

messeji

Typeset multi-page chat histories, including from external JSON files.

v0.3.0

April 20, 2025

<https://github.com/Tanikai/messeji>

Kai Anter <<https://www.anter.dev>>

ABSTRACT

messēji ("Message" in Japanese) is a Typst package for typesetting chat histories in a modern, minimal design, inspired by popular messengers. No manual copying to your Typst document required, just pass in a JSON file.

NOTE

Build this document with `typst compile messeji-guide.typ --root ..`
Function reference generated with [tidy](#).

CONTENTS

I Introduction	2
II Usage	3
II.a Import	3
II.b Message Structure	3
II.c Basic example from JSON	3
II.d Basic example directly in Typst	4
II.e Messages with Images	6
III Customization	8
III.a Themes	8
III.b Custom timestamp and date-change format	9
IV Function Reference	11

I INTRODUCTION

With *messēji*, you can easily typeset long chats by passing in your history as a JSON file. No clutter in your Typst document, and you can focus on your actual content. Main features of *messēji* include:

- Support for quoted messages
- Image messages (with and without captions)
- Displaying timestamps between messages
- Simple data model to read from external files (JSON, YAML, ...) if you want to typeset very long chat histories
- Reacting to messages

Currently, it does not support:

- Group chats (only 1-on-1 chats)
- Displaying names / profile pictures next to the messages

These features are currently not planned to be implemented. However, if you need them, [create an issue](#) and I'll look into it (if I have the time).

II USAGE

II.a Import

Add the following import statement to the top of your document:

```
1 #import "@preview/messeji:0.3.0": messeji typ
```

II.b Message Structure

The messages have to be in the following structure (dictionary formatted as JSON):

```
1 { json5
2   // Optional, can be combined with image
3   "msg": "Actual Message",
4
5   // Optional, can be combined with msg
6   "image": "path_to_image.jpg",
7
8   // Required, true for right, false for left
9   "from_me": true,
10
11  // Optional, in ISO 8601 format
12  "date": "2026-12-25T09:41:00",
13
14  // Optional
15  "ref": "Previous message that is being quoted",
16
17  // Optional, only single emojis are tested / supported
18  "reaction": "♥"
19 }
```

The order of the fields is not important. You can use other filetypes if you want to, but the key names have to be the same, and the `date` value has to be in ISO 8601 format.

II.c Basic example from JSON

```
1 #set text(font: "Helvetica Neue") typ
2 #let parsed-data = json("output.example.json") // list of messages
3 #messeji(chat-data: parsed-data)
```

The code snippet above assumes the following structure from the JSON file:

```
1 [ json5
2   {
3     "date": "2026-12-25T09:41:00",
4     "msg": "Merry Christmas! 🎄",
5     "from_me": false
6   },
7   {
8     "msg": "Thank you, you too! 😊",
9     "ref": "Merry Christmas! 🎄",
10    "from_me": true
11  }
```

```

11 },
12 //...
13 ]

```

It then produces the following chat:

2026-12-25 09:41

Hey! ❤️

Merry Christmas! 🎄 ❤️

Merry Christmas! 🎄

Thank you! 🙏

You too! 😊 ❤️

2027-01-01 02:01

Hey!

Happy New Year! 🎉

Thanks!

Happy New Year! 🎉 📺

II.d Basic example directly in Typst

```

1  #let my-messages = (
2    (
3      date: "2024-01-01T12:00:00",
4      msg: "This is defined directly in the Typst file.",
5      from_me: false,
6    ),
7    (
8      msg: "Nice!",
9      from_me: true,
10   ),
11 )
12 #messeji(chat-data: my-messages)

```

Produces the following chat:

2024-01-01 12:00

This is defined directly in the Typst file.

Nice!

II.e Messages with Images

As typst does not support directory walking¹, loading messages with images from a JSON file is a bit more complicated. Additionally, Typst packages cannot access the working directory of your project, which means that you have to define the function that loads the images, and pass them to `messeji` as parameters. However, there is still a way to do this automatically.

In short, you just have to add a single function to your Typst document that handles loading the images from your directory. The general workflow looks like this:

1. Save all images in the same directory (e.g., `img`)
2. Import and use the `get-image-names` function from `messeji` to get all image names that are defined in your message list. It returns the image names in a dictionary, and still has to be filled with the loaded images.
3. Pass the image names to your own `load-images` function (see below) to load them into Typst.
4. Pass the loaded images to `messeji`.

In detail, add the following code to your document:

```
1  #import "@preview/messeji:0.3.0": messeji, get-image-names // import
   image name function typ
2
3  // this function has to be defined in your own document, as it accesses files
4  // located in your project directory (with the `image()` function).
5  #let load-images(
6    directory, // with trailing slash!
7    image-names,
8  ) = {
9    for img-name in image-names.keys() {
10      image-names.insert(img-name, image(directory + img-name, fit: "contain"))
11    }
12    return image-names
13  }
14
15  #let chat-with-images = json("image.example.json") // Load chat data from
   JSON file
16
17  #let image-names = get-image-names(chat-with-images) // Load image names from
   loaded chat data
18
19  #let loaded-images = load-images("img/", image-names) // Load actual images
   into Typst
20
21  #messeji(
22    chat-data: chat-with-images,
23    images: loaded-images,
24  )
```

This produces the following chat:

¹<https://forum.typst.app/t/is-there-a-way-to-retrieve-the-current-file-name-list-files-etc-within-typst/155/3>

2025-04-02 12:15

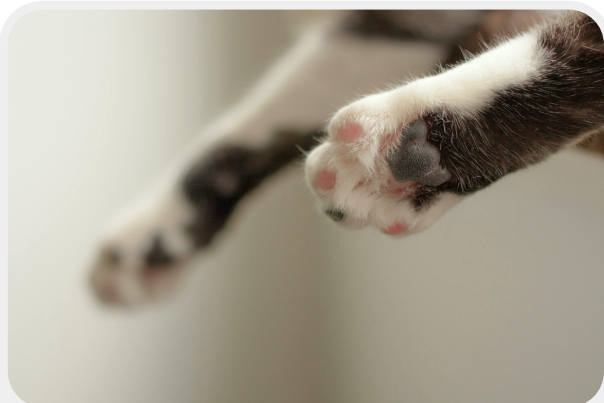
Hey!



Look at this cute cat!

Cute right?

Wow, very cute!



Look at these cute toebeans!

(Image)

Cute!

III CUSTOMIZATION

III.a Themes

You can customize the text colors, backgrounds, and font sizes. Currently, the default theme has the following keys and values:

```
1 #let default-theme = (typ
2   timestamp-color: rgb("#65676b"),
3   timestamp-size: 0.9em,
4   date-changed-color: luma(0%),
5   date-changed-size: 1.5em,
6   me-right: (
7     color: luma(100%),
8     background-color: rgb("#3797f0"),
9     quote-color: luma(0%),
10    quote-background-color: rgb("#efefef"),
11  ),
12  other-left: (
13    color: luma(0%),
14    background-color: rgb("#efefef"),
15    quote-color: luma(0%),
16    quote-background-color: rgb("#efefef"),
17  ),
18  reaction-stroke-color: luma(100%),
19  bubble-inset: 0.6em,
20  bubble-radius: 1.5em,
21  bubble-tail: true,
22  image-width: 50%,
23 )
```

If you want to change the theme, you just have to override the keys that you need. Everything else that is undefined will be taken from the default theme:

```
1 #let custom-theme = (typ
2   me-right: (
3     background-color: green
4   )
5 )
6 #messeji(
7   chat-data: parsed-data,
8   theme: custom-theme
9 )
```

Produces the following output:

2026-12-25 09:41

Hey!



Merry Christmas! 🎄



Merry Christmas! 🎄

Thank you!

You too! 😊



2027-01-01 02:01

Happy New Year! 🎉



Hey!

Happy New Year! 🎉

Thanks!

III.b Custom timestamp and date-change format

By default, every time a message has a `date` value, timestamps are displayed in the format `YYYY-MM-DD HH:DD`. However, you can customize it by passing a format string to `timestamp-format` ([Click here for Typst documentation](#)).

If you want to highlight that a new day started, you can use `date-changed-format`. By default, this is deactivated by setting it to `""`.

```
1 #messeji(  
2   chat-data: parsed-data,  
3   date-changed-format: "[year]/[month]/[day]",  
4   timestamp-format: "[hour]:[minute]",  
5 )
```

2026/12/25

09:41

Hey!



Merry Christmas! 🎄



Merry Christmas! 🎄

Thank you!

You too! 😊



2027/01/01

02:01

Happy New Year! 🎉



Hey!

Happy New Year! 🎉

Thanks!

IV FUNCTION REFERENCE

- [get-image-names\(\)](#)
- [messeji\(\)](#)

get-image-names

Helper function to get image names that are present in the chat for further processing.

Parameters

```
get-image-names(chat-data: array) -> dictionary
```

chat-data array

Array of messages.

messeji

Main function to render chat history.

Parameters

```
messeji(  
    chat-data: array,  
    date-changed-format: str,  
    timestamp-format: str,  
    images: dictionary,  
    theme: dictionary  
)
```

chat-data array

Array of messages.

Default: []

date-changed-format str

Date format when the day changes.

Default: none

timestamp-format str

Timestamp format every time a message contains a `date` value.

Default: "[year]-[month]-[day] [hour]:[minute]"

images dictionary

Image dictionary. Key is filename, value is loaded image. Empty version with filenames can be generated with `get-image-names`, but has to be filled with respective `image`s for every filename before passing it to `messeji`.

Default: (:)

theme dictionary

Theme dictionary. If value is not set, the default value is used.

Default: (:)