

If you want to **set the initial state of an Angular Material <mat-button-toggle> from a variable without using a <mat-button-toggle-group>**, you can simply bind the checked property to your variable.

Here's a **complete runnable example**:

HTML

```
<mat-button-toggle  
  [checked]="isToggled"  
  (change)="onToggleChange($event)">  
  Toggle Me  
</mat-button-toggle>
```

TypeScript (Component)

```
import { Component } from '@angular/core';  
import { MatButtonToggleChange } from '@angular/material/button-toggle';  
  
@Component({  
  selector: 'app-toggle-example',  
  templateUrl: './toggle-example.component.html',  
  styleUrls: ['./toggle-example.component.css']  
})  
export class ToggleExampleComponent {  
  // Initial value from a variable
```

```
isToggled = true; // or false, depending on your logic

// Handle toggle change
onToggleChange(event: MatButtonToggleChange) {
  this.isToggled = event.source.checked;
  console.log('Toggle state:', this.isToggled);
}

}
```

Key Points

- [checked] Binding
 - Directly binds the toggle's checked state to your variable.
 - The variable can be set dynamically (e.g., from API data).
- (change) Event
 - Captures when the user changes the toggle state.
 - Updates your variable to keep it in sync.
- No <mat-button-toggle-group> Needed
 - This works for a single toggle without grouping.

✓ Example Behavior:

If `isToggled = true` in the component, the toggle will be **on** when the page loads.

If you later set `this.isToggled = false`, Angular will automatically update the UI.

If you want, I can also show you **how to set this value dynamically after an async API call** so the toggle updates when data

arrives.

Do you want me to include that?