messēji

MIT License 2025 Kai Anter

Contents

Note: Build this document with `typst compile main.typ --root ..`

Introduction

messēji is a Typst package for typesetting chat histories in a modern, minimal design, inspired by popular messengers.

Main features of *messēji* include:

- Support for quoted messages
- Displaying timestamps between messages
- Simple data model to read from external files (JSON, YAML, …) if you want to typeset very long chat histories

Currently, it does not support:

- Group chats (only 1-on-1 chats)
- Reactions to messages
- Images

These issues will likely be fixed in a future release.

Usage

Import

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

```
#import "@preview/messeji:0.2.0": messeji
```

Message Structure

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

```
{
  // Optional, can be combined with image
  "msg": "Actual Message",

  // Optional, can be combined with msg
  "image": "path_to_image.jpg"

  // Required, true for right, false for left
  "from_me": true,

  // Optional, in ISO 8601 format
  "date": "2026-12-25T09:41:00",

  // Optional
  "ref": "Previous message that is being quoted",
}
```

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.

Basic example from JSON

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

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

```
[
  {
    "date": "2026-12-25T09:41:00",
    "msg": "Merry Christmas! 🎄",
    "from_me": false
  },
  {
    "msg": "Thank you, you too! 😊",
    "ref": "Merry Christmas! 🎄",
    "from_me": true
  },
  //...
]
```

It then produces the following chat:

<div align="center">2026-12-25 09:41</div>

Hey!

Merry Christmas! 🎄

Merry Christmas! 🎄

Thank you, you too! 😊

<div align="center">2026-12-25 12:15</div>

Thanks!

<div align="center">2027-01-01 02:01</div>

Happy New Year! 🎉

Happy New Year! 🎉

Thanks!

Basic example directly in Typst

```
#let my-messages = (
  (
    date: "2024-01-01T12:00:00",
    msg: "This is defined directly in the Typst file.",
    from_me: false,
  ),
  (
    msg: "Nice!",
    from_me: true,
  ),
)
#messeji(chat-data: my-messages)
```

Produces the following chat:

<div align="center">2024-01-01 12:00</div>

This is defined directly in the Typst file.

Nice!

Messages with Images

As typst does not support directory walking[1], 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:

```
#import "@preview/messeji...": messeji, get-image-names // import image name function

// this function has to be defined in your own document, as it accesses files
// located in your project directory (with the `image()` function).
#let load-images(
  directory, // with trailing slash!
  image-names,
) = {
  for img-name in image-names.keys() {
    image-names.insert(img-name, image(directory + img-name, fit: "contain"))
  }
  return image-names
}

#let chat-with-images = json("image.example.json") // Load chat data from JSON file

#let image-names = get-image-names(chat-with-images) // Load image names from loaded
chat data

#let loaded-images = load-images("img/", image-names) // Load actual images into
Typst

#messeji(
  chat-data: chat-with-images,
  images: loaded-images,
)
```
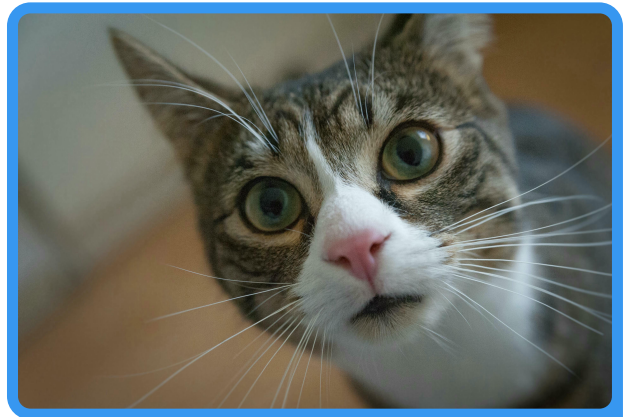
This produces the following chat:
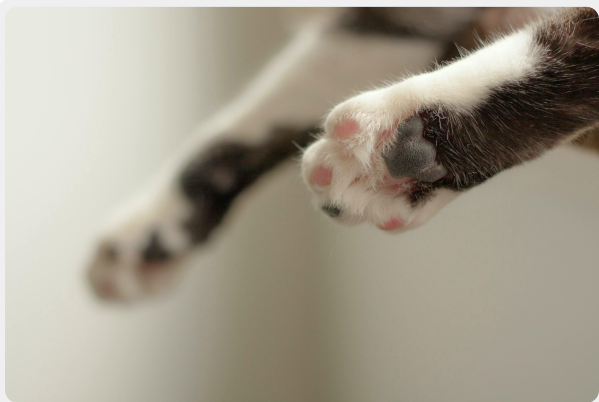
2025-04-02 12:15

Hey!

---

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

Look at this cute cat!

Cute right?

Wow, very cute!

Look at these cute toebeans!

(Image)

Cute!

Customization

Themes

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

```
(
  timestamp-color: rgb("#65676b"),
  timestamp-size: 0.9em,
  date-changed-color: luma(0%),
  date-changed-size: 1.5em,
  me-right: (
    color: luma(100%),
    background-color: rgb("#3797f0"),
    quote-color: luma(0%),
    quote-background-color: rgb("#efefef"),
```

```
  ),
  other-left: (
    color: luma(0%),
    background-color: rgb("#efefef"),
    quote-color: luma(0%),
    quote-background-color: rgb("#efefef"),
  ),
)
```

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:

```
#let custom-theme = (
  me-right: (
    background-color: green
  )
)
#messeji(
  chat-data: parsed-data,
  theme: custom-theme
)
```

Produces the following output:

<div align="center">2026-12-25 09:41</div>

Hey!

Merry Christmas! 🎄

<div align="right">Merry Christmas! 🎄</div>

<div align="right">Thank you, you too! 😊</div>

<div align="center">2026-12-25 12:15</div>

Thanks!

<div align="center">2027-01-01 02:01</div>

<div align="right">Happy New Year! 🎉</div>

Happy New Year! 🎉

Thanks!

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 `""`.

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

2026/12/25

09:41

Hey!

Merry Christmas! 🎄

Merry Christmas! 🎄

Thank you, you too! 😊

12:15

Thanks!

2027/01/01

02:01

Happy New Year! 🎉

Happy New Year! 🎉

Thanks!