React & React Native > Comments > Styling

Styling the Social Comment Section

The social-style comment section requires a style configuration object to properly style its UI. This object holds all the values needed for styling various elements of the comment section, ensuring flexibility and adaptability to your app's design needs.

Using the default configuration is straightforward, and for most use cases, it provides a good balance of aesthetics and usability. However, you can also customize it to align perfectly with your app's visual identity.

Getting the Style Configuration

To use the social comment section, you must pass a `styleConfig` object to the `useSocialComments` hook. You can obtain this configuration object using the `useSocialStyle` hook.

Default Configuration

For a basic setup, you can use the default configuration by simply calling `useSocialStyle` without any arguments:

```
const styleConfig = useSocialStyle();
```

Pass this `styleConfig` to the `useSocialComments` hook along with the `entityId` and any other optional properties.

Custom Configuration

If you wish to customize the styles, you can pass a partial configuration object to the `useSocialStyle` hook. Only the styles you want to change need to be included in the object.

Here's an example:

```
import React, { useMemo } from "react";
import { useSocialStyle, UseSocialStyleProps } from "@replyke/comments-social-react-nativ
```

```
const customStyles = useMemo<Partial<UseSocialStyleProps>>(
  () => ({
    newCommentFormProps: {
      verticalPadding: 24,
      paddingLeft: 24,
      paddingRight: 24,
    },
  }),
  []
);

const styleConfig = useSocialStyle(customStyles);
```

The returned `styleConfig` will include your customizations while retaining the default values for all other properties.

Style Configuration Object Interface

The style configuration object has the following structure:

```
export type SocialStyleConfig = {
  commentFeedProps: CommentFeedStyleProps;
  commentProps: CommentStyleProps;
  newCommentFormProps: NewCommentFormStyleProps;
};
```

Each section below details the properties available for customization:

CommentFeedStyleProps

Styles the comment feed container:

```
export type CommentFeedStyleProps = {
  backgroundColor: string;  // Background color of the comment feed.
  commentsGap: number;      // Space between individual comments.
};
```

CommentStyleProps

Styles individual comments:

```
export interface CommentStyleProps {
  horizonalItemsGap: number;        // Horizontal gap between items.
  verticalItemsGap: number;         // Vertical gap between items.
  authorAvatarSize: number;         // Size of the author’s avatar.
  authorFontSize: number;           // Font size for the author’s name.
  authorFontWeight: FontWeight;     // Font weight for the author’s name.
  authorFontColor: string;          // Font color for the author’s name.
  fromNowFontSize: number;          // Font size for the "time ago" text.
  fromNowFontColor: string;         // Font color for the "time ago" text.
  commentBodyFontSize: number;      // Font size for the comment text.
  commentBodyFontColor: string;     // Font color for the comment text.
  actionsItemGap: number;           // Gap between action items (e.g., like, reply).
  replyButtonFontSize: number;      // Font size for the reply button.
  replyButtonFontWeight: FontWeight;// Font weight for the reply button.
  replyButtonFontColor: string;     // Font color for the reply button.
  heartIconSize: number;            // Size of the heart (like) icon.
  heartIconFullColor: string;       // Color for a filled heart icon.
  heartIconEmptyColor: string;      // Color for an empty heart icon.
  heartIconPaddingBottom: number;   // Padding below the heart icon.
  likesCountFontSize: number;       // Font size for the likes count.
  likesCountFontWeight: FontWeight; // Font weight for the likes count.
  likesCountFontColor: string;      // Font color for the likes count.
  viewRepliesPaddingTop: number;    // Padding above the "view replies" text.
  viewMoreRepliesText: string;      // Text for "view more replies" action.
  hideRepliesText: string;          // Text for "hide replies" action.
  repliesGap: number;               // Gap between replies.
  repliesPaddingTop: number;        // Padding above the replies section.
  justNowText: string;              // Text to display for very recent comments.
}
```

NewCommentFormStyleProps

Styles the new comment input form:

```
export type NewCommentFormStyleProps = {
  backgroundColor: string;          // Background color of the form.
  withAvatar: boolean;              // Whether to show the user’s avatar.
  itemsGap: number;                 // Gap between items in the form.
  verticalPadding: number;          // Vertical padding inside the form.
  paddingLeft: number;              // Left padding inside the form.
  paddingRight: number;             // Right padding inside the form.
  authorAvatarSize: number;         // Size of the user’s avatar.
  placeholderText: string;          // Placeholder text for the input.
  textareaTextSize: number;         // Font size of the textarea text.
  postButtonText: string;           // Text on the post button.
  postButtonFontSize: number;       // Font size for the post button text.
  postButtonFontWeight: FontWeight; // Font weight for the post button text.
  postButtonFontColor: string;      // Font color for the post button text.
};
```

FontWeight and BorderStyles

You can use the following types for font weights and border styles:

```
export type FontWeight =
  | 100
  | 200
  | 300
  | 400
  | 500
  | 600
  | 700
  | 800
  | 900
  | "100"
  | "200"
  | "300"
  | "400"
  | "500"
  | "600"
  | "700"
  | "800"
  | "900";

export type BorderStyles =
  | "solid"
  | "dashed"
  | "dotted"
  | "double"
  | "groove"
  | "ridge"
  | "inset"
  | "outset"
  | "none"
  | "hidden";
```

Conclusion

The `useSocialStyle` hook provides an intuitive way to configure the styles for your comment section. Whether you stick with the default styles or create a completely custom design, this configuration system ensures your comment section fits seamlessly into your app. Customize as needed and create an engaging experience for your users.

Last updated on May 6, 2025