📝 Simplified Guide: Angular Content Projection

The article is about **Angular Projection**, which is a way to insert content (HTML, elements, text) from one component into another. It's done with `ng-content` .

✨ Why Use Content Projection?

Let's say you want to build two forms:

- A **Signup form**
- A **Login form**

Both are almost the same — only the **heading** and **button text** are different.

👉 A common (but inefficient) approach: create two separate components.
👉 A better approach: create **one component** (e.g., `auth-component`) and just "project" the different parts (like heading and button) into it.

That's where **ng-content** helps.

📌 Example 1: Simple Content Projection

`auth-component.html`

```
<div class="container">
  <form action="">
    <ng-content></ng-content>
    <input type="text" placeholder="Firstname" />
    <input type="text" placeholder="Lastname" />
  </form>
  <button type="submit">Send!</button>
</div>
```

Here, `ng-content` is a placeholder. Whatever you put between `<app-auth> ... </app-auth>` will appear there.

Usage in `app.component.html`:

```
<div class="container">
  <div class="signup">
    <app-auth>
      <h2>Signup here</h2>
    </app-auth>
  </div>

  <div class="login">
    <app-auth>
      <h2>Login here</h2>
    </app-auth>
  </div>
</div>
```

💡 Result:

- The first `<app-auth>` shows “Signup here” as heading.
- The second `<app-auth>` shows “Login here.”

So one component handles both cases.

📌 Example 2: Projection into Selected Slots

What if you pass **multiple elements** (like heading + button)?
How do you place them at different places inside your component?

That’s where **projection slots** come in.

You can have multiple `ng-content` tags, each with a `select` attribute.

`auth-component.html`:

```
<div class="container">
  <form action="">
    <ng-content select="h2"></ng-content>
    <input type="text" placeholder="Firstname" />
    <input type="text" placeholder="Lastname"/>
  </form>
  <button type="submit">Send!</button>
  <ng-content select="button"></ng-content>
</div>
```

Here we made **two slots**:

- One for `<h2>` (the heading)
- One for `<button>`

Usage in `app.component.html`:

```
<div class="container">
  <div class="signup">
    <app-auth>
      <h2>Signup here</h2>
      <button type="submit">Sign Up!</button>
    </app-auth>
  </div>

  <div class="login">
    <app-auth>
      <h2>Login here</h2>
      <button type="submit">Login!</button>
    </app-auth>
  </div>
</div>
```

💡 Result:

- “Signup here” goes into the first slot (`ng-content select="h2"`)
- “Sign Up!” button goes into the second slot (`ng-content select="button"`)
- Same for the Login version.

⚡ Note: `select` works like a **CSS selector** → you can target by tag name, class, or ID.

📖 Summary

- **Content Projection** lets you design one reusable component and insert external HTML into it.
- You can use `ng-content` for basic projection.
- You can use **multiple `ng-content` with `select`** to control *where specific elements go* (projection slots).
- This keeps components flexible, clean, and avoids duplication.

🔮 What's Next?

The author mentions the next article will cover **creating and destroying components dynamically** in Angular.

✅ So in simpler terms:
Content projection = placeholder slots inside a component where you can "inject" custom HTML.