Search...

Getting started
Components

M4Q
About
Population
Constructor
Ajax
Animation
Loops
Visibility
Effects
Subtree functions
Attributes
Html, text and value
Css and classes
Position and size
Manipulation
DataSet
Events
Utils

Base
Containers
Grid system
Typography
Tables
Forms
Buttons
Images
Figures
Lists

Form components
Checkbox
File input
Input
Input material
Keypad
Rating
Radio
Select
Slider
Double Slider
Spinner
Switch
Tag input
Textarea

Menus
App bar
Bottom navigation
Bottom sheet
Menu
Ribbon Menu
Side bar
Side navigation

Controls
Accordion
Badge
Carousel
Cards
Cube
Counter
Charms
Chat
Donut
Image compare
Image magnifier
Gravatar
List
ListView
Master
NavView
Panels
Progress & Activity
Streamer
Stepper
Splitter
Tabs
Tabs material
Table
Tiles
TreeView
Wizard

Information
Dialogs
Info box
Hints
Notify system
Popovers
Toasts
Windows

Date & time
Calendar
Calendar picker
Date picker
Time picker
Countdown
Formatting date

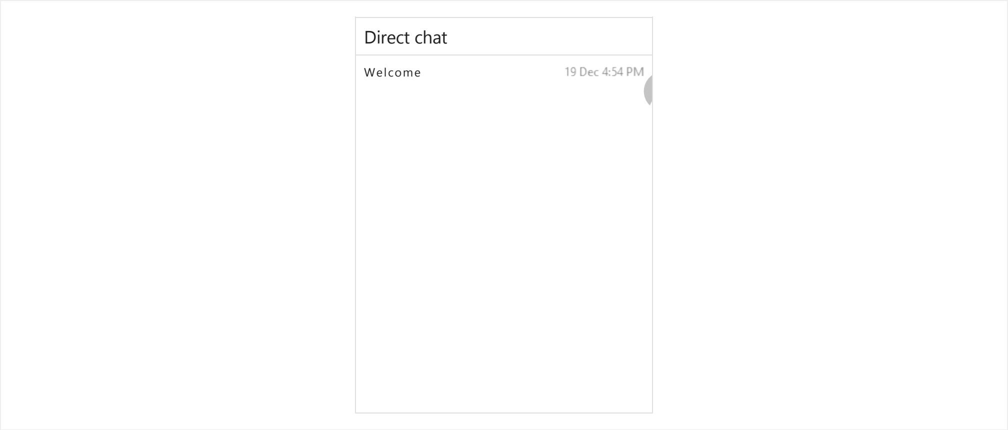
Media

Chat

Metro 4 implements component to create simple chat.

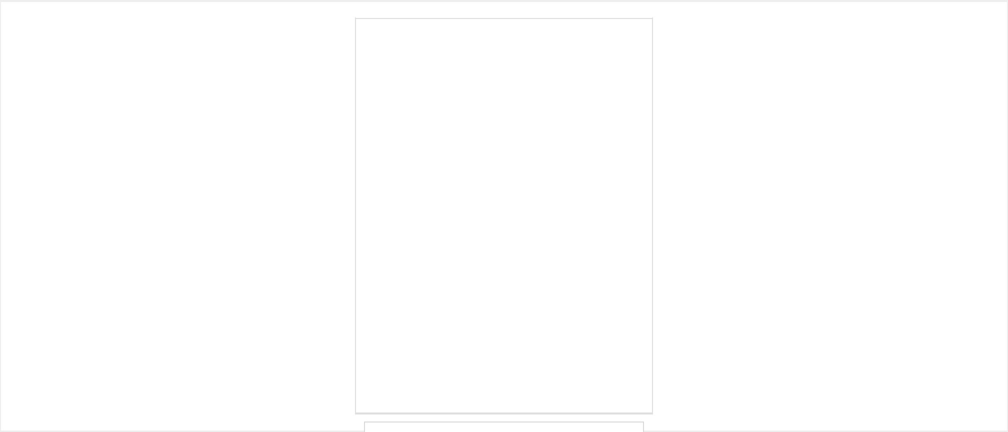
About

In Metro 4 you can create simple chat.



Create chat

To create simple interface for your chat you must create `container` with role `chat` and set size roy you chat. That's all.



```
<div
  data-role="chat"
  data-width="300"
  data-height="400"
></div>
```

Options

Option	Data-*	Default	Desc
`timeFormat`	`data-time-format`	%d %b %l:%M %p	Message time format. More about date-time formats
`inputTimeFormat`	`data-input-time-format`	%m-%d-%y	Message input time format. More about date-time formats
`name`	`data-name`	John Doe	Receiver name
`avatar`	`data-avatar`		Receiver avatar. Must be string with image source (used for img src attribute)
`welcome`	`data-welcome`	null	Welcome message
`title`	`data-title`	null	Chat title
`width`	`data-width`	100%	Chat width
`height`	`data-height`	auto	Chat height
`randomColor`	`data-random-color`	false	If true, messages colorized with random color from predefined colors
`messages`	`data-messages`	null	Object with initialization messages

Events

Event	Data-*	Desc
`onMessage(msg)`	`data-on-message`	Fired when message added to chat
`onSend(msg)`	`data-on-send`	Fired when user click on Send button

Video player
Audio player
Tools
Collapse
Color module
Draggable
Dropdown
Form validator
Hotkeys
Micro templates
Ripple
Storage
Session storage
Sorter
Touch and swipe
Utilities
Animations
Additional

Event	Data-*	Desc
`onChatCreate`	`data-on-chat-create`	Fired when chat was created

Methods

Method	Desc
`clear()`	Clear all message
`add(msg)`	Add message to chat
`addMessages(messages)`	Add messages to chat
`delMessage(msg_id)`	Del message from chat
`updMessage(msg)`	Update message on chat

Customize

You can customize chat with special attributes.

Attr	Data-*	Desc
`clsChat`	`data-cls-chat`	Class for chat component
`clsName`	`data-cls-name`	Class for sender or receiver name
`clsTime`	`data-cls-time`	Class for message time
`clsInput`	`data-cls-input`	Class for chat input control
`clsSendButton`	`data-cls-send-button`	Class for send button
`clsMessageLeft`	`data-cls-message-left`	Class for left message
`clsMessageRight`	`data-cls-message-right`	Class for right message

Message structure

Chat message is a plain object.

```
message = {
    id,
    name,
    time,
    avatar,
    position
}
```

Property	Type	Desc
`id`	`any`	Message unique id
`name`	`string`	Sender name
`time`	valid ECMA script datetime string	Message time
`avatar`	`uri`	Sender avatar, used for img src attribute
`position`	`string`	left or right