INF 354

Example 2022

Typescript:

ihero.ts interface:

export interface Ihero

{

id: number;

name: string;

image: string;

}

hero.ts file:

import { Ihero } from "./ihero";

type allowNull = number | string | null;

export class Hero implements Ihero

{

id!: number;

name!: string;

age!: allowNull;

birthday!: allowNull;

height!: string;

image!: string;

alive!: boolean;

}

Angular:

chatstats.component.html:

<table class="table">

<thead>

<tr>

<th>Stat</th>

<th>Value</th>

</tr>

</thead>

<tbody>

<tr>

<td># Msgs from bot</td>

<td>{{ messages.filtered( message => message.isBotAuthored).length }}</td>

</tr>

<tr>

<td>># Msgs from user</td>

<td>{{ messages.filtered( message => !message.isBotAuthored).length }}</td>

</tr>

<tr>

<td># Msgs exceeding 10 words</td>

<td>{{ messages.filter(message => message.text.split(' ').length > 10).length }}</td>

</tr>

</tbody>

</table>

chatpopup.component.ts:

onAddNewMessage(): void {

if (this.newMessageText.trim() !== '') {

const newMessage: ChatMessage = {

text: this.newMessageText.trim(),

isBotAuthored: false,

timestamp: new Date()

};

this.chatMessageManagerService.AddMessage(newMessage);

this.newMessageText = '';

}

}

chatstats.component.ts:

export class ChatstatsComponent implements OnInit {

botMessagesCount: number = 0;

userMessagesCount: number = 0;

longMessagesCount: number = 0;

constructor(private chatMessageManagerService: ChatMessageManagerService) { }

ngOnInit(): void {

this.chatMessageManagerService.messageAdded.subscribe((newMessage: ChatMessage) => {

if (newMessage.isBotAuthored) {

this.botMessagesCount++;

} else {

this.userMessagesCount++;

}

const words = newMessage.text.split(' ');

if (words.length > 10) {

this.longMessagesCount++;

}

});

}

}

Example 2021:

Web API and Angular:

CourseController.cs:

[HttpGet]

[Route("GetCourseCode/{courseCode}")]

public async Task<IActionResult> GetCourseCode(string courseCode)

{

try

{

var course = await \_courseRepository.GetCourseByCourseCode(courseCode);

if (course == null)

{

return NotFound(400, “Not Found”);

}

return Ok(course);

}

catch (Exception)

{

return StatusCode(500, “Internal service error”);

}

}

course-details.component.ts: