

[CGA-JAVA-WFDA] Web Front-end Development Angular 2.1

[Dashboard](#) / [My courses](#) / [CGA-JAVA-WFDA-2.1](#) / [4. Angular Component & Template](#) / [\[Thực hành\] Tạo ứng dụng Hackernews](#)

[Thực hành] Tạo ứng dụng Hackernews

Mục tiêu

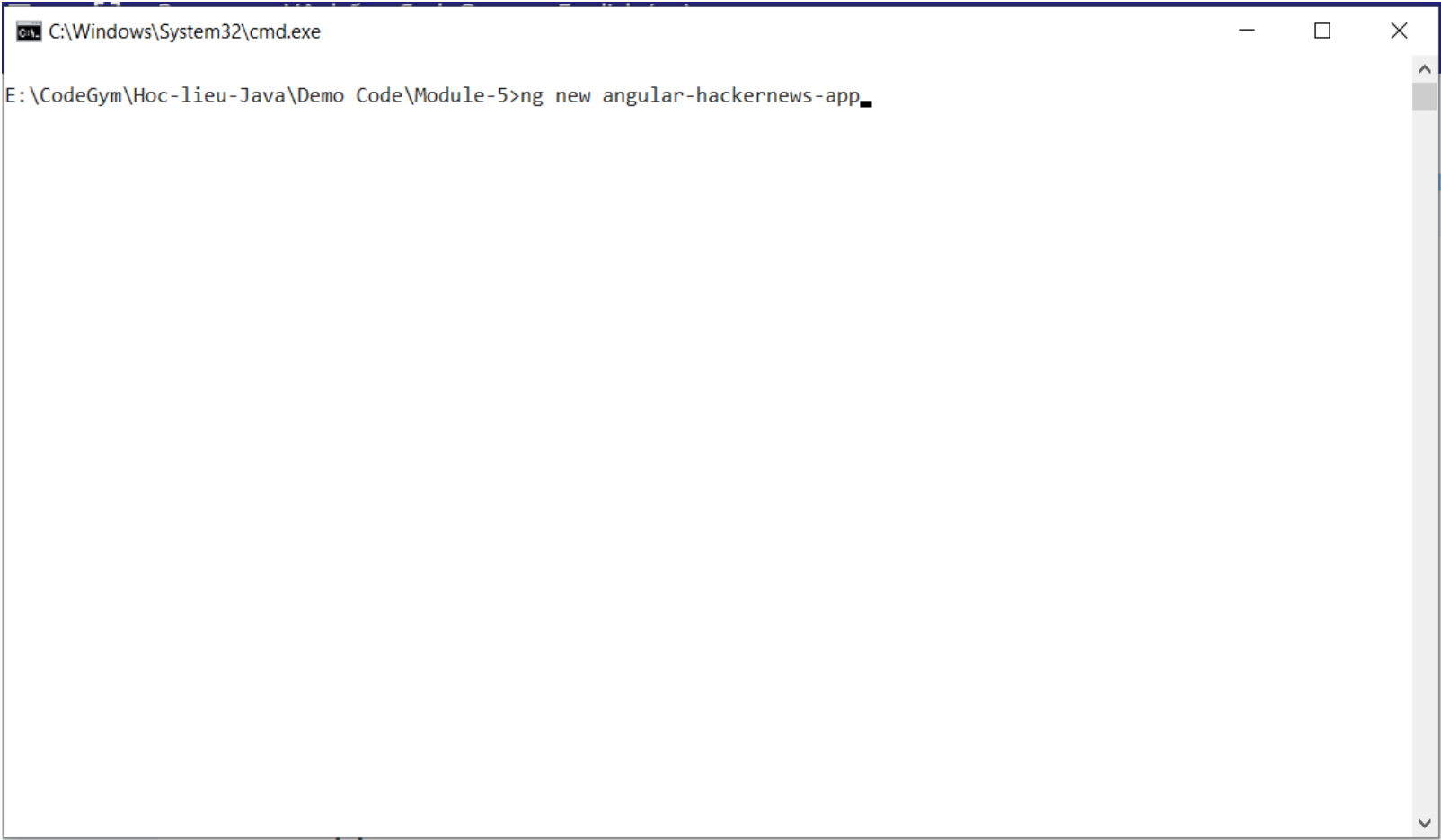
Thực hành tạo component và thao tác với template syntax, binding, directives.

Mô tả

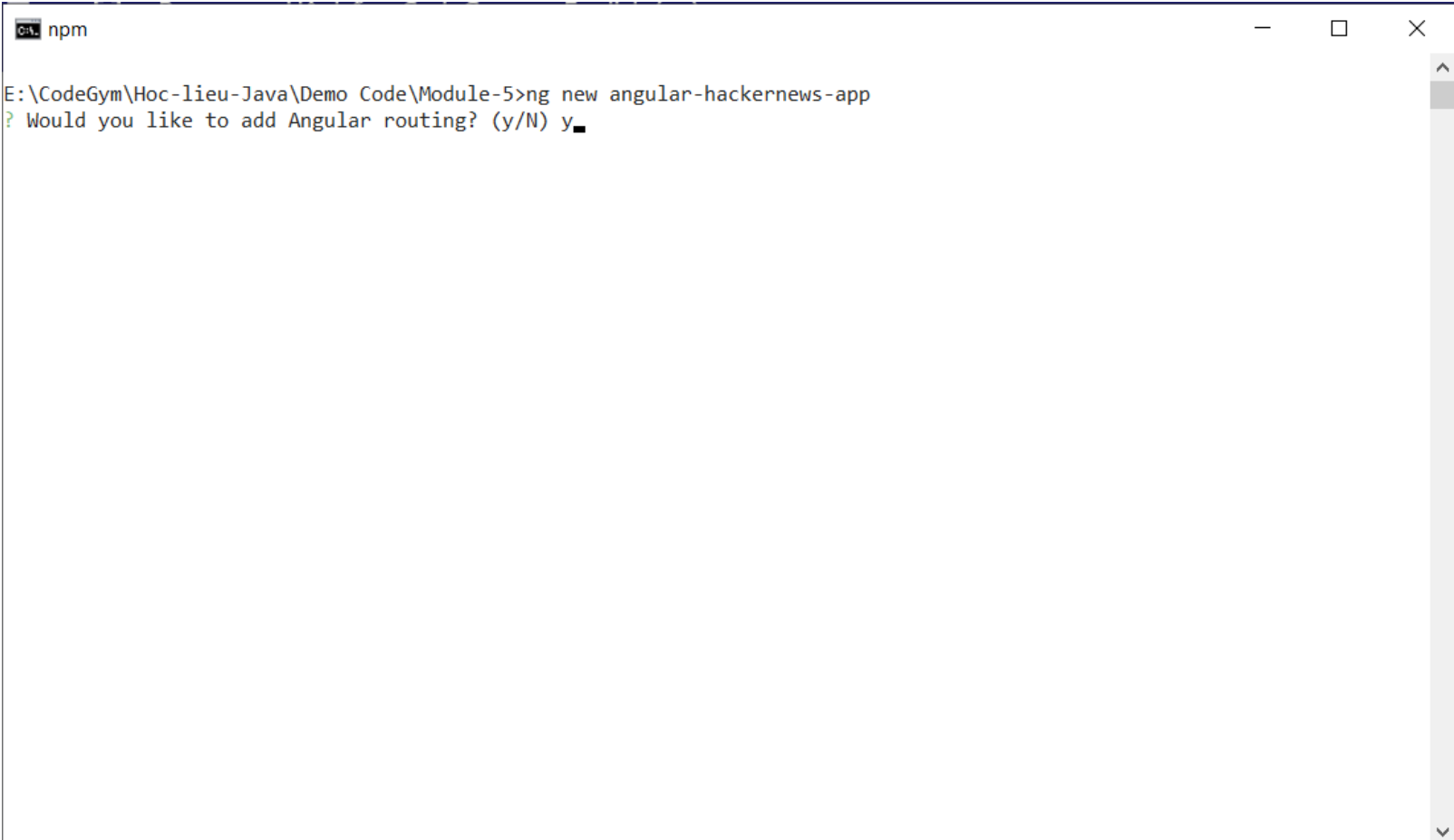
Tạo một ứng dụng mô phỏng <https://news.ycombinator.com> - một trang tổng hợp tin tức cho lập trình viên. Mỗi bản tin sẽ bao gồm một tiêu đề và một đường dẫn đến nguồn. Ngoài ra còn có thông số ngày đăng và số lượt vote.

Hướng dẫn

Bước 1: Mở cửa sổ terminal và gõ lệnh **ng new angular-hackernews-app** để tạo mới một dự án Angular



Tại cửa sổ tiếp theo gõ chữ **y** và ấn enter



Tiếp theo chọn CSS và ấn enter và chờ ứng dụng được khởi tạo

tiếp theo chọn CSS và div editor và cho ứng dụng được khởi tạo

```
C:\Windows\System32\cmd.exe
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.component.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/app/app.module.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.prod.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/environments/environment.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/polyfills.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/styles.css.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test.ts.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.app.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tsconfig.spec.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in tslint.json.
The file will have its original line endings in your working directory
Successfully initialized git.

E:\CodeGym\Hoc-lieu-Java\Demo Code\Module-5>
```

Ứng dụng đã được tạo thành công

Bước 2: Mở dự án vừa tạo bằng Webstorm

Bước 3: Mở cửa sổ terminal trong Webstorm và gõ lệnh **ng g c article** để tạo một component mới trong dự án

Bước 4: Tạo một interface có tên là Article bằng câu lệnh **ng g i article** và sau đó thêm đoạn code sau vào trong file article.component.ts:

```
export interface Article {  
  title?: string;  
  url?: string;  
}
```

Bước 5: Hiển thị một article lên màn hình chính bằng cách khai báo một đối tượng article trong file article.component.ts như sau:

```
import { Component, OnInit } from '@angular/core';
import { Article } from '../article';

@Component({
  selector: 'app-article',
  templateUrl: './article.component.html',
  styleUrls: ['./article.component.css']
})
export class ArticleComponent implements OnInit {
  article: Article = {
    title: 'The Evolution of Async JavaScript: From Callbacks, to Promises, to Async/Await',
    url: 'https://medium.freecodecamp.org/the-evolution-of-async-javascript-from-callbacks-to-promises-to-async-await-e73b047f2f40'
  };
  constructor() {}

  ngOnInit() {}
}
```

Và sau đó chúng ta sử dụng đoạn mã sau để hiển thị thông tin của article đó bên file `article.component.html`

```
<h2>Today's article</h2>
<hr>
<div>
  <a href="{{article.url}}" target="_blank">{{article.title}}</a>
</div>
<hr>
@CodeGym, 2021
```

Bước 6: sử dụng thẻ <app-article></app-article> trong file app.component.html để hiện thị thông tin của component article:

```
<app-article></app-article>
```

Bước 7: Trong file article.component.html, thêm thẻ textarea như sau để chỉnh sửa thông tin của article:

```
<div>
  <h2>Article Information</h2>
  <table>
    <tr>
      <td>Title:</td>
      <td><textarea [(ngModel)]="article.title" cols="70"></textarea></td>
    </tr>
    <tr>
      <td>Url:</td>
      <td>
        <textarea [(ngModel)]="article.url" cols="70"></textarea>
      </td>
    </tr>
  </table>
</div>
```

Đoạn mã đầy đủ của file article.component.html lúc này sẽ như sau:

```
<h2>Today's article</h2>
<hr>
<div>
  <a href="{{article.url}}" target="_blank">{{article.title}}</a>
</div>
<hr>
<div>
  <h2>Article Information</h2>
  <table>
    <tr>
      <td>Title:</td>
      <td><textarea [(ngModel)]="article.title" cols="70"></textarea></td>
    </tr>
    <tr>
      <td>Url:</td>
      <td>
        <textarea [(ngModel)]="article.url" cols="70"></textarea>
      </td>
    </tr>
  </table>
</div>
<hr>
@CodeGym, 2021
```

Bước 8: Bổ sung dữ liệu về một danh sách các article trong file article.component.ts như sau:

```
articles: Article[] = [
  {
    title: 'The Evolution of Async JavaScript: From Callbacks, to Promises, to Async/Await',
    url: 'https://medium.freecodecamp.org/the-evolution-of-async-javascript-from-callbacks-to-promises-to-async-await-e73b047f2f40'
  },
  {
    title: 'Game of Life',
    url: 'https://thefullsnack.com/posts/game-of-life.html'
  },
  {
    title: 'Nguyên tắc thiết kế REST API',
    url: 'https://medium.com/eway/nguyen-tack-thiet-ke-rest-api-23add16968d7'
  },
  {
    title: 'Why You Only Need to Test with 5 Users',
    url: 'https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/'
  },
  {
    title: 'Let’s Build A Web Server. Part 1.',
    url: 'https://ruslanspivak.com/lsbaws-part1/'
  }
];
```

Đoạn mã đầy của article.component.ts sẽ như sau:

```
import {Component, OnInit} from '@angular/core';
import {Article} from '../article';

@Component({
  selector: 'app-article',
  templateUrl: './article.component.html',
  styleUrls: ['./article.component.css']
})
export class ArticleComponent implements OnInit {
  article: Article = {};
  articles: Article[] = [
    {
      title: 'The Evolution of Async JavaScript: From Callbacks, to Promises, to Async/Await',
      url: 'https://medium.freecodecamp.org/the-evolution-of-async-javascript-from-callbacks-to-promises-to-async-await-e73b047f2f40'
    },
    {
      title: 'Game of Life',
      url: 'https://thefullsnack.com/posts/game-of-life.html'
    },
    {
      title: 'Nguyên tắc thiết kế REST API',
      url: 'https://medium.com/eway/nguyên-tắc-thiết-kế-rest-api-23add16968d7'
    },
    {
      title: 'Why You Only Need to Test with 5 Users',
      url: 'https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/'
    },
    {
      title: 'Let’s Build A Web Server. Part 1.',
      url: 'https://ruslanspivak.com/lsbaws-part1/'
    }
  ];

  constructor() {}

  ngOnInit() {}
}
```

Cập nhật lại thông tin hiển thị danh sách article bên file article.component.html bằng cách sử dụng *ngFor để duyệt toàn bộ các article có trong mảng articles:

```
<div *ngFor="let art of articles">
  <a href="{{art.url}}" target="_blank">{{art.title}}</a>
</div>
```

Bước 9: Bổ sung tính năng thêm article vào danh sách bằng cách tạo một phương thức addNewArticle() và thêm article vào trong mảng như sau:

```
addNewArticle() {
  this.articles.push(this.article);
}
```

Bổ sung button vào file template và cài đặt sự kiện click như sau:

```
<div>
  <h2>Article Information</h2>
  <table>
    <tr>
      <td>Title:</td>
      <td><textarea [(ngModel)]="article.title" cols="70"></textarea></td>
    </tr>
    <tr>
      <td>Url:</td>
      <td>
        <textarea [(ngModel)]="article.url" cols="70"></textarea>
      </td>
    </tr>
    <tr>
      <td></td>
      <td><button style="float: right" (click)="addNewArticle()">Add</button></td>
    </tr>
  </table>
</div>
```

Bước 10: Bổ sung tính năng hiển thị số lượt thích

Tạo component để hiển thị số lượt like bằng cách sử dụng lệnh sau: ng g c like

Khai báo thuộc tính like thể hiện số lượt like trong file like.component.ts và phương thức likeThis() để tăng số lượt like của mỗi article:

```
import {Component, OnInit} from '@angular/core';

@Component({
  selector: 'app-like',
  templateUrl: './like.component.html',
  styleUrls: ['./like.component.css']
})
export class LikeComponent implements OnInit {
  like = 0;

  constructor() {}

  ngOnInit() {}

  likeThis() {
    this.like++;
  }
}
```

Trong file like.component.html, thêm đoạn code như sau:

```
{{like}} likes. <button (click)="likeThis()">Like</button>
```

Trong file article.component.html sửa lại đoạn code thành như sau:

```
<div *ngFor="let art of articles">
  <a href="{{art.url}}" target="_blank">{{art.title}}</a>: <app-like></app-like>
</div>
```

Checkout sang nhánh develop để xem mã nguồn tham khảo: <https://github.com/codegym-vn/wfda-angular-hackernews-app>

Hướng dẫn nộp bài:

up bài tập lên github

paste link github vào phần nộp bài

Submission status

Submission status	No attempt
Grading status	Not graded
Last modified	-
Submission comments	<div>▶ Comments (0)</div>

Add submission

You have not made a submission yet.

Previous Activity

Jump to...

Next Activity

CHƯƠNG TRÌNH

- Career
- Premium
- Accelerator

TÀI NGUYÊN

- Blog
- Tạp chí Lập trình
- AgileBreakfast

Follow Us

