

Hello 🙌

Welcome to Avvoka's editor task! This task forms the first of two rounds for Avvoka's interview process (the second round is just a chat, don't worry). Please read through the instructions below and respond to us at your earliest convenience

Background to the task

Please ensure your code meets the following requirements:

JavaScript ESNext (or TypeScript)

Use only Quill, Delta, and Parchment libraries; no other frameworks or libraries are allowed

Do not use external resources, except for Quill documentation and Mozilla Developer Network (MDN) documentation

Do not use AI for code generation

The requirements are stringent due to the nature of our projects, which often involve unique challenges and solutions not readily available on the internet. As such, it is crucial that the code is crafted independently and originates from your expertise.

Objective

Part 1

Add a custom module to the Quill editor that enables the insertion of a new format containing an image and text. The format should appear as follows:

```
```html

<q-img-text>

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
</q-img-text>

```
```

The Delta format representation should be:

```
```json

{
 "insert": {
 "qImage": {
 "image": "https://picsum.photos/200/300",
 "text": "Lorem ipsum dolor sit amet, consectetur adipiscing elit."
 }
 }
}

```
```

The custom format should be insertable into the editor using the toolbar and via drag and drop. For drag and drop, users should be able to drag an image into the editor and add text. Using the toolbar, users should be able to insert both the image and text simultaneously. Drag and Drop should convert the image to a base64 format, enabling it to be saved in the Delta format without server-side storage. The custom format must be fundamentally insertable through Delta (e.g., using `editor.updateContents(new Delta().retain(..10..).insert({...}))`).

Part 2

The second part of the task requires modifying the qImage format to display a number in the "title" attribute, indicating how many other qImage elements are above it. If no qImage elements are above it, the title attribute should remain empty. When another qImage is added, all existing qImage elements must be updated to reflect the correct order. This also applies when removing a qImage.

```
```html
```

```
... (no q-img-text above this)
```

```
<q-img-text>
```

```

```

```
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
</q-img-text>
```

```
... (there is a q-img-text above this one, so title="1")
```

```
<q-img-text>
```

```

```

```
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

```
</q-img-text>
```

```
...
```

```
...
```

Your code should efficiently handle recalculations for up to 10,000 qImage elements in a single document.

## How to get back in touch

When you're finished, email us at [engineer-hiring@avvoka.com](mailto:engineer-hiring@avvoka.com) to talk us through what you've come up with 😊