

 HONORIS UNITED UNIVERSITIES	Année Universitaire : 2025-2026 
<h2>Workshop n°10 :</h2> <h3>Manipulation des composants Standalone</h3>	

Les objectifs :

- Comprendre et appliquer les composants Standalone
- Comprendre la différence entre Standalone Components et NgModule-based components.
- Savoir créer, importer et réutiliser un composant standalone dans différents modules.

Les instructions à suivre :

- 1- Dans le composant HomeComponent, nous souhaitons afficher les **3 événements les plus populaires** selon le nombre de likes .
 - a. Déclarer une variable liste dans le composant Home et y récupérer la liste complète des événements à partir du service.
 - b. Appliquer un **filtre ou un tri** pour n'afficher que les **3 événements ayant le plus grand nombre de likes**.
 - c. Utiliser le composant app-event-card (déjà déclaré dans le module Events) pour afficher ces 3 événements.
 - d. **Question :** Que constatez-vous ?
- 2- Créer un composant **standalone** nommé CardComponent dans le dossier layout/.

Commande à exécuter : `ng g c layout/card --standalone=true --skip-tests`

```
~\Desktop\ESPRIT\Cours\cours Esprit 2026\AngularTraining git:[main]
ng g c layout/card --standalone=true --skip-tests
CREATE src/app/layout/card/card.component.html (20 bytes)
CREATE src/app/layout/card/card.component.ts (238 bytes)
CREATE src/app/layout/card/card.component.css (0 bytes)
```

Figure: le terminal de l'excusion de la commande : `ng g c layout/card --standalone=true --skip-tests`

```

1 import {Component} from '@angular/core';
2
3 @Component({
4   selector: 'app-card',
5   standalone: true,
6   imports: [
7     ],
8   templateUrl: './card.component.html',
9   styleUrls: ['./card.component.css']
10 })
11 export class CardComponent {}
12

```

Figure : le fichier card.component.ts

3- Répondre à la question 1 en utilisant le composant Card créé

- Importer** le composant standalone CardComponent (layout/card) dans le composant Home (puisque'il est standalone, on l'importe directement dans le @Component de Home).
- Utiliser** le composant app-card dans le template du composant Home avec *ngFor pour afficher la liste des événements (Top 3).
- Reprendre** le même HTML et la même logique que le composant EventCard précédemment créé dans le module Events (mise en forme Bootstrap, affichage du titre, lieu, prix, date, etc.).
- Utiliser** la propriété @Input() dans le composant CardComponent pour recevoir et afficher correctement l'objet event.

4- Remplacer l'ancien composant de carte d'événement

- Supprimer le composant EventCard créé dans le module Events.
- Utiliser à sa place le composant **standalone** CardComponent partout où une carte d'événement est nécessaire (liste, détails, etc.).

1- Activité noté : Gestion Feedback

- Sous le dossier Features, créer le module feedback avec le lazy loading
ng g module features/feedback --route feedback --module app.module
- Sous le dossier models/, créer la classe modèle Feedback avec les attributs :: id, userID, eventID, content, date, rate: number
- Dans le module Feedback, créer et manipuler un composant FormFeedback permettant d'**ajouter un commentaire** sur un événement.
- Utiliser **Reactive Forms** ou **Template Driven Forms** pour ajouter les **contrôles de saisie nécessaires** (validation du contenu, note obligatoire, etc.).
- Ajouter un bouton “**Add comment**” dans l’interface EventDetail permettant de rediriger l’utilisateur vers l’interface FormFeedback en récupérant l’id de l’événement dans l’URL (route paramétrée)

The screenshot shows two browser windows of the EventHub application. The top window displays the 'Liste des événements' (Event List) page with three event cards: 'Angular' (100 places), 'Latave' (50 places), and 'Symfony' (50 places). The bottom window shows the 'Detail Event' page for the 'Angular Workshop' event, which is described as 'From Beginner to Pro, Creating Applications for the Real World'. It features a purple-themed illustration of two people working at laptops. The event details include 'Date: November 10, 2025', 'Lieu: Tunis', 'Price: €50.00', and 'Places disponibles: 25'. A red box highlights the 'Ajouter Feedback' button on the right side of the detail page.

- f. Après avoir rempli le formulaire, un clic sur le bouton “Save feedback” doit permettre de persister le feedback côté serveur (appel HTTP vers l’API / backend).

The screenshot shows the 'Gestion des Feedbacks' (Feedback Management) interface. It includes a form for adding a new feedback with fields for 'Contenu' (Content) and 'Note (1 - 5)' (Rating), and a large blue 'Ajouter Feedback' (Add Feedback) button. Below the form is a table titled 'Liste des Feedbacks' (List of Feedbacks) with columns: Contenu, Note, Date, and Actions. A single row is shown with the content 'tset', rating '0', date '12/1/25, 1:57 PM', and buttons for 'Edit' and 'Delete'. A red arrow points from the 'Ajouter Feedback' button down to the 'Liste des Feedbacks' table.

- g. Dans cette même interface (FormFeedback), ajouter un bloc “Suggestions” affichant la liste des événements dont la localisation est la même que celle de l’événement sélectionné (suggestion d’événements similaires))

localhost:62304/feedbacks/1

EventHub

Accueil Événements Mes Tickets Feedback Se connecter

Gestion des Feedbacks

Ajouter un Feedback

Contenu

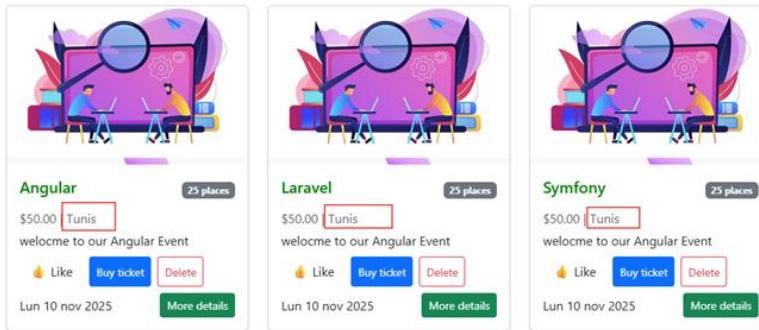
Note (1 - 5)

0

Ajouter Feedback

Liste des Feedbacks

Contenu	Note	Date	Actions
tset	0	12/1/25, 1:57 PM	Edit Delete



Explication :

Commande	Rôle
ng g c layout/card --standalone=true --skip-tests	Créer un composant standalone

correction : <https://github.com/badi3a/AngularTraining>