super_tooltip: One-Page App Summary

What It Is

A Flutter package that renders highly customizable tooltips as overlay entries, beyond the built-in Tooltip behavior. It supports configurable placement, custom bubble shapes, barrier and blur effects, and controller-driven show/hide lifecycle.

Who It’s For

Primary persona: Flutter developers building mobile, web, or desktop apps who need guided overlays, product hints, or tutorial callouts with fine-grained control.

What It Does (Key Features)

- Displays tooltip content in screen overlay using OverlayEntry with fade animation.
- Supports four directions (up/down/left/right) and dynamic popupDirectionBuilder.
- Customizes bubble visuals: arrow geometry, border, radius, shadows, decorationBuilder.
- Adds barrier options: color, tap-to-dismiss behavior, and optional blur filter.
- Provides touch-through cutout area (oval/rectangle) for spotlight guidance flows.
- Exposes controller API: showTooltip/hideTooltip, visibility state, onShow/onHide.
- Supports desktop and web hover interactions (showOnHover/hideOnHoverExit).

How It Works (Repo-Evidence Architecture)

- UI trigger layer: SuperTooltip wraps child with GestureDetector/MouseRegion.
- State/control: SuperTooltipController (ChangeNotifier + Completer) emits show/hide events.
- Overlay composition: creates blur entry, barrier entry, and bubble entry in order.
- Positioning/layout: ToolTipPositionDelegate + SuperUtils compute constraints/placement.
- Rendering: BubbleShape draws arrowed bubble; ShapeOverlay draws barrier cutout.
- Services/storage/network: Not found in repo.

How To Run (Minimal Getting Started)

1. Install package: flutter pub add super_tooltip
2. Import: import ’package:super_tooltip/super_tooltip.dart’;
3. In a StatefulWidget, create SuperTooltipController and pass it to SuperTooltip.
4. Trigger tooltip with controller.showTooltip() and controller.hideTooltip().
5. Run the repo example app: cd example && flutter run
6. Platform/setup matrix details: Not found in repo.