API ROOT URL : <http://api.chatndate.com/web/api>

While using API in your java code, please keep this url in variable because root URL can be changed in future.

To test API I am using Chrom extension POST

These services are avaible with current API

 GET /users: list all users page by page;

 POST /users: create a new user;

 GET /users/123: return the details of the user 123;

 GET /profile: list all users profiles page by page;

 GET /profiles/123: return the full profile of the user 123;

Error handler:

In case of error you will get response in following format:

Here important is status code, which will point you exact issue with request, these status code are standards provided by all API services, so we will follow same. Message is custom attribute it will get more specific explanation of error.

{

"name": "Unprocessable entity",

"message": "{\"email\":[\"This email address has already been taken\"]}",

"code": 0,

"status": 422,

"type": "yii\\web\\HttpException"

}

Complete Error responses Guide

* 200: OK. Everything worked as expected.
* 201: A resource was successfully created in response to a POST request. The Location header contains the URL pointing to the newly created resource.
* 204: The request was handled successfully and the response contains no body content (like a DELETE request).
* 304: The resource was not modified. You can use the cached version.
* 400: Bad request. This could be caused by various actions by the user, such as providing invalid JSON data in the request body, providing invalid action parameters, etc.
* 401: Authentication failed.
* 403: The authenticated user is not allowed to access the specified API endpoint.
* 404: The requested resource does not exist.
* 405: Method not allowed. Please check the Allow header for the allowed HTTP methods.
* 415: Unsupported media type. The requested content type or version number is invalid.
* 422: Data validation failed (in response to a POST request, for example). Please check the response body for detailed error messages.
* 429: Too many requests. The request was rejected due to rate limiting.
* 500: Internal server error. This could be caused by internal program errors.

Webservice Objects

GET /users

Purpose: List all users by page

**User object URL :** <http://api.chatndate.com/web/api/users>

Here we will call user is object on user object we can perform several action our first action is list down all the users stored in database.

To access user object you need to call above user, It will return you list of users in json format, currently per feed limit is 20, we may increase or decrease per page limit

**Header information for above request**

**Connection →**Keep-Alive

**Content-Length →**7586

**Content-Type →**application/json; charset=UTF-8

**Date →**Sun, 01 May 2016 03:03:58 GMT

**Keep-Alive →**timeout=5, max=100

**Link →**<http://localhost/ChatWeb/web/api/users?page=1>; rel=self, <http://localhost/ChatWeb/web/api/users?page=2>; rel=next, <http://localhost/ChatWeb/web/api/users?page=2>; rel=last

**Server →**Apache/2.4.9 (Win64) PHP/5.5.12

**X-Pagination-Current-Page →**1

**X-Pagination-Page-Count →**2

**X-Pagination-Per-Page →**20

**X-Pagination-Total-Count →**26

**X-Powered-By →**PHP/5.5.12

Header contained very useful information

**X-Pagination Tell us how many total record found respective to our query, what is our current page number**

<http://api.chatndate.com/web/api/users?page=2>

It will fetch next 20 feeds, because we have limit of 20 feeds per page

Get Users along with Profile

<http://api.chatndate.com/web/api/users?expand=profile>

OR

<http://api.chatndate.com/web/api/users/1?expand=profile>

Limited user table fields along with Profile

<http://api.chatndate.com/web/api/users?expand=profile&fields=id,username>

Or

<http://api.chatndate.com/web/api/users/30?expand=profile&fields=id,username>

The above webservice will return Users full profile and id,username field from user table

GET /users/30

Purpose: List individual users detail

**User object URL :** <http://api.chatndate.com/web/api/users/30>

This API’s working is same but rather then providing list of users in array it will give you json object for user with ID 30

POST /users

Purpose: Create new user

**User object URL :** <http://api.chatndate.com/web/api/users>

|  |  |  |
| --- | --- | --- |
| Input name | Data type | Example |
| email | Email address | Test50@test.com |
| password | string | 123456 |
| first\_name | string | john |
| last\_name | string | smith |
| languages\_id | Small int | 1 = English  2 = spanish |

When you hit above API url with post data, This API assume you are trying to insert data in users object, which mean it is create new user request.

Currently I am taking just simple information

email

password

first\_name

last\_name

languages\_id

Because our High priority work is to build chat API, we will expand profile information later, I already created table to store users full profile but it will make our testing job hard if need to fill big form to create new user.So we will expand this Profile creation API later with more inputs.

Now we have API where you can create new user, list users, view user individually.

It will help us to reach chat screen, then our next move will be sending and receiving chat messages.

GET /profiles

Purpose: List all profiles of users by page

**User object URL :** http://api.chatndate.com/web/api/profiles

GET /profiles

Purpose: detail view of individual profile

**User object URL :** http://api.chatndate.com/web/api/profiles/30

Languages:

GET /languages

<http://api.chatndate.com/web/api/languages>

Output:

[

{

"id": 1,

"language\_name": "English",

"language\_code": "en"

},

{

"id": 2,

"language\_name": "Spanish",

"language\_code": "es"

}

]

CHAT Web Service

<http://api.chatndate.com/web/api/chats>

POST /chats

Chat window will have language section

Each message will take input as follow:

|  |  |  |
| --- | --- | --- |
| Input name | Data type | Example |
| from\_id | int | 30  Id of user who is sending this message |
| to\_id | int | 31  Id of user who will receive this message |
| chat\_message | string | Context of the message, it will be plain text message |
| languages\_id | Small int | 1 = English  2 = spanish |
| rabbitmq\_exchange\_name | string | chat.message.exchange  this attribute belongs to user object, you can retrieve and pass here |
| rabbitmq\_queue\_name | string | chat.message.user.1  this attribute belongs to user object, you can retrieve and pass here |
| rabbitmq\_routing\_key | string | chat.message.user.1  this attribute belongs to user object, you can retrieve and pass here |
|  |  |  |
|  |  |  |

When we push data to our Chat Web Service

First our publisher (RabbitMQ) will add this data in 2 queues

1. User specific queue (Android app will be consumer of this queue)
2. Mysql Database queue (at server my php script will be consumer of this queue)

Chat History

http://api.chatndate.com/web/api/chats

http://api.chatndate.com/web/api/chats?page=2

http://api.chatndate.com/web/api/chats?users=1,2

http://api.chatndate.com/web/api/chats?history=today

http://api.chatndate.com/web/api/chats?history=yesterday

http://api.chatndate.com/web/api/chats?history=currentweek

http://api.chatndate.com/web/api/chats?history=currentmonth

http://api.chatndate.com/web/api/chats?history=last2days

http://api.chatndate.com/web/api/chats?history=last7days

http://api.chatndate.com/web/api/chats?history=last10days

http://api.chatndate.com/web/api/chats?history=last31days

http://api.chatndate.com/web/api/chats?orderby=desc&startpoint=78124568

Options:

orderby = asc|desc

startpoint: chat\_message\_id from where you want to get records

Authenticates Web Service

[http://api.chatndate.com/web/api/authenticates](http://api.chatndate.com/web/api/authenticates%20)

GET /authenticates

Our Authenticate API use Basic Auth protocol method to validate user

Where you need to send extra header when you hit api/authenticates url

In basic auth method we send authorization header with value

addHeader("authorization", "Basic dGVzdDI6MTIzNDU2")

Code to generate basic auth header in java

Where test2@gmail.com is username and 123456 is password

String encoding = Base64Encoder.encode ("test2@test.com:123456");

HttpPost httppost = new HttpPost("[http://api.chatndate.com/web/api/authenticates](http://api.chatndate.com/web/api/authenticates%20)

");

httppost.setHeader("Authorization", "Basic " + encoding);

**Code generated from postman, how it will create request in java**

OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()

.url("http://api.chatndate.com/web/api/authenticates")

.get()

.addHeader("authorization", "Basic dGVzdDI6MTIzNDU2")

.addHeader("cache-control", "no-cache")

.addHeader("postman-token", "d28a4598-5e6b-dcaa-b3c3-20446a60934d")

.build();

Response response = client.newCall(request).execute();

At very first login screen you will take user inputs with email and password, and send this as a header to authenticate these details, in return of if user credential exist I will return you user object with details same we found in api/users/1

Now further you can set this return user as current user, store this user data in sessions or global variable and use through application.