\_ME, cbRememberMe.isChecked());

if(cbRememberMe.isChecked()){

SpikaApp.getEnterpriseSharedPreferences().setCustomString(Const.PreferencesKeys.SHA1\_PASSWORD, sha1Password);

SpikaApp.getEnterpriseSharedPreferences().setCustomString(Const.PreferencesKeys.USERNAME\_LOGIN, etUsername.getText().toString());

SpikaApp.getEnterpriseSharedPreferences().setCustomString(Const.PreferencesKeys.ORGANIZATION\_LOGIN, etOrganization.getText().toString());

}

hideProgress();

HomeActivity.startActivity(getActivity());

finish();

}

@Override

public void onTryAgain(Call<SignInDataModel> call, Response<SignInDataModel> response) {

super.onTryAgain(call, response);

showProgress();

loginToServer(sha1Password, sha1Secret);

}

});

PostSignModel is model with data which are send to server. It contains username, password(crypted), secret(crypted), push token, organization and uuid(unique id of device). If combination of user and password is correct, then server will return all user data to SignInDataModel. Next step is for save all data to shared preferences and start using chat. Token for every next api is in SignInDataModel. All user data is saved to sharedpreferences and UserSingleton class is used to manage and fetching logined user data. Eg, for user token:

UserSingleton.getInstance().getUser().myToken

Get messages (Example)

There are three api for get messages from server. But first for get message there are some values we need to send to server:

1. roomID - room id of selected chat
2. lastMessageId - id of last message showed in list
3. direction - old, new or allto. With old key - server returns last 50 messages, with new key - server returns all new messages from lastmessageid, with key allto - server returns all message from lastmessageid include message with lastmessageid.
4. access-token - token of user
5. uuid - uuid of devices. First three values goes to GET params, and last two goes in header.

Room id is passed from previously activity (from recent, from notification,...). When ChatActivity is opened, we need to call api with direction old, and lastMessageId of last message from local database. On ChatActivity resume (eq. send application to background from ChatActivity and bring it to foreground back) we need to call api with direction new, and lastMessageId of last message showed in list. Direction allto is used when ChatActivity is opened from favorite or from message search. In that api lastMessageId is id of searched or favorited message. Method for old direction:

private void getMessages(final boolean isInit, String lastMessageId, final boolean toShowProgress) {

if (toShowProgress) showProgress();

//chatactivity is opened from search or favorite feature

if (!TextUtils.isEmpty(getAllFromMessageId)) {

getAllFromMessageId();

return;

}

//there is not last message, last message is "0"

if (TextUtils.isEmpty(lastMessageId)) {

lastMessageId = "0";

}

final String finalLastMessageId = lastMessageId;

SpikaOSRetroApiInterface retroApiInterface = getRetrofit().create(SpikaOSRetroApiInterface.class);

Call<GetMessagesModel> call = retroApiInterface.getMessages(activeUser.roomID, lastMessageId, "old", UserSingleton.getInstance().getUser().myToken, UserSingleton.getUUID(getActivity()));

call.enqueue(new CustomResponseNoNewToken<GetMessagesModel>(getActivity(), false, false) {

@Override

public void onCustomSuccess(Call<GetMessagesModel> call, Response<GetMessagesModel> response) {

super.onCustomSuccess(call, response);

dataLoadedFromNetwork = true;

lastDataFromServer.clear();

lastDataFromServer.addAll(response.body().data.messages);

if (isInit) {

adapterMessages.clearMessages();

lastVisibleItem = response.body().data.messages.size();

}

boolean isPaging = !isInit;

adapterMessages.addMessages(response.body().data.messages, isPaging);

if (isInit && rvMessages.getAdapter().getItemCount() > 1 && getAllFromMessageId == null) {

scrollRecyclerToBottomWithAnimation();

}

//add undelivered (message which are failed to send) to bottom of list

if(isInit){

MessageJsonDB.getUnDeliveredMessagesAsync(activeUser.roomID, new MessageJsonDB.OnDatabaseFinish() {

@Override

public void onGetMessages(List<Message> messages) {

adapterMessages.addUndeliveredMessages(messages);

scrollRecyclerToBottom();

}

});

}

hideProgress();

//for forward message

ChatActivity.this.runOnUiThread(new Runnable() {

@Override

public void run() {

ChatActivity.this.sendPassedMessage();

}

});

//save to database

MessageJsonDB.addMessagesList(activeUser.roomID, response.body().data.messages, new MessageJsonDB.OnDatabaseFinish() {

@Override

public void onGetMessages(List<Message> messages) {

LogCS.d("DATABASE ADDED MESSAGES");

}

});

}

@Override

public void onTryAgain(Call<GetMessagesModel> call, Response<GetMessagesModel> response) {

super.onTryAgain(call, response);

showProgress();

getMessages(isInit, finalLastMessageId, toShowProgress);

}

});

}

Method for new direction:

private void getLatestMessages(String lastMessageId) {

if (TextUtils.isEmpty(lastMessageId)) {

lastMessageId = "0";

}

final String finalLastMessageId = lastMessageId;

SpikaOSRetroApiInterface retroApiInterface = getRetrofit().create(SpikaOSRetroApiInterface.class);

Call<GetMessagesModel> call = retroApiInterface.getMessages(activeUser.roomID, lastMessageId, "new", UserSingleton.getInstance().getUser().myToken, UserSingleton.getUUID(getActivity()));

call.enqueue(new CustomResponseNoNewToken<GetMessagesModel>(getActivity(), false, false) {

@Override

public void onCustomSuccess(Call<GetMessagesModel> call, Response<GetMessagesModel> response) {

super.onCustomSuccess(call, response);

if (response.body().data.messages.size() == 0) {

return;

}

boolean toScrollBottom = false;

LinearLayoutManager llManager = (LinearLayoutManager) rvMessages.getLayoutManager();

if (llManager.findLastVisibleItemPosition() == rvMessages.getAdapter().getItemCount() - 1) {

toScrollBottom = true;

}

adapterMessages.addLatestMessagesWithCheck(response.body().data.messages);

lastVisibleItem = adapterMessages.getItemCount();

if (toScrollBottom) {

scrollRecyclerToBottom();

} else {

if (newMessagesButton.getVisibility() == View.GONE) {

AnimationUtils.fadeThenGoneOrVisible(newMessagesButton, 0, 1, 250);

}

}

}

});

}

Method for allto direction:

private void getAllFromMessageId() {

SpikaOSRetroApiInterface retroApiInterface = getRetrofit().create(SpikaOSRetroApiInterface.class);

Call<GetMessagesModel> call = retroApiInterface.getMessages(activeUser.roomID, getAllFromMessageId, "allto", UserSingleton.getInstance().getUser().myToken, UserSingleton.getUUID(getActivity()));

call.enqueue(new CustomResponseNoNewToken<GetMessagesModel>(getActivity(), false, false) {

@Override

public void onCustomSuccess(Call<GetMessagesModel> call, Response<GetMessagesModel> response) {

super.onCustomSuccess(call, response);

lastDataFromServer.clear();

lastDataFromServer.addAll(response.body().data.messages);

adapterMessages.clearMessages();

adapterMessages.setSelectedMessageId(getAllFromMessageId); //animate circle by selected message

lastVisibleItem = response.body().data.messages.size();

adapterMessages.addMessages(response.body().data.messages, false);

disablePaging = true;

int position = 0;

int i = 0;

//find selected message, and scroll listview to that message

for(Message item : adapterMessages.getAllMessage()){

if(item.\_id.equals(getAllFromMessageId)){

position = i;

break;

}

i++;

}

rvMessages.scrollToPosition(position);

getAllFromMessageId = null;

hideProgress();

}

@Override

public void onTryAgain(Call<GetMessagesModel> call, Response<GetMessagesModel> response) {

super.onTryAgain(call, response);

showProgress();

getAllFromMessageId();

}

});

}

RoomId is stored in user object. User object contains roomId, logined userId, logined user avatar, logined user name and when logined user is created his account. It 'lives' in memory and used in current ChatActivity. It is helper model for storing variables which we will need in current ChatActivity. RoomId is from other previous activity, and other data in user are from UserSingleton.

# Send message ( Socket )

For sending message, receiving message while application is in foreground, typing and signaling for call application uses socket.io. more information about socket.io can be found at <https://socket.io/>. Every time when user brings back application to foreground, application tries to establised socket connection, and when user sends application to background application disconnects from socket. Connecting and disconnecting from socket are managed in SpikaApp class (Application class).

//\*\*\*\*\*\*stuff for check if application is enter background

private BroadcastReceiverImplementation broadcastReceiverImplementation = new BroadcastReceiverImplementation();

private class BroadcastReceiverImplementation extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

if (intent.getAction().equals(ApplicationStateManager.APPLICATION\_PAUSED)) {

LogCS.e("\*\*\*\*\*\*\* PAUSE APP \*\*\*\*\*\*\*");

SocketManager.getInstance().closeAndDisconnectEnterpriseSocket();

SocketManager.getInstance().closeSocketManager();

LocalBroadcastManager.getInstance(getAppContext()).sendBroadcast(new Intent(Const.ReceiverIntents.APPLICATION\_TO\_BACKGROUND));

} else if (intent.getAction().equals(ApplicationStateManager.APPLICATION\_RESUMED)) {

LogCS.e("\*\*\*\*\*\*\* RESUMED APP \*\*\*\*\*\*\*" + this.getClass().getName());

SocketManager.getInstance().initSocketManager();

SocketManager.getInstance().tryToReconnectToEnterprise();

LocalBroadcastManager.getInstance(getAppContext()).sendBroadcast(new Intent(Const.ReceiverIntents.APPLICATION\_TO\_FOREGROUND));

new OfflineFeatureHelper().sentUndeliveredMessages(context, false);

}

}

}

SocketManager is class which manage with socket connection. There is lots of events for subscribe, eq for receive message event is "newmessage".

### **Send text message flow**

When user typed text and clicked on send button, first check if edittext is empty, if edittext is not empty, then application continues with send message flow. (1) Next step is to create message object. Fill that object with data, with type TYPE\_TEXT and with crypted text from edtittext and other information like localId. If text from edittext is link, then message change message type and use flow for sending link. If text is not link, then continue with send message flow.(2) Next step is to check if socket is connected. If it is true, then send message with socket, else use api for send message. (Case when socket would't be connected: user click on send button and immediately send application to background).(3)

##### **Send message with socket**

When send message with socket, first is to create JSONObject which will be passed to socket. In EmitJsonCreator class there is method for generating JSNOObject from Message object. Then emit sendMessage event with JSONObject to socket.(3) After that add new message to message list. Current status of sent message is 'sending'. When server get message, then server save message to database and return it with event 'newmessage'. That event is for all receive message, so we need to check if message from event is our sent message or message from other user. Use local id to check if message is our or from other user. If message from newmessage event is message which user sent then update message status to 'sent' and refresh message list.(5)

##### **Send message with api**

If socket is not connected, use send message api to send message. Create SendMessagemModel, and pass it to api, api returns message in response (4), and then change message status and refresh message list. (5)

##### **Send message flow in code**

(1)

protected void onButtonSendClicked() {

String message = etMessage.getText().toString();

String trimmedMessage = message.trim();

if(trimmedMessage.isEmpty()){

return;

}

sendMessage();

}

(2)

private void sendMessage(String messageText) {

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*parse link\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

boolean hasLink = false;

String textMessage = messageText;

String checkForLink = Utils.checkForLink(textMessage);

if (checkForLink != null) {

hasLink = true;

//set hasLink to true when attributes implements on api

}

final Message message = new Message();

message.fillMessageForSend(activeUser, Cryptor.getInstance().encryptString(messageText), Const.MessageType.TYPE\_TEXT, null, null);

if (hasLink) {

btnSend.setVisibility(View.INVISIBLE);

pbAboveSend.setVisibility(View.VISIBLE);

new ParseUrlLinkMetadata(checkForLink, new ParseUrlLinkMetadata.OnUrlParsed() {

@Override

public void onUrlParsed(ParsedUrlData data) {

btnSend.setVisibility(View.VISIBLE);

pbAboveSend.setVisibility(View.GONE);

etMessage.setText("");

Attributes att = new Attributes();

att.linkData = data;

message.attributes = att;

sendMessageWithSocket(message);

}

}).execute();

} else {

etMessage.setText("");

sendMessageWithSocket(message);

}

}

(3)

private void sendMessageWithSocket(Message message){

if (SocketManager.getInstance().isEnterpriseSocketConnect()) {

JSONObject emitMessage = EmitJsonCreator.createEmitSendMessage(message);

SocketManager.getInstance().emitEnterpriseMessage(Const.EmitEnterpriseMessages.SEND\_MESSAGE, emitMessage);

} else {

sendMessageWithApi(message, true);

}

onMessageSent(message);

}

(4)

private void sendMessageWithApi(final Message message, final boolean toCallOnReceive){

String sendMessage = message.message;

int type = message.type;

final String roomID = message.roomID;

MessageRetroApiInterface retroApiInterface = getRetrofit().create(MessageRetroApiInterface.class);

SendMessageModel postModel = new SendMessageModel(roomID, sendMessage, type, message.localID);

if(message.file != null){

postModel.file = message.file;

}

if(message.location != null){

postModel.location = message.location;

}

if(message.attributes != null){

postModel.attributes = message.attributes;

}

postModel.localID = message.localID;

Call<GetMessageModel> call = retroApiInterface.sendMessage(postModel, UserSingleton.getInstance().getUser().myToken, UserSingleton.getUUID(getActivity()));

LogCS.logCallRetro(call);

call.enqueue(new CustomResponse<GetMessageModel>(getActivity(), false, false) {

@Override

public void onNewToken() {

sendMessageWithApi(message, toCallOnReceive);

}

@Override

public void onCustomSuccess(Call<GetMessageModel> call, Response<GetMessageModel> response) {

super.onCustomSuccess(call, response);

message.created = response.body().data.message.created;

message.\_id = response.body().data.message.\_id;

if(toCallOnReceive) {

if(!SocketManager.getInstance().isEnterpriseSocketConnect()){

onMessageReceived(message);

}

}

}

@Override

public void onCustomFailed(Call<GetMessageModel> call, Response<GetMessageModel> response) {

super.onCustomFailed(call, response);

message.status = Const.MessageStatus.NOT\_SENT;

rvMessages.getAdapter().notifyDataSetChanged();

MessageJsonDB.addMessage(roomID, message, null);

}

@Override

public void onFailure(Call<GetMessageModel> call, Throwable t) {

super.onFailure(call, t);

message.status = Const.MessageStatus.NOT\_SENT;

adapterMessages.setUnsentMessage(message);

MessageJsonDB.addMessage(roomID, message, null);

}

});

}

(5)

private void onMessageReceived(final Message message) {

if(sentUndeliveredMessages.contains(message.localID)){

return;

}

if(!message.roomID.equals(activeUser.roomID)){

return;

}

if (sentMessages.contains(message.localID)) {

this.runOnUiThread(new Runnable() {

@Override

public void run() {

adapterMessages.setDeliveredMessage(message);

}

});

sentMessages.remove(message.localID);

} else {

message.status = Const.MessageStatus.RECEIVED;

this.runOnUiThread(new Runnable() {

@Override

public void run() {

boolean toScrollBottom = false;

LinearLayoutManager llManager = (LinearLayoutManager) rvMessages.getLayoutManager();

if (llManager.findLastVisibleItemPosition() == rvMessages.getAdapter().getItemCount() - 1) {

toScrollBottom = true;

}

adapterMessages.addReceivedMessage(message);

if (toScrollBottom) {

scrollRecyclerToBottom();

} else {

if (newMessagesButton.getVisibility() == View.GONE) {

AnimationUtils.fadeThenGoneOrVisible(newMessagesButton, 0, 1, 250);

}

}

mpNotificationSound.start();

}

});

if (!message.user.\_id.equals(activeUser.userID)) {

sendOpenMessage(message.\_id);

}

}

this.runOnUiThread(new Runnable() {

@Override

public void run() {

lastVisibleItem = rvMessages.getAdapter().getItemCount();

}

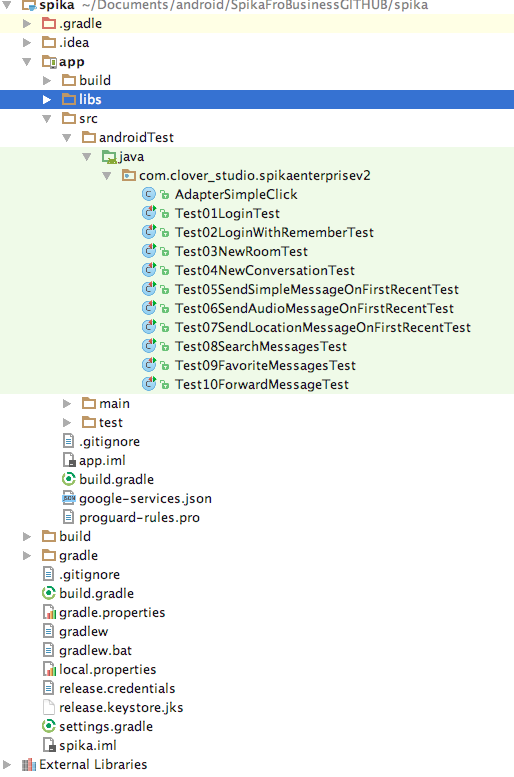
});

MessageJsonDB.addMessage(activeUser.roomID, message, null);

LocalBroadcastManager.getInstance(getActivity()).sendBroadcast(new Intent(Const.ReceiverIntents.NEW\_MESSAGE\_IN\_CHAT));

}

UITests

Espresso librarry is used for UITests in Android. Information for setting up and using espresso can be found [here](https://developer.android.com/training/testing/ui-testing/espresso-testing.html). In S4B for android there are 10 test for now. Every new test we added to S4B will be pushed on git immediately. Tests are located in src/androidTest folder.To run test just right click on test and select run test. This Test02LoginWithRememberTest is login test with remember me checked. This is example of code.

@RunWith(AndroidJUnit4.class)

public class Test02LoginWithRememberTest {

@Rule

public ActivityTestRule<LoginActivity> mActivityRule = new ActivityTestRule<>(LoginActivity.class);

@Test

public void loginWithRememberMe() {

onView(withId(R.id.etOrganization)).perform(clearText()).perform(typeText("clover"));

onView(withId(R.id.etUserName)).perform(clearText()).perform(typeText("pavel.nedved"));

onView(withId(R.id.etPassword)).perform(clearText()).perform(typeText("cloverpass013"), closeSoftKeyboard());

onView(withId(R.id.cbRememberMe)).perform(click());

onView(withId(R.id.tvLogin)).perform(click());

try {

Thread.sleep(2000);

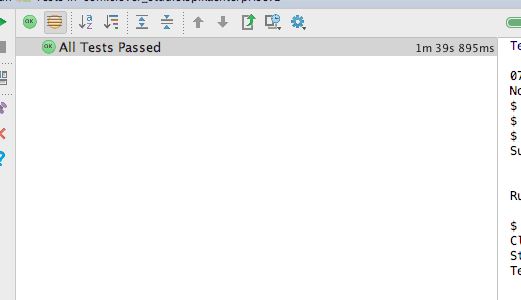
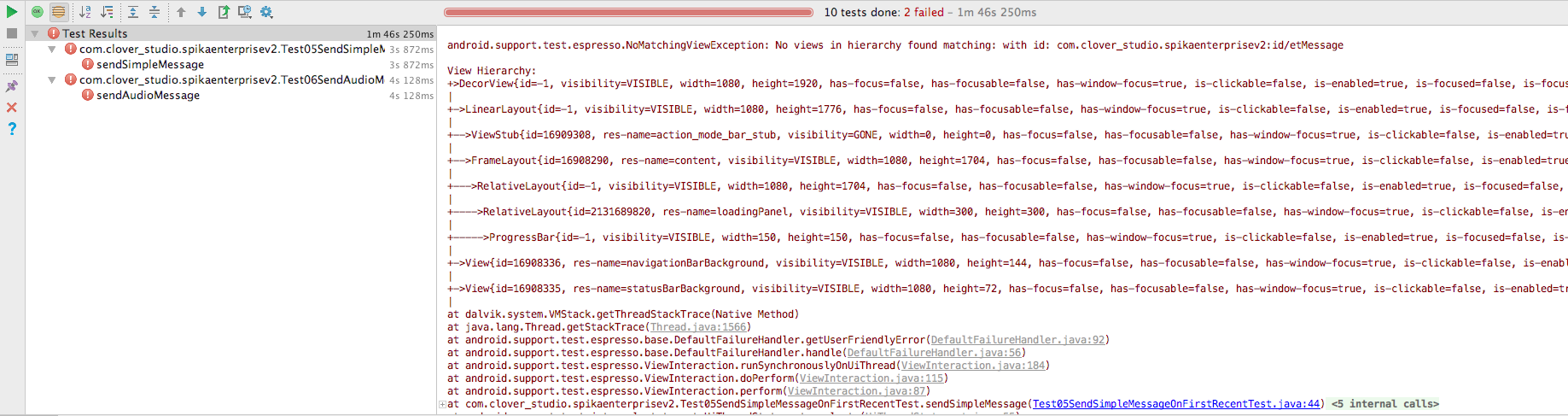
} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

To run all tests at once, right click on com.clover\_studio.spikaenterprisev2 package in androidTest/java folder and select run tests.  
All tests successfully passed:Some tests failed:

# Part 1 - import SDK

e released Android SDK so our clients use Spika from their service easily. Here is simple tutorial how to use the SDK. Android SDK can be used on two way, as module or as ar library. First thing is to download Spika for Business from GIT. To try our SDK, download Spika For Business open menu i click on try SDK option.

#### **Use SDK like ar library**

After downloaded source code, open Spika for Business in Android Studio. Run or just build Spika for Business. After that ar library is generated in folder spikasdk/build/outputs/arr. Copy release arr file to project which will use SDK in folder libs. Internet permissions is required for SDK, so it si required to allow internet permission in project AndroidManifest. Retrofit module is also required for SDK, if project doesn’t use retrofit, please add it. Please add this lines in app gradle above dependencies

repositories{

flatDir{

dirs 'libs'

}

}

Then add this lines in dependencies (first 3 lines is for retrofit):

compile 'com.squareup.retrofit2:retrofit:2.2.0'

compile 'com.squareup.retrofit2:converter-scalars:2.2.0'

compile 'com.squareup.retrofit2:converter-gson:2.2.0'

compile(name:'spikasdk-release', ext:'aar')

SDK is now ready for use. Please read [SDK](http://document.spika.chat/android/using-android-sdk/part-2-use-sdk) documentation before using SDK.

#### **Use SDK like module (recommended)**

After downloaded source code, copy spikasdk folder to project. spikasdk folde has to be at same level as app folder. Next step is to add include ':spikasdk' below include ':app' in settings.gradle file.

include ':app'

include ':spikasdk'

Add this line to app gradle dependencies:

compile project(path: ':spikasdk')

Internet permissions is required for SDK, so it si required to allow internet permission in project AndroidManifest. This is better way for using SDK and it is recommended to use. SDK is now ready for use. Please read [SDK](http://document.spika.chat/android/using-android-sdk/part-2-use-sdk) documentation before using SDK.

# Part 2 - use SDK

For use SDK apiKey is required. Api key can be generated in Admin Console by organization Admin.

#### **Init SDK**

MainSpikaSDK.getInstance().init(apiKey, baseUrl, new SpikaSDKListener() {

@Override

public void onLog(String log) {

tvOutput.setText(log);

}

@Override

public void onData(String data) {

}

@Override

public void onError(int code) {

}

});

* apiKey - key generated in Admin Console
* baseUrl - endpoint of your Spika installation. Most of cases it will be http(s)://[your server]/api/v3
* listener
  + onLog(String log) - data for logging
  + onError(int code) - error with code - (here is code list error)
  + onData(String data) - success with data (data can be in string ready for json formatting or just simple string)

#### **Sing in**

MainSpikaSDK.getInstance().signIn(organization, username, password, new SpikaSDKListener() {

@Override

public void onLog(String log) {

hideProgress();

tvOutput.setText(log);

}

@Override

public void onData(String dataInJson) {

hideProgress();

try {

JSONObject object = new JSONObject(dataInJson);

String json = object.toString(4);

tvOutput.setText(json);

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText(dataInJson);

}

}

@Override

public void onError(int code) {

hideProgress();

tvOutput.setText("Error code: " + code);

}

});

Listener returns string ready for json formatting. [(Sample of response)](http://document.spika.chat/android/using-android-sdk/sdk-json-responses-sample)  
This response return access token and it is recommended to save access token to shared preference of main application. SDK keeps access token in memory for future calls, but if main application provide access token to SDK, then SDK uses provided access token.

#### **Send Message**

MainSpikaSDK.getInstance().sendMessage(message, target, targetType, messageType, fileModel, new SpikaSDKListener() {

@Override

public void onLog(String log) {

hideProgress();

tvOutput.setText(log);

}

@Override

public void onData(String dataInJson) {

hideProgress();

try {

JSONObject object = new JSONObject(dataInJson);

String json = object.toString(4);

tvOutput.setText(json);

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText(dataInJson);

}

}

@Override

public void onError(int code) {

hideProgress();

tvOutput.setText("Error code: " + code);

}

});

* message - message for send
* targetType - type of chat (1 private, 2 group, 3 room)
* target - target to send message (username if type is private)
* messageType - message type, 1 - text message, 2 - file message.
* fileModel - parsed response from upload file, ignored if messageType is 1.

Listener returns string ready for json formatting contains new sent message. [(Sample of response)](http://document.spika.chat/android/using-android-sdk/sdk-json-responses-sample)

#### **Send File**

To send file, first step is upload file to server.

MainSpikaSDK.getInstance().uploadFile(file, new SpikaSDKListener() {

@Override

public void onLog(String log) {

dialog.dismiss();

tvOutput.setText(log);

}

@Override

public void onData(String dataInJson) {

dialog.dismiss();

try {

JSONObject object = new JSONObject(dataInJson);

String json = object.toString(4);

tvOutput.setText(json);

sendMessageFile(object, fileName);

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText(dataInJson);

}

}

@Override

public void onError(int code) {

dialog.dismiss();

tvOutput.setText("Error code: " + code);

}

}, new SpikaSDKProgressListener() {

@Override

public void onProgress(int percentage, long finished) {

dialog.setCurrent((int) finished);

}

@Override

public void onMax(int max) {

dialog.setMax(max);

}

});

* file - file for saving downloaded file
* listener - listener with response data. (Sample is here) If file is type of image then thumb and file are returned otherway just file is returned.
* progressListener - listener for upload progess

After file is uploaded, parse response data and send message.

showProgress();

try {

PostSendMessage.File fileAll = new PostSendMessage.File();

JSONObject file = jsonObject.getJSONObject("file");

PostSendMessage.FileInside fileModel = new PostSendMessage.FileInside();

fileModel.id = file.getString("fileId");

fileModel.size = file.getInt("size");

fileModel.mimeType = file.getString("mimeType");

fileModel.name = fileName;  fileAll.file = fileModel;

PostSendMessage.FileInside thumbModel = null;

if(jsonObject.has("thumbnail")){

JSONObject thumbnail = jsonObject.getJSONObject("thumbnail");

thumbModel = new PostSendMessage.FileInside();

thumbModel.id = thumbnail.getString("fileId");

thumbModel.size = thumbnail.getInt("size");

thumbModel.mimeType = thumbnail.getString("mimeType");

thumbModel.name = fileName;

fileAll.thumb = thumbModel;

}

MainSpikaSDK.getInstance().sendMessage("", username, targetType, messageType, fileAll, new SpikaSDKListener() {

@Override

public void onLog(String log) {

hideProgress();

tvOutput.setText(log);

}

@Override

public void onData(String dataInJson) {

hideProgress();

try {

JSONObject object = new JSONObject(dataInJson);

String json = object.toString(4);

tvOutput.setText(json);

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText(dataInJson);

}

}

@Override

public void onError(int code) {

hideProgress();

tvOutput.setText("Error code: " + code);

}

});

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText("Error: " + e.toString());

}

#### **Get messages**

MainSpikaSDK.getInstance().getLastMessages(roomId, new SpikaSDKListener() {

@Override

public void onLog(String log) {

hideProgress();

tvOutput.setText(log);

}

@Override

public void onData(String dataInJson) {

hideProgress();

try {

JSONObject object = new JSONObject(dataInJson);

String json = object.toString(4);

tvOutput.setText(json);

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText(dataInJson);

}

}

@Override

public void onError(int code) {

hideProgress();

tvOutput.setText("Error code: " + code);

}

});

This is method for get last 50 messages from passed room id.

MainSpikaSDK.getInstance().getMessages(roomId, lastMessageId, direction, new SpikaSDKListener() {

@Override

public void onLog(String log) {

hideProgress();

tvOutput.setText(log);

}

@Override

public void onData(String dataInJson) {

hideProgress();

try {

JSONObject object = new JSONObject(dataInJson);

String json = object.toString(4);

tvOutput.setText(json);

} catch (JSONException e) {

e.printStackTrace();

tvOutput.setText(dataInJson);

}

}

@Override

public void onError(int code) {

hideProgress();

tvOutput.setText("Error code: " + code);

}

});

* direction - old or new
* lastMessageId - used for paging, if you want read from first positon send "0".
* roomId - id of chat
* listener - string ready for json formating [(Sample of response)](http://document.spika.chat/android/using-android-sdk/sdk-json-responses-sample)

#### **Download image**

MainSpikaSDK.getInstance().downloadFile(fileId, filePath, new SpikaSDKProgressListener() {

@Override

public void onProgress(int percentage, final long finished) {

runOnUiThread(new Runnable() {

@Override

public void run() {

dialog.setCurrent((int) finished);

}

});

}

@Override

public void onMax(final int max) {

runOnUiThread(new Runnable() {

@Override

public void run() {

dialog.setMax(max);

}

});

}

}, new SpikaSDKListener() {

@Override

public void onLog(final String log) {

runOnUiThread(new Runnable() {

@Override

public void run() {

dialog.dismiss();

tvOutput.setText(log);

}

});

}

@Override

public void onData(final String dataInJson) {

runOnUiThread(new Runnable() {

@Override

public void run() {

dialog.dismiss();

String path = dataInJson;

showImage.setImageBitmap(BitmapFactory.decodeFile(path));

rlForImage.setVisibility(View.VISIBLE);

}

});

}

@Override

public void onError(final int code) {

runOnUiThread(new Runnable() {

@Override

public void run() {

dialog.dismiss();

tvOutput.setText("Error code: " + code);

}

});

}

});

* fileId - id of file for download
* filePath - path where file will be saved
* progressListener - listener for download progress

Access token can be passed to all methods.