

GABRIELE ROMANATO

Come creare un componente Vue.js per valutare la forza di una password

In questa guida realizzerai un componente Vue 3 che analizza la forza di una password in tempo reale. Imparerai a usare la Composition API, le props, il reattività e le classi dinamiche per fornire all'utente un feedback visivo e testuale mentre digita la password.

1. La logica del componente

Cominciamo definendo tutta la logica dentro al blocco `<script setup>`. Questo ci permette di scrivere codice più semplice e diretto, sfruttando la Composition API di Vue 3.

```
import { ref, computed, watch, onMounted } from 'vue';
```

Importiamo gli strumenti principali di Vue:

- `ref` serve per creare variabili reattive
- `computed` per proprietà calcolate
- `watch` per reagire ai cambiamenti
- `onMounted` per eseguire codice quando il componente è pronto

```
const props = defineProps({  
  password: {  
    type: String,  
    required: true,  
  },  
});
```

Il componente riceve una prop chiamata `password`. Deve essere una stringa e deve essere obbligatoriamente fornita dal genitore. Questa sarà la password da valutare.

```
const strength = ref(0);
```

`strength` è il punteggio della password. Partiamo da 0 e lo aggiorneremo in base ai criteri di sicurezza.

```
const checkStrength = () => {  
  const pass = props.password;  
  let score = 0;  
  
  if (!pass) {  
    strength.value = 0;  
    return;  
  }  
  if (pass.length >= 8) score++;  
  if (/[A-Z]/.test(pass)) score++;  
  if (/[a-z]/.test(pass)) score++;  
  if (/[0-9]/.test(pass)) score++;  
}
```

```
    if (/[\W_]/.test(pass)) score++;

    strength.value = score;
  };
```

Questa funzione controlla la forza della password e aggiorna il punteggio. I criteri sono:

- Minimo 8 caratteri
- Lettere maiuscole
- Lettere minuscole
- Numeri
- Simboli o caratteri speciali

Ogni criterio soddisfatto aggiunge 1 punto al punteggio totale (massimo 5).

```
const strengthLabel = computed(() => {
  switch (strength.value) {
    case 0:
      return '';
    case 1:
    case 2:
      return 'Debole';
    case 3:
      return 'Media';
    case 4:
      return 'Buona';
    case 5:
      return 'Forte';
    default:
      return '';
  }
});
```

In base al punteggio, questa proprietà calcolata restituisce un'etichetta che descrive la qualità della password: da “Debole” fino a “Forte”.

```
const strengthClass = computed(() => {
  return {
    none: strength.value === 0,
    weak: strength.value > 0 && strength.value <= 2,
    medium: strength.value === 3,
    good: strength.value === 4,
    strong: strength.value === 5,
  };
});
```

Qui costruiamo un oggetto di classi CSS che useremo per colorare dinamicamente la barra. Solo una classe sarà vera alla volta, in base al punteggio.

```
onMounted(checkStrength);
watch(() => props.password, checkStrength);
```

Quando il componente è pronto (onMounted) o la prop password cambia (watch), ricalcoliamo la forza della password.

2. Il template HTML

Ora creiamo la parte visiva del componente. Mostriamo una barra e una label. Entrambe reagiscono in tempo reale alla password digitata.

```
<div class="password-strength" :class="{ inactive: !props.password }">
  <div class="strength-bar" :class="strengthClass"></div>
  <p class="strength-label">{{ strengthLabel }}</p>
</div>
```

Il contenitore ha la classe `inactive` se la password è vuota. La barra cambia aspetto in base alla classe dinamica `strengthClass`. Il testo mostra la valutazione attuale.

3. Lo stile visivo

Infine, aggiungiamo uno stile semplice ma efficace per rendere tutto più chiaro all'utente.

```
.password-strength {
  width: 100%;
  margin-top: 0.6rem;
}

.inactive {
  display: none !important;
}

.strength-bar {
  height: 8px;
  border-radius: 4px;
  transition: background-color 0.3s;
  width: 0%;
}

.strength-bar.none {
  background-color: transparent;
}

.strength-bar.weak {
  background-color: #d00;
  width: 25%;
}

.strength-bar.medium {
  background-color: orange;
  width: 50%;
}

.strength-bar.good {
  background-color: yellowgreen;
  width: 75%;
}

.strength-bar.strong {
  background-color: green;
  width: 100%;
}
```

```
.strength-label {  
  margin-top: 0.25rem;  
}
```

La barra cresce in larghezza e cambia colore in base alla forza. Con una password forte, è verde e al 100%; se è debole, è rossa e corta. Il tutto è fluido grazie alla `transition`.

Conclusione

Ora hai un componente riutilizzabile per mostrare la forza di una password in qualsiasi parte della tua applicazione Vue 3. Puoi collegarlo a un campo di input, migliorare lo stile o renderlo accessibile. È un ottimo esempio pratico per capire props, reattività e classi dinamiche in Vue.

Applicazioni Correlate



-

JavaScript Password Mask

Un esempio in JavaScript di mascheramento di una password con l'aggiunta della funzionalità di copia negli appunti.

[Docker](#)[Docker Compose](#)[JavaScript](#)



Python Placeholder Image

Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.

Docker Docker Compose Python Flask JavaScript



-

Go Placeholder Image

Applicazione in Go per la creazione di immagini segnaposto.

[Docker](#)[Docker Compose](#)[Go](#)[JavaScript](#)