

Month: Dec-18

Question 1:

Tags: javascript, setstate, typescript, lodash, reactjs

Question: Problem I am writing an application in React and was unable to avoid a super common pitfall, which is calling `setState(...)` after `componentWillUnmount(...)`. I looked very carefully at my code and tried to put some guarding clauses in place, but the problem persisted and I am still observing the warning. Therefore, I've got two questions: How do I figure out from the stack trace, which particular component and event handler or lifecycle hook is responsible for the rule violation? Well, how to fix the problem itself, because my code was written with this pitfall in mind and is already trying to prevent it, but some underlying component's still generating the warning. Browser console

```
Warning: Can't perform a React state update on an unmounted component. This
is a no-op, but it indicates a memory leak in your application. To fix,
cancel all subscriptions and asynchronous tasks in the componentWillUnmount
method. in TextLayerInternal (created by Context.Consumer) in TextLayer
(created by PageInternal) index.js:1446 d/console[e] index.js:1446
warningWithoutStack react-dom.development.js:520 warnAboutUpdateOnUnmounted
react-dom.development.js:18238 scheduleWork react-dom.development.js:19684
enqueueSetState react-dom.development.js:12936
./node_modules/react/cjs/react.development.js/Component.prototype.setState
react.development.js:356 _callee$ TextLayer.js:97 tryCatch runtime.js:63
invoke runtime.js:282 defineIteratorMethods/prototype[method]
runtime.js:116 asyncGeneratorStep asyncToGenerator.js:3 _throw
asyncToGenerator.js:29 Code Book.tsx import { throttle } from 'lodash'; import
* as React from 'react'; import { AutoWidthPdf } from
'../shared/AutoWidthPdf'; import BookCommandPanel from
'../shared/BookCommandPanel'; import BookTextPath from
'../static/pdf/sde.pdf'; import './Book.css'; const DEFAULT_WIDTH = 140;
class Book extends React.Component { setDivSizeThrottleable: () => void;
pdfWrapper: HTMLDivElement | null = null; isComponentMounted: boolean =
false; state = { hidden: true, pdfWidth: DEFAULT_WIDTH, };
constructor(props: any) { super(props); this.setDivSizeThrottleable =
throttle(() => { if (this.isComponentMounted) { this.setState({ pdfWidth:
this.pdfWrapper!.getBoundingClientRect().width - 5, }); } }, 500, ); }
componentDidMount = () => { this.isComponentMounted = true;
this.setDivSizeThrottleable(); window.addEventListener("resize",
this.setDivSizeThrottleable); }; componentWillUnmount = () => {
this.isComponentMounted = false; window.removeEventListener("resize",
this.setDivSizeThrottleable); }; render = () => ( <div className="Book"> {
this.state.hidden && <div className="Book__LoadNotification centered">Book
is being loaded...</div> } <div
className={this.getPdfContentContainerClassName()}> <BookCommandPanel
bookTextPath={BookTextPath} /> <div className="Book__PdfContent" ref={ref =>
this.pdfWrapper = ref}> <AutoWidthPdf file={BookTextPath}
width={this.state.pdfWidth} onLoadSuccess={(_: any) =>
this.onDocumentComplete()} /> </div> <BookCommandPanel
bookTextPath={BookTextPath} /> </div> </div> );
getPdfContentContainerClassName = () => this.state.hidden ? 'hidden' : '';
onDocumentComplete = () => { try { this.setState({ hidden: false });
this.setDivSizeThrottleable(); } catch (caughtError) { console.warn({
caughtError }); } }; } export default Book; AutoWidthPdf.tsx import * as React
from 'react'; import { Document, Page, pdfjs } from 'react-pdf';
```

```

pdfjs.GlobalWorkerOptions.workerSrc =
`//cdnjs.cloudflare.com/ajax/libs/pdf.js/${pdfjs.version}/pdf.worker.js`;
interface IProps { file: string; width: number; onLoadSuccess: (pdf: any) =>
void; } export class AutoWidthPdf extends React.Component<IProps> { render =
() => ( <Document file={this.props.file} onLoadSuccess={(_: any) =>
this.props.onLoadSuccess(_)} > <Page pageNumber={1} width={this.props.width}
/> </Document> ); } Update 1: Cancel throttleable function (still no luck) const
DEFAULT_WIDTH = 140; class Book extends React.Component {
setDivSizeThrottleable: (((() => void) & Cancelable) | undefined; pdfWrapper:
HTMLDivElement | null = null; state = { hidden: true, pdfWidth:
DEFAULT_WIDTH, }; componentDidMount = () => { this.setDivSizeThrottleable =
throttle( () => { this.setState({ pdfWidth:
this.pdfWrapper!.getBoundingClientRect().width - 5, }); }, 500, );
this.setDivSizeThrottleable(); window.addEventListener("resize",
this.setDivSizeThrottleable); }; componentWillUnmount = () => {
window.removeEventListener("resize", this.setDivSizeThrottleable!);
this.setDivSizeThrottleable!.cancel(); this.setDivSizeThrottleable =
undefined; }; render = () => ( <div className="Book"> { this.state.hidden &&
<div className="Book__LoadNotification centered">Book is being
loaded...</div> } <div className={this.getPdfContentContainerClassName()}>
<BookCommandPanel BookTextPath={BookTextPath} /> <div
className="Book__PdfContent" ref={ref => this.pdfWrapper = ref}>
<AutoWidthPdf file={BookTextPath} width={this.state.pdfWidth}
onLoadSuccess={(_: any) => this.onDocumentComplete()} /> </div>
<BookCommandPanel BookTextPath={BookTextPath} /> </div> </div> );
getPdfContentContainerClassName = () => this.state.hidden ? 'hidden' : '';
onDocumentComplete = () => { try { this.setState({ hidden: false });
this.setDivSizeThrottleable!(); } catch (caughtError) { console.warn({
caughtError }); } }; } export default Book;

```

Answer Score: 470

Stack Overflow answer, Verb Count: 64 , **Adjective Count:** 38

Stack Overflow Answer: Here is a React Hooks specific solution for Error Warning: Can't perform a React state update on an unmounted component. Solution You can declare `let isMounted = true` inside `useEffect`, which will be changed in the cleanup callback, as soon as the component is unmounted. Before state updates, you now check this variable conditionally: `useEffect(() => {`

```

let isMounted = true; // note mutable flag someAsyncOperation().then(data =>
{ if (isMounted) setState(data); // add conditional check }) return () => {
isMounted = false }; // cleanup toggles value, if unmounted }, []); //
adjust dependencies to your needs const Parent = () => { const [mounted,
setMounted] = useState(true); return ( <div> Parent: <button onClick={() =>
setMounted(!mounted)}> {mounted ? "Unmount" : "Mount"} Child </button>
{mounted && <Child />} <p> Unmount Child, while it is still loading. It
won't set state later on, so no error is triggered. </p> </div> ); }; const
Child = () => { const [state, setState] = useState("loading (4 sec)...");
useEffect(() => { let isMounted = true; fetchData(); return () => {
isMounted = false; }; // simulate some Web API fetching function fetchData()
{ setTimeout(() => { // drop "if (isMounted)" to trigger error again //
(take IDE, doesn't work with stack snippet) if (isMounted) setState("data
fetched") else console.log("aborted setState on unmounted component") },
4000); } }, []); return <div>Child: {state}</div>; };
ReactDOM.render(<Parent />, document.getElementById("root")); <script src="h

```

```

https://cdnjs.cloudflare.com/ajax/libs/react/16.13.0/umd/react.production.min
.js" integrity="sha256-32Gmw5rBDXyMjg/73FgpkotZdMrxuYW7tj8adbN8z4="
crossorigin="anonymous"></script> <script src="https://cdnjs.cloudflare.com/
ajax/libs/react-dom/16.13.0/umd/react-dom.production.min.js"
integrity="sha256-bjQ42ac3EN0GqK40pC9gGi/YixvKyZ24qMP/9HiGW7w="
crossorigin="anonymous"></script> <div id="root"></div> <script>var {
useReducer, useEffect, useState, useRef } = React</script> Extension: Custom
useAsync Hook We can encapsulate all the boilerplate into a custom Hook, that automatically aborts
async functions in case the component unmounts or dependency values have changed before:
function useAsync(asyncFn, onSuccess) { useEffect(() => { let isActive =
true; asyncFn().then(data => { if (isActive) onSuccess(data); }); return ()
=> { isActive = false }; }, [asyncFn, onSuccess]); } // custom Hook for
automatic abortion on unmount or dependency change // You might add
onFailure for promise errors as well. function useAsync(asyncFn, onSuccess)
{ useEffect(() => { let isActive = true; asyncFn().then(data => { if
(isActive) onSuccess(data) else console.log("aborted setState on unmounted
component") }); return () => { isActive = false; }; }, [asyncFn,
onSuccess]); } const Child = () => { const [state, setState] =
useState("loading (4 sec)..."); useAsync(simulateFetchData, setState);
return <div>Child: {state}</div>; }; const Parent = () => { const [mounted,
setMounted] = useState(true); return ( <div> Parent: <button onClick={() =>
setMounted(!mounted)}> {mounted ? "Unmount" : "Mount"} Child </button>
{mounted && <Child />} <p> Unmount Child, while it is still loading. It
won't set state later on, so no error is triggered. </p> </div> ); }; const
simulateFetchData = () => new Promise( resolve => setTimeout(() =>
resolve("data fetched"), 4000)); ReactDOM.render(<Parent />,
document.getElementById("root")); <script src="https://cdnjs.cloudflare.com/
ajax/libs/react/16.13.0/umd/react.production.min.js"
integrity="sha256-32Gmw5rBDXyMjg/73FgpkotZdMrxuYW7tj8adbN8z4="
crossorigin="anonymous"></script> <script src="https://cdnjs.cloudflare.com/
ajax/libs/react-dom/16.13.0/umd/react-dom.production.min.js"
integrity="sha256-bjQ42ac3EN0GqK40pC9gGi/YixvKyZ24qMP/9HiGW7w="
crossorigin="anonymous"></script> <div id="root"></div> <script>var {
useReducer, useEffect, useState, useRef } = React</script> More on effect
cleanups: Overreacted: A Complete Guide to useEffect

```

GPT answer, Verb Count: 51 , Adjective Count: 37

GPT-4o Answer: Solution 1. Identifying the source of the warning: To determine which component or lifecycle hook is causing the issue, carefully examine the stack trace in the console warning. In your example, the stack trace indicates that the warning originates from `TextLayer.js`, specifically the line: `_callee$ TextLayer.js:97` This suggests that the `TextLayerInternal` component (likely part of the React-PDF library) is invoking `setState` after the component has unmounted. Verify any asynchronous operations, event listeners, or state updates within this component that might not be properly cleaned up in its `componentWillUnmount` method. Additionally, you can use tools like React DevTools or add logging to suspect components to trace the issue further.

2. Fixing the issue: Here are steps to address the problem:

- Guard State Updates:** Ensure every `setState` invocation is wrapped with a check for the component's mounted status. You have already implemented this with the `isComponentMounted` flag in the `Book` component. Apply a similar approach to any child components that might cause the warning, particularly in `TextLayer` and `AutoWidthPdf`.
- Cancel Asynchronous Tasks:** Confirm that all asynchronous tasks, such as throttled functions, API calls, or event listeners, are correctly canceled or cleaned up in `componentWillUnmount`. Your updated code already cancels the throttled function: `this.setDivSizeThrottleable!.cancel();` Make sure similar precautions are applied in `TextLayerInternal` or other library components.
- Review**

Library Components: Since the warning originates from a library (React-PDF), ensure you are using the latest version of the library. Check the documentation and issues for React-PDF for any known bugs or updates addressing this specific problem. d. Use Functional Components with Hooks: If feasible, refactor your components to use React hooks like `useEffect` and `useRef`. Hooks simplify cleanup logic and reduce the likelihood of this issue:

```
import React, { useEffect, useRef, useState } from 'react'; const Book = () => { const [pdfWidth, setPdfWidth] = useState(DEFAULT_WIDTH); const pdfWrapper = useRef(null); useEffect(() => { const updateWidth = throttle(() => { if (pdfWrapper.current) { setPdfWidth(pdfWrapper.current.getBoundingClientRect().width - 5); } }, 500); updateWidth(); window.addEventListener('resize', updateWidth); return () => { window.removeEventListener('resize', updateWidth); updateWidth.cancel(); }; }, []); return ( ); }; 
```

 This ensures proper cleanup of event listeners and throttled functions.

Co-Pilot, Verb Count: 47 , Adjective Count: 28

Co-Pilot Answer: To determine which component and event handler or lifecycle hook is responsible for the rule violation from the stack trace, you need to carefully analyze the stack trace provided in the browser console. Specifically, look at the sequence of function calls leading up to the warning. In this case, the warning indicates that a state update is attempted on an unmounted component: `Warning: Can't perform a React state update on an unmounted component.` The stack trace shows the sequence of function calls: `in TextLayerInternal (created by Context.Consumer) in TextLayer (created by PageInternal) warningWithoutStack @ react-dom.development.js:520 warnAboutUpdateOnUnmounted @ react-dom.development.js:18238 scheduleWork @ react-dom.development.js:19684 enqueueSetState @ react-dom.development.js:12936 ./node_modules/react/cjs/react.development.js/Component.prototype.setState @ react.development.js:356 _callee$ @ TextLayer.js:97 tryCatch @ runtime.js:63 invoke @ runtime.js:282 defineIteratorMethods/` From this, you can identify that the `TextLayerInternal` component, which is created by `Context.Consumer`, is attempting to update its state after it has been unmounted. This is likely due to a state update occurring in an asynchronous operation that has not been properly cleaned up. To fix the issue, ensure that any asynchronous operations, subscriptions, or timeouts are properly canceled or cleaned up when the component is unmounted. For the provided code, this can be done by checking the component's mounted status before calling `setState` and canceling any ongoing operations in the `componentWillUnmount` method. Here's an example of how you can modify the `Book` component to address the issue:

```
import { throttle } from 'lodash'; import * as React from 'react'; import { AutoWidthPdf } from '../shared/AutoWidthPdf'; import BookCommandPanel from '../shared/BookCommandPanel'; import BookTextPath from '../static/pdf/sde.pdf'; import './Book.css'; const DEFAULT_WIDTH = 140; class Book extends React.Component { setDivSizeThrottleable: () => void; pdfWrapper: HTMLDivElement | null = null; isComponentMounted: boolean = false; state = { hidden: true, pdfWidth: DEFAULT_WIDTH, }; constructor(props: any) { super(props); this.setDivSizeThrottleable = throttle( () => { if (this.isComponentMounted) { this.setState({ pdfWidth: this.pdfWrapper!.getBoundingClientRect().width - 5, }); } }, 500, ); componentDidMount = () => { this.isComponentMounted = true; this.setDivSizeThrottleable(); window.addEventListener("resize", this.setDivSizeThrottleable); }; componentWillUnmount = () => { this.isComponentMounted = false; window.removeEventListener("resize", this.setDivSizeThrottleable); }; render = () => ( <div className="Book"> { this.state.hidden && <div className="Book__LoadNotification centered">Book is being loaded...</div> } <div className={this.getPdfContentContainerClassName()}> <BookCommandPanel
```

```

bookTextPath={BookTextPath} /> <div className="Book__PdfContent" ref={ref =>
this.pdfWrapper = ref}> <AutoWidthPdf file={BookTextPath}
width={this.state.pdfWidth} onLoadSuccess={(_: any) =>
this.onDocumentComplete()} /> </div> <BookCommandPanel
bookTextPath={BookTextPath} /> </div> </div> );
getPdfContentContainerClassName = () => this.state.hidden ? 'hidden' : '';
onDocumentComplete = () => { try { this.setState({ hidden: false });
this.setDivSizeThrottleable(); } catch (caughtError) { console.warn({
caughtError }); } }; } export default Book;

```

Question 2:

Tags: python, slice, multi-index, dataframe, pandas

Question: What are the most common pandas ways to select/filter rows of a dataframe whose index is a MultiIndex? Slicing based on a single value/label Slicing based on multiple labels from one or more levels Filtering on boolean conditions and expressions Which methods are applicable in what circumstances Assumptions for simplicity: input dataframe does not have duplicate index keys input dataframe below only has two levels. (Most solutions shown here generalize to N levels) Example input: `mux = pd.MultiIndex.from_arrays([list('aaaabbbbccddddd'), list('tuvwttuvwttuvw')], names=['one', 'two'])` `df = pd.DataFrame({'col': np.arange(len(mux))}, mux)`

col	one	two
0	a	t
1	u	v
2	w	3
3	b	t
4	u	5
5	v	6
6	w	7
7	8	c
8	u	9
9	v	10
10	d	w
11	t	12
12	u	13
13	v	14
14	w	15

Question 1: Selecting a Single Item How do I select rows having "a" in level "one"? `col one two a t 0 u 1 v 2 w 3` Additionally, how would I be able to drop level "one" in the output? `col two t 0 u 1 v 2 w 3` Question 1b How do I slice all rows with value "t" on level "two"? `col one two a t 0 b t 4 t 8 d t 12` Question 2: Selecting Multiple Values in a Level How can I select rows corresponding to items "b" and "d" in level "one"? `col one two b t 4 u 5 v 6 w 7 t 8 d w 11 t 12 u 13 v 14 w 15` Question 2b How would I get all values corresponding to "t" and "w" in level "two"? `col one two a t 0 w 3 b t 4 w 7 t 8 d w 11 t 12 w 15` Question 3: Slicing a Single Cross Section (x, y) How do I retrieve a cross section, i.e., a single row having a specific values for the index from `df`? Specifically, how do I retrieve the cross section of ('c', 'u'), given by `col one two c u 9` Question 4: Slicing Multiple Cross Sections [(a, b), (c, d), ...] How do I select the two rows corresponding to ('c', 'u'), and ('a', 'w')? `col one two c u 9 a w 3` Question 5: One Item Sliced per Level How can I retrieve all rows corresponding to "a" in level "one" or "t" in level "two"? `col one two a t 0 u 1 v 2 w 3 b t 4 t 8 d t 12` Question 6: Arbitrary Slicing How can I slice specific cross sections? For "a" and "b", I would like to select all rows with sub-levels "u" and "v", and for "d", I would like to select rows with sub-level "w". `col one two a u 1 v 2 b u 5 v 6 d w 11 w 15` Question 7 will use a unique setup consisting of a numeric level: `np.random.seed(0)` `mux2 = pd.MultiIndex.from_arrays([list('aaaabbbbccddddd'), np.random.choice(10, size=16)], names=['one', 'two'])` `df2 = pd.DataFrame({'col': np.arange(len(mux2))}, mux2)`

col	one	two
5	0	0
1	3	2
3	3	3
7	4	9
5	3	6
5	7	2
8	4	9
7	10	d
6	11	8
12	8	13
1	14	6
15		

Question 7: Filtering by numeric inequality on individual levels of the multiindex How do I get all rows where values in level "two" are greater than 5? `col one two b 7 4 9 5 c 7 10 d 6 11 8 12 8 13 6 15` Note: This post will not go through how to create MultiIndexes, how to perform assignment operations on them, or any performance related discussions (these are separate topics for another time).

Answer Score: 387

Stack Overflow answer, Verb Count: 296 , **Adjective Count:** 268

Stack Overflow Answer: MultiIndex / Advanced Indexing Note This post will be structured in the following manner: The questions put forth in the OP will be addressed, one by one For each question, one or more methods applicable to solving this problem and getting the expected result will be

demonstrated. Notes (much like this one) will be included for readers interested in learning about additional functionality, implementation details, and other info cursory to the topic at hand. These notes have been compiled through scouring the docs and uncovering various obscure features, and from my own (admittedly limited) experience. All code samples have been created and tested on pandas v0.23.4, python3.7. If something is not clear, or factually incorrect, or if you did not find a solution applicable to your use case, please feel free to suggest an edit, request clarification in the comments, or open a new question,as applicable. Here is an introduction to some common idioms (henceforth referred to as the Four Idioms) we will be frequently re-visiting [DataFrame.loc](#) - A general solution for selection by label (+ [pd.IndexSlice](#) for more complex applications involving slices) [DataFrame.xs](#) - Extract a particular cross section from a Series/DataFrame. [DataFrame.query](#) - Specify slicing and/or filtering operations dynamically (i.e., as an expression that is evaluated dynamically). Is more applicable to some scenarios than others. Also see this section of the docs for querying on MultiIndexes. Boolean indexing with a mask generated using [MultiIndex.get_level_values](#) (often in conjunction with [Index.isin](#), especially when filtering with multiple values). This is also quite useful in some circumstances. It will be beneficial to look at the various slicing and filtering problems in terms of the Four Idioms to gain a better understanding what can be applied to a given situation. It is very important to understand that not all of the idioms will work equally well (if at all) in every circumstance. If an idiom has not been listed as a potential solution to a problem below, that means that idiom cannot be applied to that problem effectively.

Question 1 How do I select rows having "a" in level "one"? `col one two a t 0 u 1 v 2 w 3` You can use `loc`, as a general purpose solution applicable to most situations: `df.loc[['a']]` At this point, if you get `TypeError: Expected tuple, got str` That means you're using an older version of pandas. Consider upgrading! Otherwise, use `df.loc[('a', slice(None)), :]`. Alternatively, you can use `xs` here, since we are extracting a single cross section. Note the `levels` and `axis` arguments (reasonable defaults can be assumed here). `df.xs('a', level=0, axis=0, drop_level=False)` # `df.xs('a', drop_level=False)` Here, the `drop_level=False` argument is needed to prevent `xs` from dropping level "one" in the result (the level we sliced on). Yet another option here is using `query`: `df.query("one == 'a'")` If the index did not have a name, you would need to change your query string to be `"ilevel_0 == 'a'"`. Finally, using `get_level_values`: `df[df.index.get_level_values('one') == 'a']` # If your levels are unnamed, or if you need to select by position (not label), # `df[df.index.get_level_values(0) == 'a']` Additionally, how would I be able to drop level "one" in the output? `col two t 0 u 1 v 2 w 3` This can be easily done using either `df.loc['a']` # Notice the single string argument instead the list. Or, `df.xs('a', level=0, axis=0, drop_level=True)` # `df.xs('a')` Notice that we can omit the `drop_level` argument (it is assumed to be `True` by default). Note You may notice that a filtered DataFrame may still have all the levels, even if they do not show when printing the DataFrame out. For example, `v = df.loc[['a']]` `print(v)` `col one two a t 0 u 1 v 2 w 3` `print(v.index)` `MultiIndex(levels=[['a', 'b', 'c', 'd'], ['t', 'u', 'v', 'w']], labels=[[0, 0, 0, 0], [0, 1, 2, 3]], names=['one', 'two'])` You can get rid of these levels using `MultiIndex.remove_unused_levels`: `v.index = v.index.remove_unused_levels()` `print(v.index)` `MultiIndex(levels=[['a'], ['t', 'u', 'v', 'w']], labels=[[0, 0, 0, 0], [0, 1, 2, 3]], names=['one', 'two'])` Question 1b How do I slice all rows with value "t" on level "two"? `col one two a t 0 b t 4 t 8 d t 12` Intuitively, you would want something involving `slice()`: `df.loc[(slice(None), 't'), :]` It Just Works! Ñ But it is clunky. We can facilitate a more natural slicing syntax using the `pd.IndexSlice` API here. `idx = pd.IndexSlice` `df.loc[idx[:, 't'], :]` This is much, much cleaner. Note Why is the trailing slice `:` across the columns required? This is because, `loc` can be used to select and slice along both axes (`axis=0` or `axis=1`). Without explicitly making it clear which axis the slicing is to be done on, the operation becomes ambiguous. See the big red box in the documentation on slicing. If you want to remove any shade of ambiguity, `loc` accepts an `axis` parameter: `df.loc(axis=0)[pd.IndexSlice[:, 't']]` Without the `axis` parameter (i.e., just by doing `df.loc[pd.IndexSlice[:, 't']]`), slicing is assumed to be on the columns, and a `KeyError` will be raised in this circumstance. This is

documented in slicers. For the purpose of this post, however, we will explicitly specify all axes. With `xs`, it is `df.xs('t', axis=0, level=1, drop_level=False)` With `query`, it is `df.query("two == 't'")` # Or, if the first level has no name, # `df.query("ilevel_1 == 't'")` And finally, with `get_level_values`, you may do `df[df.index.get_level_values('two') == 't']` # Or, to perform selection by position/integer, # `df[df.index.get_level_values(1) == 't']` All to the same effect.

Question 2 How can I select rows corresponding to items "b" and "d" in level "one"? `col one two b t 4 u 5 v 6 w 7 t 8 d w 11 t 12 u 13 v 14 w 15` Using `loc`, this is done in a similar fashion by specifying a list. `df.loc[['b', 'd']]` To solve the above problem of selecting "b" and "d", you can also use `query`: `items = ['b', 'd'] df.query("one in @items")` # `df.query("one == @items", parser='pandas')` # `df.query("one in ['b', 'd'])` # `df.query("one == ['b', 'd']", parser='pandas')` Note Yes, the default parser is 'pandas', but it is important to highlight this syntax isn't conventionally python. The Pandas parser generates a slightly different parse tree from the expression. This is done to make some operations more intuitive to specify. For more information, please read my post on Dynamic Expression Evaluation in pandas using `pd.eval()`. And, with `get_level_values` + `Index.isin`: `df[df.index.get_level_values("one").isin(['b', 'd'])]`

Question 2b How would I get all values corresponding to "t" and "w" in level "two"? `col one two a t 0 w 3 b t 4 w 7 t 8 d w 11 t 12 w 15` With `loc`, this is possible only in conjunction with `pd.IndexSlice`. `df.loc[pd.IndexSlice[:, ['t', 'w']], :]` The first colon `:` in `pd.IndexSlice[:, ['t', 'w']]` means to slice across the first level. As the depth of the level being queried increases, you will need to specify more slices, one per level being sliced across. You will not need to specify more levels beyond the one being sliced, however. With `query`, this is `items = ['t', 'w'] df.query("two in @items")` # `df.query("two == @items", parser='pandas')` # `df.query("two in ['t', 'w'])` # `df.query("two == ['t', 'w']", parser='pandas')` With `get_level_values` and `Index.isin` (similar to above): `df[df.index.get_level_values('two').isin(['t', 'w'])]`

Question 3 How do I retrieve a cross section, i.e., a single row having a specific values for the index from `df`? Specifically, how do I retrieve the cross section of ('c', 'u'), given by `col one two c u 9` Use `loc` by specifying a tuple of keys: `df.loc[('c', 'u'), :]` Or, `df.loc[pd.IndexSlice[('c', 'u')]]` Note At this point, you may run into a `PerformanceWarning` that looks like this: `PerformanceWarning: indexing past lexsort depth may impact performance`. This just means that your index is not sorted. pandas depends on the index being sorted (in this case, lexicographically, since we are dealing with string values) for optimal search and retrieval. A quick fix would be to sort your DataFrame in advance using `DataFrame.sort_index`. This is especially desirable from a performance standpoint if you plan on doing multiple such queries in tandem: `df_sort = df.sort_index()` `df_sort.loc[('c', 'u')]` You can also use `MultiIndex.is_lexsorted()` to check whether the index is sorted or not. This function returns `True` or `False` accordingly. You can call this function to determine whether an additional sorting step is required or not. With `xs`, this is again simply passing a single tuple as the first argument, with all other arguments set to their appropriate defaults: `df.xs(('c', 'u'))` With `query`, things become a bit clunky: `df.query("one == 'c' and two == 'u'")` You can see now that this is going to be relatively difficult to generalize. But is still OK for this particular problem. With accesses spanning multiple levels, `get_level_values` can still be used, but is not recommended: `m1 = (df.index.get_level_values('one') == 'c') m2 = (df.index.get_level_values('two') == 'u') df[m1 & m2]`

Question 4 How do I select the two rows corresponding to ('c', 'u'), and ('a', 'w')? `col one two c u 9 a w 3` With `loc`, this is still as simple as: `df.loc[[('c', 'u'), ('a', 'w')]]` # `df.loc[pd.IndexSlice[('c', 'u'), ('a', 'w')]]` With `query`, you will need to dynamically generate a query string by iterating over your cross sections and levels: `cses = [('c', 'u'), ('a', 'w')] levels = ['one', 'two']` # This is a useful check to make in advance. `assert all(len(levels) == len(cs) for cs in cses)` `query = '(' + ') or ('.join([' and '.join([f"({l} == {repr(c)}" for l, c in zip(levels, cs)]) for cs in cses]) + ')'` `print(query)` # `((one == 'c') and (two == 'u'))`

or `((one == 'a') and (two == 'w')) df.query(query)` 100% DO NOT RECOMMEND! But it is possible. What if I have multiple levels? One option in this scenario would be to use `droplevel` to drop the levels you're not checking, then use `isin` to test membership, and then boolean index on the final result. `df[df.index.droplevel(used_level).isin(['c', 'u']), ('a', 'w')]` Question 5 How can I retrieve all rows corresponding to "a" in level "one" or "t" in level "two"? `col one two a t 0 u 1 v 2 w 3 b t 4 t 8 d t 12` This is actually very difficult to do with `loc` while ensuring correctness and still maintaining code clarity. `df.loc[pd.IndexSlice['a', 't']]` is incorrect, it is interpreted as `df.loc[pd.IndexSlice[('a', 't')]]` (i.e., selecting a cross section). You may think of a solution with `pd.concat` to handle each label separately: `pd.concat([df.loc[['a'],:], df.loc[pd.IndexSlice[:, 't'],:]])` `col one two a t 0 u 1 v 2 w 3 t 0 # Does this look right to you? No, it isn't! b t 4 t 8 d t 12` But you'll notice one of the rows is duplicated. This is because that row satisfied both slicing conditions, and so appeared twice. You will instead need to do `v = pd.concat([df.loc[['a'],:], df.loc[pd.IndexSlice[:, 't'],:]])` `v[~v.index.duplicated()]` But if your DataFrame inherently contains duplicate indices (that you want), then this will not retain them. Use with extreme caution. With `query`, this is stupidly simple: `df.query("one == 'a' or two == 't'")` With `get_level_values`, this is still simple, but not as elegant: `m1 = (df.index.get_level_values('one') == 'a') m2 = (df.index.get_level_values('two') == 't') df[m1 | m2]` Question 6 How can I slice specific cross sections? For "a" and "b", I would like to select all rows with sub-levels "u" and "v", and for "d", I would like to select rows with sub-level "w". `col one two a u 1 v 2 b u 5 v 6 d w 11 w 15` This is a special case that I've added to help understand the applicability of the Four Idioms, this is one case where none of them will work effectively, since the slicing is very specific, and does not follow any real pattern. Usually, slicing problems like this will require explicitly passing a list of keys to `loc`. One way of doing this is with: `keys = [('a', 'u'), ('a', 'v'), ('b', 'u'), ('b', 'v'), ('d', 'w')] df.loc[keys, :]` If you want to save some typing, you will recognise that there is a pattern to slicing "a", "b" and its sublevels, so we can separate the slicing task into two portions and `concat` the result: `pd.concat([df.loc[('a', 'b'), ('u', 'v')], :, df.loc[('d', 'w'), :]], axis=0)` Slicing specification for "a" and "b" is slightly cleaner `(('a', 'b'), ('u', 'v'))` because the same sub-levels being indexed are the same for each level. Question 7 How do I get all rows where values in level "two" are greater than 5? `col one two b 7 4 9 5 c 7 10 d 6 11 8 12 8 13 6 15` This can be done using `query`, `df2.query("two > 5")` And `get_level_values`, `df2[df2.index.get_level_values('two') > 5]` Note Similar to this example, we can filter based on any arbitrary condition using these constructs. In general, it is useful to remember that `loc` and `xs` are specifically for label-based indexing, while `query` and `get_level_values` are helpful for building general conditional masks for filtering. Bonus Question What if I need to slice a `MultiIndex` column? Actually, most solutions here are applicable to columns as well, with minor changes. Consider: `np.random.seed(0) mux3 = pd.MultiIndex.from_product([list('ABCD'), list('efgh')], names=['one', 'two']) df3 = pd.DataFrame(np.random.choice(10, (3, len(mux3))), columns=mux3) print(df3)`

	one	A	B	C	D	two	e	f	g	h	e	f	g	h	e	f	g	h
0	3	3	7	9	3	5	2	4	7	6	8	8	1	6	1	7	7	8
1	5	9	8	9	4	3	0	3	5	0	2	3	2	8	1	3	3	7
2	0	1	9	9	0	4	7	3	2	7								

These are the following changes you will need to make to the Four Idioms to have them working with columns. To slice with `loc`, use `df3.loc[:, ...]` # Notice how we slice across the index with ``:``. or, `df3.loc[:, pd.IndexSlice[...]]` To use `xs` as appropriate, just pass an argument `axis=1`. You can access the column level values directly using `df3.columns.get_level_values`. You will then need to do something like `df3.loc[:, {condition}]` Where `{condition}` represents some condition built using `columns.get_level_values`. To use `query`, your only option is to transpose, query on the index, and transpose again: `df3.T.query(...).T` Not recommended, use one of the other 3 options.

GPT answer, Verb Count: 24 , Adjective Count: 20

GPT-4o Answer: Answers Question 1: Selecting a Single Item To select rows having "a" in level "one":

`df.loc[("a",)]` To drop level "one" in the output: `df.loc[("a",)].droplevel("one")`

Question 1b To slice all rows with value "t" on level "two": `df.xs("t", level="two")` Question 2:

Selecting Multiple Values in a Level To select rows corresponding to items "b" and "d" in level "one":

`df.loc[["b", "d"]]` To get all values corresponding to "t" and "w" in level "two":

`df[df.index.get_level_values("two").isin(["t", "w"])]` Question 3: Slicing a Single

Cross Section (x, y) To retrieve the cross section of ('c', 'u'): `df.loc[("c", "u")]`

Question 4: Slicing Multiple Cross Sections [(a, b), (c, d), ...] To select the two rows

corresponding to ('c', 'u') and ('a', 'w'): `df.loc[("c", "u"), ("a", "w")]`

Question 5: One Item Sliced per Level To retrieve all rows corresponding to "a" in level "one" or "t" in

level "two": `df[(df.index.get_level_values("one") == "a") |`

`(df.index.get_level_values("two") == "t")]` Question 6: Arbitrary Slicing To slice specific

cross sections as described: `df[(df.index.get_level_values("one").isin(["a", "b"])`

`& df.index.get_level_values("two").isin(["u", "v"])]`

`(df.index.get_level_values("one") == "d") &`

`(df.index.get_level_values("two") == "w")]` Question 7: Filtering by numeric inequality

on individual levels of the MultiIndex To get all rows where values in level "two" are greater than 5:

`df2[df2.index.get_level_values("two") > 5]`

Co-Pilot, Verb Count: 19 , Adjective Count: 21

Co-Pilot Answer: To select rows with a single value/label in a MultiIndex DataFrame, you can use the

`xs` method: `df.xs('a', level='one')` To drop the level "one" from the output, use `droplevel`:

`df.xs('a', level='one').droplevel('one')` To slice all rows with value "t" on level "two",

use: `df.xs('t', level='two')` To select rows with multiple values/labels in a level, use

`IndexSlice`: `idx = pd.IndexSlice df.loc[idx[["b", "d"]], :]` To get all values

corresponding to "t" and "w" in level "two", use: `df.loc[idx[:, ["t", "w"]], :]` To retrieve a

cross section, i.e., a single row having specific values for the index, use `xs`: `df.xs(('c', 'u'))` To

select multiple cross sections, use: `df.loc[[("c", "u"), ("a", "w")]]` To retrieve all rows

corresponding to "a" in level "one" or "t" in level "two", use: `df.loc[idx[["a"], :]]` # 'a' in

level 'one' `df.loc[idx[:, ["t"]], :]` # 't' in level 'two' To slice specific cross

sections, use: `df.loc[idx[["a", "b"], ["u", "v"]]]` `df.loc[idx[["d"], ["w"]]]` To

get all rows where values in level "two" are greater than 5, use:

`df2[df2.index.get_level_values('two') > 5]`

Question 3:

Tags: javascript, reactjs, react-hooks

Question: Are there ways to simulate `componentDidMount` in React functional components via hooks?

Answer Score: 649

Stack Overflow answer, Verb Count: 18 , Adjective Count: 11

Stack Overflow Answer: For the stable version of hooks (React Version 16.8.0+) For

`componentDidMount` `useEffect(() => { // Your code here }, []);` For

`componentDidUpdate` `useEffect(() => { // Your code here }, [yourDependency]);`

For `componentWillUnmount` `useEffect(() => { // componentWillUnmount return ()`

`=> { // Your code here } }, [yourDependency]);` So in this situation, you need to pass

your dependency into this array. Let's assume you have a state like this `const [count, setCount]`

`= useState(0);` And whenever count increases you want to re-render your function component.

Then your `useEffect` should look like this `useEffect(() => { // <div>{count}</div> },`

`[count]);` This way whenever your count updates your component will re-render. Hopefully this will

help a bit.

GPT answer, Verb Count: 12 , Adjective Count: 10

GPT-4o Answer: Simulating `componentDidMount` in React Functional Components Yes, `componentDidMount` behavior can be simulated in functional components using the `useEffect` hook. By providing an empty dependency array (`[]`), the effect runs only once after the component is mounted, replicating the behavior of `componentDidMount`.

```
import React, { useEffect } from 'react'; const MyComponent = () => { useEffect(() => { // Code to run once on mount console.log("Component mounted"); // Optional cleanup function return () => { console.log("Component unmounted"); }; }, []); return My Functional Component; };
```

Co-Pilot, Verb Count: 6 , Adjective Count: 11

Co-Pilot Answer: Yes, you can simulate `componentDidMount` in React functional components using the `useEffect` hook.

```
import React, { useEffect } from 'react'; function MyComponent() { useEffect(() => { // Code to run on component mount console.log('Component mounted'); }, []); // Empty dependency array ensures this runs only once return My Component; } export default MyComponent;
```

Question 4:

Tags: javascript, reactjs, react-hooks

Question: Lets say I have some state that is dependent on some other state (eg when A changes I want B to change). Is it appropriate to create a hook that observes A and sets B inside the `useEffect` hook? Will the effects cascade such that, when I click the button, the first effect will fire, causing b to change, causing the second effect to fire, before the next render? Are there any performance downsides to structuring code like this?

```
let MyComponent = props => { let [a, setA] = useState(1) let [b, setB] = useState(2) useEffect( () => { if (/*some stuff is true*/) { setB(3) } }, [a], ) useEffect( () => { // do some stuff }, [b], ) return ( <button onClick={() => { setA(5) }} > click me </button> ) }
```

Answer Score: 46

Stack Overflow answer, Verb Count: 13 , Adjective Count: 6

Stack Overflow Answer: Effects are always executed after the render phase is completed even if you `setState` inside the one effect, another effect will read the updated state and take action on it only after the render phase. Having said that its probably better to take both actions in the same effect unless there is a possibility that `b` can change due to reasons other than `changing a` in which case too you would want to execute the same logic

GPT answer, Verb Count: 40 , Adjective Count: 23

GPT-4o Answer: Observing State Changes with Hooks and Effects It is appropriate to create a `useEffect` hook that observes state `a` and updates state `b`. However, this may introduce cascading effects, where changing `a` triggers the first effect, which updates `b`, subsequently triggering the second effect before the next render. This behavior is by design in React and ensures that state updates propagate correctly, but there are important considerations. The cascade occurs as follows:

1. `setA(5)` is called, updating state `a`.
2. The first `useEffect` fires due to the dependency on `a`, updating `b` with `setB(3)`.
3. The second `useEffect` fires because `b` changes.

Performance Considerations: While this approach works, it can introduce unnecessary renders and performance overhead, especially in components with complex state logic or frequent updates. This pattern may also make the code harder to debug due to the indirect relationship between effects.

Optimized Approach: Consider deriving `b` directly from `a` using a derived state pattern or memoization. This avoids the need for cascading effects:

```
let MyComponent = props => { let [a, setA] = useState(1); let b = useMemo(() => { return a === /*some
```

```
condition*/ ? 3 : 2; }, [a]); useEffect(() => { // Do some stuff based on b
}, [b]); return ( <button onClick={() => { setA(5); }} > click me </button>
); }; This approach eliminates cascading effects, improving performance and maintainability.
```

Co-Pilot, Verb Count: 31 , **Adjective Count:** 10

Co-Pilot Answer: Yes, it is appropriate to create a hook that observes state A and sets state B inside the `useEffect` hook. This ensures that state B updates whenever state A changes. Regarding effects cascading, when you click the button, the first effect will fire, causing B to change, and then the second effect will fire before the next render. React will batch the state updates and process them together in one re-render. There can be performance downsides if the dependencies in the `useEffect` hooks lead to unnecessary re-renders. It is essential to ensure that the effects are optimized and do not contain any heavy computations that can slow down the rendering process. `let MyComponent = props => { let [a, setA] = useState(1) let [b, setB] = useState(2) useEffect(() => { if (/some stuff is true/) { setB(3) } }, [a],) useEffect(() => { // do some stuff }, [b],) return (onClick={() => { setA(5) }} > click me) }`

Question 5:

Tags: python, merge, join, concatenation, pandas

Question: How can I perform a (`INNER`|(`LEFT`|`RIGHT`|`FULL`) `OUTER`) `JOIN` with pandas? How do I add NaNs for missing rows after a merge? How do I get rid of NaNs after merging? Can I merge on the index? How do I merge multiple DataFrames? Cross join with pandas `merge`? `join`? `concat`? `update`? Who? What? Why?! ... and more. I've seen these recurring questions asking about various facets of the pandas merge functionality. Most of the information regarding merge and its various use cases today is fragmented across dozens of badly worded, unsearchable posts. The aim here is to collate some of the more important points for posterity. This Q&A is meant to be the next installment in a series of helpful user guides on common pandas idioms (see this post on pivoting, and this post on concatenation, which I will be touching on, later). Please note that this post is not meant to be a replacement for the documentation, so please read that as well! Some of the examples are taken from there. Table of Contents For ease of access. Merging basics - basic types of joins (read this first) Index-based joins Generalizing to multiple DataFrames Cross join

Answer Score: 1288

Stack Overflow answer, Verb Count: 148 , **Adjective Count:** 159

Stack Overflow Answer: This post aims to give readers a primer on SQL-flavored merging with Pandas, how to use it, and when not to use it. In particular, here's what this post will go through: The basics - types of joins (`LEFT`, `RIGHT`, `OUTER`, `INNER`) merging with different column names merging with multiple columns avoiding duplicate merge key column in output What this post (and other posts by me on this thread) will not go through: Performance-related discussions and timings (for now). Mostly notable mentions of better alternatives, wherever appropriate. Handling suffixes, removing extra columns, renaming outputs, and other specific use cases. There are other (read: better) posts that deal with that, so figure it out! Note Most examples default to `INNER JOIN` operations while demonstrating various features, unless otherwise specified. Furthermore, all the DataFrames here can be copied and replicated so you can play with them. Also, see this post on how to read DataFrames from your clipboard. Lastly, all visual representation of `JOIN` operations have been hand-drawn using Google Drawings. Inspiration from here. Enough talk - just show me how to use `merge`! Setup & Basics

```
np.random.seed(0) left = pd.DataFrame({'key': ['A', 'B', 'C', 'D'], 'value': np.random.randn(4)}) right = pd.DataFrame({'key': ['B', 'D', 'E', 'F'], 'value': np.random.randn(4)})
```

	key	value
0	A	1.764052
1	B	0.400157
2	C	0.978738
3	D	2.240893

	key	value
0	B	1.867558
1	D	-0.977278
2	E	0.950088
3	F	-0.151357

For the sake of simplicity, the key column has the same name (for now).

An INNER JOIN is represented by Note This, along with the forthcoming figures all follow this convention: blue indicates rows that are present in the merge result red indicates rows that are excluded from the result (i.e., removed) green indicates missing values that are replaced with NaNs in the result To perform an INNER JOIN, call `merge` on the left DataFrame, specifying the right DataFrame and the join key (at the very least) as arguments. `left.merge(right, on='key')` # Or, if you want to be explicit # `left.merge(right, on='key', how='inner')`

key	value_x	value_y
0 B	0.400157	1.867558
1 D	2.240893	-0.977278

This returns only rows from `left` and `right` which share a common key (in this example, "B" and "D"). A LEFT OUTER JOIN, or LEFT JOIN is represented by This can be performed by specifying `how='left'`.

key	value_x	value_y
0 A	1.764052	NaN
1 B	0.400157	1.867558
2 C	0.978738	NaN
3 D	2.240893	-0.977278

Carefully note the placement of NaNs here. If you specify `how='left'`, then only keys from `left` are used, and missing data from `right` is replaced by NaN. And similarly, for a RIGHT OUTER JOIN, or RIGHT JOIN which is... specify `how='right'`: `left.merge(right, on='key', how='right')`

key	value_x	value_y
0 B	0.400157	1.867558
1 D	2.240893	-0.977278
2 E	NaN	0.950088
3 F	NaN	-0.151357

Here, keys from `right` are used, and missing data from `left` is replaced by NaN. Finally, for the FULL OUTER JOIN, given by specify `how='outer'`. `left.merge(right, on='key', how='outer')`

key	value_x	value_y
0 A	1.764052	NaN
1 B	0.400157	1.867558
2 C	0.978738	NaN
3 D	2.240893	-0.977278
4 E	NaN	0.950088
5 F	NaN	-0.151357

This uses the keys from both frames, and NaNs are inserted for missing rows in both. The documentation summarizes these various merges nicely: Other JOINS - LEFT-Excluding, RIGHT-Excluding, and FULL-Excluding/ANTI JOINS If you need LEFT-Excluding JOINS and RIGHT-Excluding JOINS in two steps. For LEFT-Excluding JOIN, represented as Start by performing a LEFT OUTER JOIN and then filtering to rows coming from `left` only (excluding everything from the right), `(left.merge(right, on='key', how='left', indicator=True) .query('_merge == "left_only"') .drop('_merge', axis=1))`

key	value_x	value_y
0 A	1.764052	NaN
2 C	0.978738	NaN

Where, `left.merge(right, on='key', how='left', indicator=True)`

key	value_x	value_y	_merge
0 A	1.764052	NaN	left_only
1 B	0.400157	1.867558	both
2 C	0.978738	NaN	left_only
3 D	2.240893	-0.977278	both

And similarly, for a RIGHT-Excluding JOIN, `(left.merge(right, on='key', how='right', indicator=True) .query('_merge == "right_only"') .drop('_merge', axis=1))`

key	value_x	value_y
2 E	NaN	0.950088
3 F	NaN	-0.151357

Lastly, if you are required to do a merge that only retains keys from the left or right, but not both (IOW, performing an ANTI-JOIN), You can do this in similar fashion, `(left.merge(right, on='key', how='outer', indicator=True) .query('_merge != "both"') .drop('_merge', axis=1))`

key	value_x	value_y
0 A	1.764052	NaN
2 C	0.978738	NaN
4 E	NaN	0.950088
5 F	NaN	-0.151357

Different names for key columns If the key columns are named differently, for example, `left` has `keyLeft`, and `right` has `keyRight` instead of `key`, then you will have to specify `left_on` and `right_on` as arguments instead of `on`: `left2 = left.rename({'key':'keyLeft'}, axis=1)` `right2 = right.rename({'key':'keyRight'}, axis=1)`

keyLeft	value_x	keyRight	value_y
0 A	1.764052	1 B	0.400157
2 C	0.978738	3 D	2.240893
right2 keyRight	value	0 B	1.867558
1 D	-0.977278	2 E	0.950088
3 F	-0.151357	left2.merge(right2, left_on='keyLeft', right_on='keyRight', how='inner')	
keyLeft	value_x	keyRight	value_y
0 B	0.400157	1 D	2.240893
D	-0.977278		

Avoiding duplicate key column in output When merging on `keyLeft` from `left` and `keyRight` from `right`, if you only want either of the `keyLeft` or `keyRight` (but not both) in the output, you can start by setting the index as a preliminary step. `left3 = left2.set_index('keyLeft')` `left3.merge(right2, left_index=True, right_on='keyRight')`

value_x	keyRight	value_y
0 0.400157	B	1.867558
1 2.240893	D	-0.977278

Contrast this with the output of the command just before (that is, the output of `left2.merge(right2, left_on='keyLeft', right_on='keyRight', how='inner')`), you'll notice `keyLeft` is missing. You can figure out what column to keep based on which frame's index is set as the key. This may matter when, say, performing some OUTER JOIN operation. Merging

only a single column from one of the `DataFrames` For example, consider `right3 = right.assign(newcol=np.arange(len(right)))`

key	value	newcol
0	B	1.867558
1	D	-0.977278
2	E	0.950088
3	F	-0.151357

If you are required to merge only "newcol" (without any of the other columns), you can usually just subset columns before merging: `left.merge(right3[['key', 'newcol']], on='key')`

key	value	newcol
0	B	0.400157
1	D	2.240893

If you're doing a LEFT OUTER JOIN, a more performant solution would involve `map`: `# left['newcol'] = left['key'].map(right3.set_index('key')['newcol'])`

key	value	newcol
0	A	1.764052
1	B	0.400157
2	C	0.978738
3	D	2.240893

As mentioned, this is similar to, but faster than `left.merge(right3[['key', 'newcol']], on='key', how='left')`

key	value	newcol
0	A	1.764052
1	B	0.400157
2	C	0.978738
3	D	2.240893

Merging on multiple columns To join on more than one column, specify a list for `on` (or `left_on` and `right_on`, as appropriate). `left.merge(right, on=['key1', 'key2'] ...)` Or, in the event the names are different, `left.merge(right, left_on=['lkey1', 'lkey2'], right_on=['rkey1', 'rkey2'])`

Other useful `merge`* operations and functions Merging a `DataFrame` with `Series` on index: See this answer. Besides `merge`, `DataFrame.update` and `DataFrame.combine_first` are also used in certain cases to update one `DataFrame` with another. `pd.merge_ordered` is a useful function for ordered JOINS. `pd.merge_asof` (read: `merge_asOf`) is useful for approximate joins. This section only covers the very basics, and is designed to only whet your appetite. For more examples and cases, see the documentation on `merge`, `join`, and `concat` as well as the links to the function specifications. Continue Reading Jump to other topics in Pandas Merging 101 to continue learning: Merging basics - basic types of joins * Index-based joins Generalizing to multiple `DataFrames` Cross join *You are here.

GPT answer, Verb Count: 33 , Adjective Count: 41

GPT-4o Answer: Answers to Common Pandas Merge and Join Questions

- How can I perform a (INNER | (LEFT | RIGHT | FULL) OUTER) JOIN with pandas? You can use the `merge` method in pandas to perform different types of joins by specifying the `how` argument: `# INNER JOIN result = pd.merge(df1, df2, on='key', how='inner')` `# LEFT OUTER JOIN result = pd.merge(df1, df2, on='key', how='left')` `# RIGHT OUTER JOIN result = pd.merge(df1, df2, on='key', how='right')` `# FULL OUTER JOIN result = pd.merge(df1, df2, on='key', how='outer')`
- How do I add NaNs for missing rows after a merge? Missing rows automatically get filled with NaN during a `merge` operation that results in unmatched rows. For example: `result = pd.merge(df1, df2, on='key', how='outer')`
- How do I get rid of NaNs after merging? You can use the `fillna` or `dropna` methods: `# Fill NaNs with a specific value result = result.fillna(0)` `# Drop rows with NaNs result = result.dropna()`
- Can I merge on the index? Yes, you can merge on the index by using the `left_index` and `right_index` arguments: `result = pd.merge(df1, df2, left_index=True, right_index=True)`
- How do I merge multiple `DataFrames`? You can iteratively merge multiple `DataFrames` or use `reduce` with `merge`: `from functools import reduce` `dfs = [df1, df2, df3]` `result = reduce(lambda left, right: pd.merge(left, right, on='key', how='inner'), dfs)`
- Cross join with pandas To perform a cross join, use the following pattern: `df1['key'] = 1` `df2['key'] = 1` `result = pd.merge(df1, df2, on='key').drop('key', axis=1)`
- `merge`? `join`? `concat`? `update`? Who? What? Why?! `merge`: Used for merging `DataFrames` with database-style joins. `join`: A convenience method for joining `DataFrames` on their index or a key column. `concat`: Used for concatenating `DataFrames` along a particular axis. `update`: Updates values in a `DataFrame` with values from another `DataFrame`, aligning on the index.

Co-Pilot, Verb Count: 23 , Adjective Count: 44

Co-Pilot Answer: To perform an `INNER`, `LEFT OUTER`, `RIGHT OUTER`, or `FULL OUTER` join with pandas, you can use the `merge` function: `df1.merge(df2, how='inner', on='key')`

`df1.merge(df2, how='left', on='key')` `df1.merge(df2, how='right', on='key')`
`df1.merge(df2, how='outer', on='key')` To add NaNs for missing rows after a merge, you can use the `how='outer'` parameter: `df1.merge(df2, how='outer', on='key')` To get rid of NaNs after merging, you can use the `fillna` method: `df.fillna(0)` Yes, you can merge on the index by using the `left_index` and `right_index` parameters: `df1.merge(df2, left_index=True, right_index=True)` To merge multiple DataFrames, you can use the `reduce` function from the `functools` module along with `merge`: `from functools import reduce` `dfs = [df1, df2, df3]` `df_merged = reduce(lambda left, right: pd.merge(left, right, on='key'), dfs)` For a cross join, you can create a key column in both DataFrames with the same value, merge on the key, and then drop the key column: `df1['key'] = 1` `df2['key'] = 1` `df_cross = df1.merge(df2, on='key').drop('key', axis=1)` The choice between `merge`, `join`, `concat`, and `update` depends on the specific use case: `merge` - for merging DataFrames on columns or indexes `join` - for joining DataFrames on indexes `concat` - for concatenating DataFrames along rows or columns `update` - for updating DataFrame values with another DataFrame's values

Month: Jan-19

Question 1:

Tags: javascript, reactjs, react-hooks

Question: I am trying to learn hooks and the `useState` method has made me confused. I am assigning an initial value to a state in the form of an array. The `set` method in `useState` is not working for me, both with and without the spread syntax. I have made an API on another PC that I am calling and fetching the data which I want to set into the state. Here is my code:

```
<div id="root"></div>
<script type="text/babel" defer> // import React, { useState, useEffect }
from "react"; // import ReactDOM from "react-dom"; const { useState,
useEffect } = React; // web-browser variant const StateSelector = () => {
const initialValue = [ { category: "", photo: "", description: "", id: 0,
name: "", rating: 0 } ]; const [movies, setMovies] = useState(initialValue);
useEffect(() => { (async function() { try { // const response = await
fetch("http://192.168.1.164:5000/movies/display"); // const json = await
response.json(); // const result = json.data.result; const result = [ {
category: "cat1", description: "desc1", id: "1546514491119", name:
"randomname2", photo: null, rating: "3" }, { category: "cat2", description:
"desc1", id: "1546837819818", name: "randomname1", rating: "5" } ] };
console.log("result =", result); setMovies(result); console.log("movies =",
movies); } catch (e) { console.error(e); } })(); }, []); return
<p>hello</p>; }; const rootElement = document.getElementById("root");
ReactDOM.render(<StateSelector />, rootElement); </script> <script
src="https://unpkg.com/@babel/standalone@7/babel.min.js"></script> <script
src="https://unpkg.com/react@17/umd/react.production.min.js"></script>
<script src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"
></script> Neither setMovies(result) nor setMovies(...result) works. I expect the
result variable to be pushed into the movies array.
```

Answer Score: 958

Stack Overflow answer, Verb Count: 41 , **Adjective Count:** 27

Stack Overflow Answer: Much like `.setState()` in class components created by extending `React.Component` or `React.PureComponent`, the state update using the updater provided by `useState` hook is also asynchronous, and will not be reflected immediately. Also, the main issue here

is not just the asynchronous nature but the fact that state values are used by functions based on their current closures, and state updates will reflect in the next re-render by which the existing closures are not affected, but new ones are created. Now in the current state, the values within hooks are obtained by existing closures, and when a re-render happens, the closures are updated based on whether the function is recreated again or not. Even if you add a `setTimeout` the function, though the timeout will run after some time by which the re-render would have happened, the `setTimeout` will still use the value from its previous closure and not the updated one. `setMovies(result);`
`console.log(movies) // movies here will not be updated` If you want to perform an action on state update, you need to use the `useEffect` hook, much like using `componentDidUpdate` in class components since the setter returned by `useState` doesn't have a callback pattern `useEffect(() => { // action on update of movies }, [movies]);` As far as the syntax to update state is concerned, `setMovies(result)` will replace the previous `movies` value in the state with those available from the async request. However, if you want to merge the response with the previously existing values, you must use the callback syntax of state update along with the correct use of spread syntax like `setMovies(prevMovies => ([...prevMovies, ...result]));`

GPT answer, Verb Count: 22 , Adjective Count: 17

GPT-4o Answer: Understanding `useState` and State Updates The issue lies in the way state updates are handled in React. The `setMovies` function does not update the `movies` variable immediately. React batches state updates, and the updated value is available in the next render cycle, not immediately after calling `setMovies`. Here is the corrected implementation: `const StateSelector = () => { const initialValue = [{ category: "", photo: "", description: "", id: 0, name: "", rating: 0 }]; const [movies, setMovies] = useState(initialValue); useEffect(() => { (async function() { try { const result = [{ category: "cat1", description: "desc1", id: "1546514491119", name: "randomname2", photo: null, rating: "3" }, { category: "cat2", description: "desc1", id: "1546837819818", name: "randomname1", rating: "5" }]; console.log("result =", result); setMovies(result); // This updates the state } catch (e) { console.error(e); } })(); }, []); useEffect(() => { console.log("movies =", movies); // Log updated state on re-render }, [movies]); return <p>hello</p>; }; Key Notes: 1. The updated state is not available immediately after calling setMovies. Use an effect with movies in the dependency array to observe changes. 2. setMovies(...result) is not valid because ...result spreads the array into individual elements, which is not what useState expects. Simply pass the array directly to setMovies.`

Co-Pilot, Verb Count: 21 , Adjective Count: 13

Co-Pilot Answer: The issue lies in how React's `useState` works within an asynchronous function. The `setMovies` call will indeed update the state, but the `console.log` immediately following it will show the previous state value because state updates are asynchronous. To confirm that the state has been updated, place the `console.log` in a different place. Here's a revised version of your code: `const StateSelector = () => { const initialValue = [{ category: "", photo: "", description: "", id: 0, name: "", rating: 0 }]; const [movies, setMovies] = useState(initialValue); useEffect(() => { (async function() { try { // const response = await fetch("http://192.168.1.164:5000/movies/display"); // const json = await response.json(); // const result = json.data.result; const result = [{ category: "cat1", description: "desc1", id: "1546514491119", name: "randomname2", photo: null, rating: "3" }, { category: "cat2", description: "desc1", id: "1546837819818", name: "randomname1", rating: "5" }]; console.log("result =", result); setMovies(result); } catch (e) { console.error(e); } })(); }, []); useEffect(() => { console.log("movies =",`

```
movies)); }, [movies]); return hello; }; const rootElement =
document.getElementById("root"); ReactDOM.render(, rootElement);
```

Question 2:

Tags: python, performance, apply, pandas

Question: I have seen many answers posted to questions on Stack Overflow involving the use of the Pandas method `apply`. I have also seen users commenting under them saying that "`apply` is slow, and should be avoided". I have read many articles on the topic of performance that explain `apply` is slow. I have also seen a disclaimer in the docs about how `apply` is simply a convenience function for passing UDFs (can't seem to find that now). So, the general consensus is that `apply` should be avoided if possible. However, this raises the following questions: If `apply` is so bad, then why is it in the API? How and when should I make my code `apply`-free? Are there ever any situations where `apply` is good (better than other possible solutions)?

Answer Score: 266

Stack Overflow answer, Verb Count: 193 , **Adjective Count:** 171

Stack Overflow Answer: `apply`, the Convenience Function you Never Needed We start by addressing the questions in the OP, one by one. "If `apply` is so bad, then why is it in the API?" `DataFrame.apply` and `Series.apply` are convenience functions defined on `DataFrame` and `Series` object respectively. `apply` accepts any user defined function that applies a transformation/aggregation on a `DataFrame`. `apply` is effectively a silver bullet that does whatever any existing pandas function cannot do. Some of the things `apply` can do: Run any user-defined function on a `DataFrame` or `Series` Apply a function either row-wise (`axis=1`) or column-wise (`axis=0`) on a `DataFrame` Perform index alignment while applying the function Perform aggregation with user-defined functions (however, we usually prefer `agg` or `transform` in these cases) Perform element-wise transformations Broadcast aggregated results to original rows (see the `result_type` argument). Accept positional/keyword arguments to pass to the user-defined functions. ...Among others. For more information, see Row or Column-wise Function Application in the documentation. So, with all these features, why is `apply` bad? It is because `apply` is slow. Pandas makes no assumptions about the nature of your function, and so iteratively applies your function to each row/column as necessary. Additionally, handling all of the situations above means `apply` incurs some major overhead at each iteration. Further, `apply` consumes a lot more memory, which is a challenge for memory bounded applications. There are very few situations where `apply` is appropriate to use (more on that below). If you're not sure whether you should be using `apply`, you probably shouldn't. pandas 2.2 update: `apply` now supports `engine='numba'` More info in the release notes as well as GH54666 Choose between the python (default) engine or the numba engine in `apply`. The numba engine will attempt to JIT compile the passed function, which may result in speedups for large `DataFrames`. It also supports the following `engine_kwargs` : `nopython` (compile the function in `nopython` mode) `nogil` (release the GIL inside the JIT compiled function) `parallel` (try to apply the function in parallel over the `DataFrame`) Note: Due to limitations within numba/how pandas interfaces with numba, you should only use this if `raw=True` Let's address the next question. "How and when should I make my code `apply`-free?" To rephrase, here are some common situations where you will want to get rid of any calls to `apply`. Numeric Data If you're working with numeric data, there is likely already a vectorized cython function that does exactly what you're trying to do (if not, please either ask a question on Stack Overflow or open a feature request on GitHub). Contrast the performance of `apply` for a simple addition operation. `df = pd.DataFrame({"A": [9, 4, 2, 1], "B": [12, 7, 5, 4]})` `df` A B 0 9 12 1 4 7 2 2 5 3 1 4 <!-->`df.apply(np.sum)` A 16 B 28 `dtype: int64` `df.sum()` A 16 B 28 `dtype: int64` Performance wise, there's no comparison, the cythonized equivalent is much faster. There's no need for a graph, because the difference is obvious even for toy data. `%timeit df.apply(np.sum)` `%timeit df.sum()` 2.22 ms \pm 41.2 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each) 471 μ s \pm 8.16 μ s per loop (mean \pm std.

dev. of 7 runs, 1000 loops each) Even if you enable passing raw arrays with the `raw` argument, it's still twice as slow. `%timeit df.apply(np.sum, raw=True)` 840 μ s \pm 691 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each) Another example: `df.apply(lambda x: x.max() - x.min())` A 8 B 8 dtype: int64 `df.max() - df.min()` A 8 B 8 dtype: int64 `%timeit df.apply(lambda x: x.max() - x.min())` `%timeit df.max() - df.min()` 2.43 ms \pm 450 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each) 1.23 ms \pm 14.7 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each) In general, seek out vectorized alternatives if possible.

String/Regex Pandas provides "vectorized" string functions in most situations, but there are rare cases where those functions do not... "apply", so to speak. A common problem is to check whether a value in a column is present in another column of the same row. `df = pd.DataFrame({'Name': ['mickey', 'donald', 'minnie'], 'Title': ['wonderland', 'welcome to donald's castle', 'Minnie mouse clubhouse'], 'Value': [20, 10, 86]})` `df` Name Value Title 0 mickey 20 wonderland 1 donald 10 welcome to donald's castle 2 minnie 86 Minnie mouse clubhouse This should return the row second and third row, since "donald" and "minnie" are present in their respective "Title" columns. Using `apply`, this would be done using `df.apply(lambda x: x['Name'].lower() in x['Title'].lower(), axis=1)` 0 False 1 True 2 True dtype: bool `df[df.apply(lambda x: x['Name'].lower() in x['Title'].lower(), axis=1)]` Name Title Value 1 donald welcome to donald's castle 10 2 minnie Minnie mouse clubhouse 86 However, a better solution exists using list comprehensions. `df[[y.lower() in x.lower() for x, y in zip(df['Title'], df['Name'])]]` Name Title Value 1 donald welcome to donald's castle 10 2 minnie Minnie mouse clubhouse 86 `<!-- %timeit df[df.apply(lambda x: x['Name'].lower() in x['Title'].lower(), axis=1)] %timeit df[[y.lower() in x.lower() for x, y in zip(df['Title'], df['Name'])]]` 2.85 ms \pm 38.4 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each) 788 μ s \pm 16.4 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each) The thing to note here is that iterative routines happen to be faster than `apply`, because of the lower overhead. If you need to handle NaNs and invalid dtypes, you can build on this using a custom function you can then call with arguments inside the list comprehension. For more information on when list comprehensions should be considered a good option, see my writeup: Are for-loops in pandas really bad? When should I care?.

Note Date and datetime operations also have vectorized versions. So, for example, you should prefer `pd.to_datetime(df['date'])`, over, say, `df['date'].apply(pd.to_datetime)`. Read more at the docs. A Common Pitfall: Exploding Columns of Lists `s = pd.Series([[1, 2]] * 3)` `s` 0 [1, 2] 1 [1, 2] 2 [1, 2] dtype: object People are tempted to use `apply(pd.Series)`. This is horrible in terms of performance. `s.apply(pd.Series)` 0 1 0 1 2 1 1 2 2 1 2 A better option is to listify the column and pass it to `pd.DataFrame`. `pd.DataFrame(s.tolist())` 0 1 0 1 2 1 1 2 2 1 2 `<!-- %timeit s.apply(pd.Series) %timeit pd.DataFrame(s.tolist())` 2.65 ms \pm 294 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each) 816 μ s \pm 40.5 μ s per loop (mean \pm std. dev. of 7 runs, 1000 loops each) Lastly, "Are there any situations where `apply` is good?" `Apply` is a convenience function, so there are situations where the overhead is negligible enough to forgive. It really depends on how many times the function is called. Functions that are Vectorized for Series, but not DataFrames What if you want to apply a string operation on multiple columns? What if you want to convert multiple columns to datetime? These functions are vectorized for Series only, so they must be applied over each column that you want to convert/operate on. `df = pd.DataFrame(pd.date_range('2018-12-31', '2019-01-31', freq='2D').date.astype(str).reshape(-1, 2), columns=['date1', 'date2'])` `df` date1 date2 0 2018-12-31 2019-01-02 1 2019-01-04 2019-01-06 2 2019-01-08 2019-01-10 3 2019-01-12 2019-01-14 4 2019-01-16 2019-01-18 5 2019-01-20 2019-01-22 6 2019-01-24 2019-01-26 7 2019-01-28 2019-01-30 `df.dtypes` date1 object date2 object dtype: object This is an admissible case for `apply`: `df.apply(pd.to_datetime, errors='coerce').dtypes` date1 datetime64[ns] date2

`datetime64[ns] dtype: object` Note that it would also make sense to `stack`, or just use an explicit loop. All these options are slightly faster than using `apply`, but the difference is small enough to forgive.

`%timeit df.apply(pd.to_datetime, errors='coerce') %timeit`

`pd.to_datetime(df.stack(), errors='coerce').unstack() %timeit`

`pd.concat([pd.to_datetime(df[c], errors='coerce') for c in df], axis=1)`

`%timeit for c in df.columns: df[c] = pd.to_datetime(df[c], errors='coerce')`

 5.49 ms \pm 247 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each)

 3.94 ms \pm 48.1 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each)

 3.16 ms \pm 216 μ s per loop (mean \pm std. dev. of 7 runs, 100 loops each)

 2.41 ms \pm 1.71 ms per loop (mean \pm std. dev. of 7 runs, 1 loop each) You can make a similar case for other operations such as string operations, or conversion to category.

`u = df.apply(lambda x: x.str.contains(...)) v = df.apply(lambda x: x.astype(category)) v/s u = pd.concat([df[c].str.contains(...) for c in df], axis=1) v = df.copy() for c in df: v[c] = df[c].astype(category)` And so on...

Converting Series to `str`: `astype` versus `apply` This seems like an idiosyncrasy of the API. Using `apply` to convert integers in a Series to string is comparable (and sometimes faster) than using `astype`. The graph was plotted using the `perfplot` library.

`import perfplot
perfplot.show(
 setup=lambda n: pd.Series(np.random.randint(0, n, n)),
 kernels=[lambda s: s.astype(str), lambda s: s.apply(str)],
 labels=['astype', 'apply'],
 n_range=[2**k for k in range(1, 20)],
 xlabel='N',
 logx=True,
 logy=True,
 equality_check=lambda x, y: (x == y).all())`

 With floats, I see the `astype` is consistently as fast as, or slightly faster than `apply`. So this has to do with the fact that the data in the test is integer type.

GroupBy operations with chained transformations `GroupBy.apply` has not been discussed until now, but `GroupBy.apply` is also an iterative convenience function to handle anything that the existing `GroupBy` functions do not. One common requirement is to perform a `GroupBy` and then two prime operations such as a "lagged cumsum":

`df = pd.DataFrame({"A": list('aabcccddee'), "B": [12, 7, 5, 4, 5, 4, 3, 2, 1, 10]})`

`df`

 A B 0 a 12 1 a 7 2 b 5 3 c 4 4 c 5 5 c 4 6 d 3 7 d 2 8 e 1 9 e 10

`<!--> You'd need two successive groupby calls here:`

`df.groupby('A').B.cumsum().groupby(df.A).shift()`

 0 NaN 1 12.0 2 NaN 3 NaN 4 4.0 5 9.0 6 NaN 7 3.0 8 NaN 9 1.0

 Name: B, dtype: float64

 Using `apply`, you can shorten this to a single call.

`df.groupby('A').B.apply(lambda x: x.cumsum().shift())`

 0 NaN 1 12.0 2 NaN 3 NaN 4 4.0 5 9.0 6 NaN 7 3.0 8 NaN 9 1.0

 Name: B, dtype: float64

 It is very hard to quantify the performance because it depends on the data. But in general, `apply` is an acceptable solution if the goal is to reduce a `groupby` call (because `groupby` is also quite expensive).

Other Caveats Aside from the caveats mentioned above, it is also worth mentioning that `apply` operates on the first row (or column) twice. This is done to determine whether the function has any side effects. If not, `apply` may be able to use a fast-path for evaluating the result, else it falls back to a slow implementation.

`df = pd.DataFrame({'A': [1, 2], 'B': ['x', 'y']})`

`def func(x): print(x['A']) return x`

`df.apply(func, axis=1)`

 # 1 # 1 # 2

 A B 0 1 x 1 2 y

 This behaviour is also seen in `GroupBy.apply` on pandas versions < 0.25 (it was fixed for 0.25, see here for more information.)

GPT answer, Verb Count: 29 , Adjective Count: 39

GPT-4o Answer: Pandas `apply`: Usage, Alternatives, and Recommendations

 1. Why is `apply` in the API if it is slow? The `apply` method is included in the API because it offers a flexible and convenient way to apply user-defined functions (UDFs) across rows or columns of a DataFrame or Series. While it is not optimized for performance, it provides a simple and readable solution for scenarios where vectorized operations or built-in methods are not available.

 2. How and when should I make my code `apply`-free? You should avoid `apply` whenever possible by leveraging vectorized operations or native pandas methods, which are implemented in C and are significantly faster.

 For example:

 # Using `apply`

`df['new_col'] = df['existing_col'].apply(lambda x: x * 2)`

 # Vectorized alternative

`df['new_col'] = df['existing_col'] * 2`

 To replace `apply`:

 1. Use vectorized operations for mathematical or logical transformations.

 2. Use

pandas built-in methods such as `map`, `agg`, or `transform`. 3. Leverage NumPy functions if compatible with your operation. 3. Are there ever any situations where `apply` is good? Yes, `apply` is useful in situations where: 1. The transformation logic is complex and cannot be expressed using vectorized operations. 2. You need to apply a custom Python function to rows or columns of a DataFrame. 3. The overhead of using `apply` is acceptable compared to the effort required to refactor the code for performance. Example where `apply` might be appropriate: # Applying a complex function to rows `def custom_function(row): return row['col1'] + row['col2'] * 2 df['new_col'] = df.apply(custom_function, axis=1)` In such cases, `apply` provides clarity and simplicity at the cost of performance.

Co-Pilot, Verb Count: 19 , **Adjective Count:** 21

Co-Pilot Answer: `apply` is included in the Pandas API because it provides a flexible and convenient way to apply custom functions across DataFrame rows or columns. It allows users to perform operations that might not be easily achievable with vectorized operations. To make your code `apply`-free, you should aim to use vectorized operations provided by Pandas and NumPy. These operations are typically faster and more efficient than using `apply` because they leverage optimized C code and avoid the overhead of Python loops. Situations where `apply` can be useful include cases where you need to apply a function that cannot be easily vectorized or when the logic of the function is complex and involves multiple steps or conditions that are not straightforward to express with vectorized operations. While `apply` might not be the most performant option, it can be a practical and readable solution for certain tasks.

Question 3:

Tags: javascript, node.js, npm, package-lock.json

Question: In a team set up, usually, I have faced merge conflicts in `package-lock.json` and my quick fix has always been to delete the file and regenerate it with `npm install`. I have not seriously thought about the implication of this fix because it has not caused any perceivable problem before. Is there a problem with deleting the file and having `npm` recreate it that way instead of resolving the conflicts manually?

Answer Score: 211

Stack Overflow answer, Verb Count: 36 , **Adjective Count:** 32

Stack Overflow Answer: Yes, it can and will affect all the project in really bad way. if your team does not run `npm install` after each `git pull` you all are using different dependencies' versions. So it ends with "but it works for me!!" and "I don't understand why my code does not work for you" even if all the team runs `npm install` it still does not mean everything is ok. at some moment you may find your project acts differently. in a part that you have not been changing for years. and after (probably, quite painful) debugging you will find it's because of 3rd level dependency has updated for next major version and this led some breaking changes. Conclusion: don't ever delete `package-lock.json`. Yes, for first level dependencies if we specify them without ranges (like `"react": "16.12.0"`) we get the same versions each time we run `npm install`. But we cannot say the same about dependencies of 2+ level deep (dependencies that our dependencies are relying on), so `package-lock.json` is really important for stability. In your case you better do next way: fix conflicts in `package.json` run `npm install` As easy as it looks. The same to yarn - it fixes lockfile conflict on its own. The only requirement here to resolve all the conflicts in `package.json` beforehand if any. Per docs npm will fix merge conflicts in `package-lock.json` for you. [Upd from 2021] important! If you use some library already and npm/GitHub account of its maintainer is hacked. And new version with malicious code inside is released. And you have `package-lock.json` intact. You will be fine. If you drop it you are in trouble.

GPT answer, Verb Count: 39 , Adjective Count: 24

GPT-4o Answer: Handling Merge Conflicts in `package-lock.json` Deleting the `package-lock.json` file and regenerating it with `npm install` is a quick fix, but it has potential implications that you should consider: Implications of Deleting and Regenerating `package-lock.json`: 1. The regenerated `package-lock.json` file might result in different dependency versions being resolved compared to the original file. This can lead to inconsistencies in the project across team members or environments. 2. If a previous version of `package-lock.json` contained specific dependency resolutions to address bugs or compatibility issues, regenerating the file may discard these fixes. 3. It undermines the purpose of `package-lock.json`, which is to ensure consistent dependency trees across installations. Recommended Approach: Manually resolve merge conflicts in `package-lock.json` to maintain consistency and avoid unintended changes: # Steps to resolve: 1. Open `package-lock.json` and examine the conflict markers (`<<<<<<< HEAD, =====, >>>>>>> branch_name`). 2. Determine the correct dependency versions and resolve the conflicts manually. 3. After resolving, run `npm install` to ensure the lock file is consistent with `package.json`. 4. Commit the resolved `package-lock.json` file. When to Regenerate: If resolving conflicts manually is infeasible or the file is severely corrupted, deleting and regenerating the file may be acceptable. Ensure that: 1. You communicate the change to your team to avoid surprises. 2. You test the application thoroughly after regenerating to verify that dependency updates do not introduce issues.

Co-Pilot, Verb Count: 22 , Adjective Count: 15

Co-Pilot Answer: Yes, there are potential problems with deleting `package-lock.json` and regenerating it with `npm install` instead of resolving the conflicts manually. `package-lock.json` ensures that the dependencies are consistent across different environments. By deleting and regenerating this file, you may inadvertently update dependencies to newer versions that could introduce compatibility issues or bugs. This could lead to inconsistencies between the development, testing, and production environments, causing unexpected behavior or errors. Additionally, manually resolving conflicts helps maintain the exact versions of dependencies specified in `package-lock.json`, ensuring that the application behaves as expected. It also prevents potential security vulnerabilities that may arise from unintended dependency updates. Therefore, it's best to manually resolve conflicts in `package-lock.json` to maintain consistency and avoid potential issues.

Question 4:

Tags: python, virtualenv, visual-studio-code

Question: In my project folder I created a venv folder: `python -m venv venv` When I run command `select python interpreter` in Visual Studio Code, my venv folder is not shown. I went one level up like suggested here, but Visual Studio Code doesn't see my virtual interpreter. What did I miss?

Answer Score: 239

Stack Overflow answer, Verb Count: 17 , Adjective Count: 12

Stack Overflow Answer: I have been using Visual Studio Code for a while now and found an another way to show virtual environments in Visual Studio Code. Go to the parent folder in which `venv` is there through a command prompt. Type `code .` and Enter. [It is working on both Windows and Linux for me.] That should also show the virtual environments present in that folder. Original Answer I almost run into same problem every time I am working on Visual Studio Code using venv. I follow the below steps: Go to menu File ,> Preferences ,> Settings. Click on Workspace settings. Under Files:Association, in the JSON: Schemas section, you will find Edit in settings.json. Click on that. Update "`python.defaultInterpreterPath`": "`Your_venv_path/bin/python`" under workspace settings. (For Windows): Update "`python.defaultInterpreterPath`":

"Your_venv_path\Scripts\python.exe" under workspace settings. Restart Visual Studio Code in case if it still doesn't show your venv. Note: Use `python.pythonPath` instead of `python.defaultInterpreterPath` for older versions.

GPT answer, Verb Count: 30 , Adjective Count: 16

GPT-4o Answer: Fixing the Missing Virtual Environment in Visual Studio Code To resolve the issue of Visual Studio Code not detecting your virtual environment, follow these steps: 1. Activate the Virtual Environment # On Windows: `venv\Scripts\activate` # On macOS/Linux: `source venv/bin/activate` Verify the virtual environment is active by checking the terminal prompt or running: `python --version` 2. Manually Select the Interpreter If the virtual environment is not listed automatically: 1. Open the Command Palette (Ctrl+Shift+P or Cmd+Shift+P on macOS). 2. Search for "Python: Select Interpreter". 3. Click "Enter interpreter path". 4. Choose "Find..." and navigate to the Python executable inside your virtual environment: - Windows: `venv\Scripts\python.exe` - macOS/Linux: `venv/bin/python` 3. Update the Workspace Settings Add the virtual environment to the workspace settings in `.vscode/settings.json`: `{ "python.pythonPath": "./venv/bin/python" }` // Adjust the path based on your OS } 4. Verify VS Code Python Extension Ensure the Python extension is installed and updated. You can install it from the Extensions Marketplace in Visual Studio Code. 5. Restart Visual Studio Code Sometimes, restarting Visual Studio Code is required to refresh the interpreter list. By following these steps, your virtual environment should now be recognized and selectable within Visual Studio Code.

Co-Pilot, Verb Count: 9 , Adjective Count: 8

Co-Pilot Answer: You missed selecting the correct interpreter path in Visual Studio Code settings. Try updating the `python.pythonPath` setting in your `settings.json` file to point to the virtual environment's Python interpreter: `"python.pythonPath": "venv/bin/python"` Make sure you open the directory containing the `venv` folder, not the `venv` folder itself.

Question 5:

Tags: javascript, reactjs, react-hooks

Question: What is the correct way of updating state, in a nested object, in React with Hooks? `export Example = () => { const [exampleState, setExampleState] = useState({ masterField: { fieldOne: "a", fieldTwo: { fieldTwoOne: "b" fieldTwoTwo: "c" } } })` How would one use `setExampleState` to update `exampleState` to a (appending an field)? `const a = { masterField: { fieldOne: "a", fieldTwo: { fieldTwoOne: "b", fieldTwoTwo: "c" } }, masterField2: { fieldOne: "c", fieldTwo: { fieldTwoOne: "d", fieldTwoTwo: "e" } }, }` b (Changing values)? `const b = { masterField: { fieldOne: "e", fieldTwo: { fieldTwoOne: "f" fieldTwoTwo: "g" } } }`

Answer Score: 292

Stack Overflow answer, Verb Count: 1 , Adjective Count: 3

Stack Overflow Answer: You can pass new value like this:

```
setExampleState({...exampleState, masterField2: { fieldOne: "a", fieldTwo: { fieldTwoOne: "b", fieldTwoTwo: "c" } }, })
```

GPT answer, Verb Count: 24 , Adjective Count: 11

GPT-4o Answer: Updating Nested State in React with Hooks To update nested state in React using hooks, you must ensure immutability by creating a copy of the existing state and updating the necessary fields. Here is how you can handle the scenarios described: 1. Appending a Field (Updating to a): `setExampleState(prevState => ({ ...prevState, masterField2: { fieldOne:`

```
"c", fieldTwo: { fieldTwoOne: "d", fieldTwoTwo: "e" } } }));
```

2. Changing Values (Updating to b): `setExampleState(prevState => ({ ...prevState, masterField: { fieldOne: "e", fieldTwo: { fieldTwoOne: "f", fieldTwoTwo: "g" } } }));`

Explanation: 1. Use the spread operator (...) to shallow copy the existing state. 2. Modify the specific nested fields while preserving the rest of the structure. 3. Always update the state immutably to avoid unexpected behavior in React.

Co-Pilot, Verb Count: 12 , Adjective Count: 8

Co-Pilot Answer: To update the state in a nested object in React with Hooks, you should use the spread operator to ensure that you are immutably updating the state. Below is how you can achieve this: To update `exampleState` to include `masterField2` (appending a field): `const a = { masterField: { fieldOne: "a", fieldTwo: { fieldTwoOne: "b", fieldTwoTwo: "c" } }, masterField2: { fieldOne: "c", fieldTwo: { fieldTwoOne: "d", fieldTwoTwo: "e" } } }; setExampleState(prevState => ({ ...prevState, ...a }));` To update `exampleState` to change values (changing values): `const b = { masterField: { fieldOne: "e", fieldTwo: { fieldTwoOne: "f", fieldTwoTwo: "g" } } }; setExampleState(prevState => ({ ...prevState, masterField: { ...prevState.masterField, ...b.masterField } }));`

Month: Feb-19

Question 1:

Tags: javascript, reactjs, react-hooks

Question: As far as I understood I can use refs for a single element like this: `const { useRef, useState, useEffect } = React; const App = () => { const elRef = useRef(); const [elWidth, setElWidth] = useState(); useEffect(() => { setElWidth(elRef.current.offsetWidth); }, []); return (<div> <div ref={elRef} style={{ width: "100px" }}> Width is: {elWidth} </div> </div>); }; ReactDOM.render(<App />, document.getElementById("root")); <script src="https://unpkg.com/react@16/umd/react.production.min.js"></script> <script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script> <div id="root"></div>` How can I implement this for an array of elements? Obviously not like that: (I knew it even I did not try it:) `const { useRef, useState, useEffect } = React; const App = () => { const elRef = useRef(); const [elWidth, setElWidth] = useState(); useEffect(() => { setElWidth(elRef.current.offsetWidth); }, []); return (<div> {[1, 2, 3].map(el => (<div ref={elRef} style={{ width: `${el * 100}px` })> Width is: {elWidth} </div>))} </div>); }; ReactDOM.render(<App />, document.getElementById("root")); <script src="https://unpkg.com/react@16/umd/react.production.min.js"></script> <script src="https://unpkg.com/react-dom@16/umd/react-dom.production.min.js"></script> <div id="root"></div>` I have seen this and hence this. But, I'm still confused about how to implement that suggestion for this simple case.

Answer Score: 168

Stack Overflow answer, Verb Count: 17 , Adjective Count: 13

Stack Overflow Answer: A ref is initially just `{ current: null }` object. `useRef` keeps the reference to this object between component renders. `current` value is primarily intended for component refs but can hold anything. There should be an array of refs at some point. In case the array length may vary between renders, an array should scale accordingly: `const arrLength =`

```

arr.length; const [elRefs, setElRefs] = React.useState([]);
React.useEffect(() => { // add or remove refs setElRefs((elRefs) =>
Array(arrLength) .fill() .map((_, i) => elRefs[i] || createRef()), ); },
[arrLength]); return ( <div> {arr.map((el, i) => ( <div ref={elRefs[i]}
style={...}> ... </div> ))} </div> ); This piece of code can be optimized by unwrapping
useEffect and replacing useState with useRef but it should be noted that doing side effects in
render function is generally considered a bad practice: const arrLength = arr.length; const
elRefs = React.useRef([]); if (elRefs.current.length !== arrLength) { // add
or remove refs elRefs.current = Array(arrLength) .fill() .map((_, i) =>
elRefs.current[i] || createRef()); } return ( <div> {arr.map((el, i) => (
<div ref={elRefs.current[i]} style={...}> ... </div> ))} </div> );

```

GPT answer, Verb Count: 24 , Adjective Count: 9

GPT-4o Answer: Using Refs for an Array of Elements in React To handle an array of elements with refs, you can use a `useRef` hook to store a map of refs and dynamically assign them to elements during rendering. Here is an implementation: `const { useRef, useState, useEffect } = React; const App = () => { const refs = useRef([]); const [widths, setWidths] = useState([]); useEffect(() => { const calculatedWidths = refs.current.map(ref => ref.offsetWidth); setWidths(calculatedWidths); }, []); return (<div> {[1, 2, 3].map((el, index) => (<div key={index} ref={elRef => (refs.current[index] = elRef)} style={{ width: `${el * 100}px` }} > Width is: {widths[index]} </div>))} </div>); }; ReactDOM.render(<App />, document.getElementById("root"));` Explanation: 1. Use a `useRef` to store an array of refs (`refs.current`). 2. During rendering, assign refs dynamically to elements using `refs.current[index]`. 3. After mounting, use `useEffect` to calculate the widths of all elements and update the state with the measured widths. 4. This approach ensures each element has its own ref, and their dimensions can be managed effectively.

Co-Pilot, Verb Count: 9 , Adjective Count: 4

Co-Pilot Answer: To handle multiple refs for an array of elements in React, you can use the `useRef` hook to create an array of refs. Here's how you can do it: `const { useRef, useState, useEffect } = React; const App = () => { const refs = useRef([]); const [elWidths, setElWidths] = useState([]); useEffect(() => { setElWidths(refs.current.map(ref => ref.offsetWidth)); }, []); return ({[1, 2, 3].map((el, index) => (key={index} ref={el => (refs.current[index] = el)} style={{ width: `${el * 100}px` }} > Width is: {elWidths[index]}))}); }; ReactDOM.render(, document.getElementById("root"));`

Question 2:

Tags: javascript, reactjs, react-hooks

Question: How to push element inside useState array React hook? Is that as an old method in react state? Or something new? E.g. setState push example ?

Answer Score: 809

Stack Overflow answer, Verb Count: 28 , Adjective Count: 27

Stack Overflow Answer: When you use `useState`, you can get an update method for the state item: `const [theArray, setTheArray] = useState(initialArray);` then, when you want to add a new element, you use that function and pass in the new array or a function that will create the new array. Normally the latter, since state updates are asynchronous and sometimes batched: `setTheArray(oldArray => [...oldArray, newElement]);` Sometimes you can get away without using that callback form, if you only update the array in handlers for certain specific user events

like `click` (but not like `mousemove`): `setTheArray([...theArray, newElement]);` The events for which React ensures that rendering is flushed are the "discrete events" listed here. Live Example (passing a callback into `setTheArray`):

```
const {useState, useCallback} = React;
function Example() { const [theArray, setTheArray] = useState([]); const
addEntryClick = () => { setTheArray(oldArray => [...oldArray, `Entry
${oldArray.length}`]); }; return [ <input type="button"
onClick={addEntryClick} value="Add" />, <div>{theArray.map(entry =>
<div>{entry}</div> )} </div> ]; } ReactDOM.render( <Example />,
document.getElementById("root") ); <div id="root"></div> <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.8.1/umd/react.production.min.js"></script> <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.8.1/umd/react-dom.production.min.js"></script>
Because the only update to theArray in there is the one in a click event (one of the "discrete" events), I could get away with a direct update in addEntry:
```

```
const {useState, useCallback} = React; function Example() { const
[theArray, setTheArray] = useState([]); const addEntryClick = () => {
setTheArray([...theArray, `Entry ${theArray.length}`]); }; return [ <input
type="button" onClick={addEntryClick} value="Add" />,
<div>{theArray.map(entry => <div>{entry}</div> )} </div> ]; }
ReactDOM.render( <Example />, document.getElementById("root") ); <div
id="root"></div> <script src="https://cdnjs.cloudflare.com/ajax/libs/react/16.8.1/umd/react.production.min.js"></script> <script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/16.8.1/umd/react-dom.production.min.js"></script>
```

GPT answer, Verb Count: 23 , Adjective Count: 13

GPT-4o Answer: Pushing Elements into a `useState` Array in React To push an element into a `useState` array, you should not directly modify the array (e.g., using `push`), as React state must remain immutable. Instead, create a new array with the updated elements and pass it to the state updater function. Here is the correct way to append an element:

```
import React, { useState }
from "react"; const Example = () => { const [items, setItems] =
useState([]); const addItem = () => { setItems(prevItems => [...prevItems,
"New Item"]); }; return ( <div> <button onClick={addItem}>Add Item</button>
<ul> {items.map((item, index) => ( <li key={index}>{item}</li> ))} </ul>
</div> ); }; export default Example; Explanation: 1. Use the spread operator (...) to
copy the existing array and add the new element. 2. Avoid mutating the state directly, as it can lead to
unexpected behavior. 3. The setItems function ensures that the new array replaces the old state,
maintaining immutability.
```

Co-Pilot, Verb Count: 10 , Adjective Count: 5

Co-Pilot Answer: To push an element inside a `useState` array in a React hook, you should create a new array with the new element added, instead of modifying the existing state array directly. This ensures immutability. Here's how you can do it:

```
const [myArray, setMyArray] =
useState([]); const addElement = (newElement) => { setMyArray(prevArray =>
[...prevArray, newElement]); }; addElement(4); // Example to add the number
4 to the array
```

Question 3:

Tags: javascript, state-machine, redux, xstate

Question: I am working on investigation of one front-end application of medium complexity. At this moment it is written in pure javascript, it has a lot of different event-based messages connecting few main parts of this application. We decided that we need to implement some kind of state container for

this application in scope of further refactoring. Previously I had some experience with redux and ngrx store (which actually follows the same principles). Redux is an option for us, but one of the developers proposed using a state-machine based library, in particular the xstate library. I've never worked with xstate, so I found it interesting and started reading documentation and looking at different examples. Looked promising and powerful, but at some point I understood that I don't see any significant difference between it and redux. I spent hours trying to find an answer, or any other information comparing xstate and redux. I didn't find any clear information, except some articles like "get from redux to a state machine", or links to libraries focused on using redux and xstate together (quite weird). If someone can describe the difference or tell me when developers should choose xstate - you are welcome to.

Answer Score: 392

Stack Overflow answer, Verb Count: 44 , **Adjective Count:** 34

Stack Overflow Answer: I created XState, but I'm not going to tell you whether to use one over the other; that depends on your team. Instead, I'll try to highlight some key differences. Redux XState essentially a state container where events (called actions in Redux) are sent to a reducer which update state also a state container, but separates finite state (e.g., "loading", "success") from "infinite state", or context (e.g., `items: [...]`) does not dictate how you define your reducers - they are plain functions that return the next state given the current state and event (action) a "reducer with rules" - you define legal transitions between finite states due to events, and also which actions should be executed in a transition (or on entry/exit from a state) does not have a built-in way to handle side-effects; there are many community options, like redux-thunk, redux-saga, etc. makes actions (side-effects) declarative and explicit - they are part of the `State` object that is returned on each transition (current state + event) currently has no way to visualize transitions between states, since it does not discern between finite and infinite state has a visualizer: <https://statecharts.github.io/xstate-viz> which is feasible due to the declarative nature the implicit logic/behavior represented in reducers can't be serialized declaratively (e.g., in JSON) machine definitions, which represent logic/behavior, can be serialized to JSON, and read from JSON; this makes behavior very portable and configurable by external tools not strictly a state machine adheres strictly to the W3C SCXML specification: <https://www.w3.org/TR/scxml/> relies on the developer to manually prevent impossible states uses statecharts to naturally define boundaries for handling events, which prevents impossible states and can be statically analyzed encourages the use of a single, "global" atomic store encourages the use of an Actor-model-like approach, where there can be many hierarchical statechart/"service" instances that communicate with each other I will add more key differences to the docs this week.

GPT answer, Verb Count: 50 , **Adjective Count:** 47

GPT-4o Answer: Comparison of Redux and XState 1. Core Concepts: Redux: Redux is a state management library based on the principles of a unidirectional data flow, immutability, and centralized global state. State changes are handled through actions and reducers. XState: XState is a state-machine and statechart library that models complex application states as finite states with explicit transitions and side effects. It uses statecharts to define behavior over time, which is ideal for managing state with defined workflows and transitions. 2. Use Cases: Redux: Ideal for managing global application state, such as user data, settings, or data fetched from APIs. Its simplicity and predictable flow make it great for apps requiring straightforward state management. XState: Better suited for applications with complex state transitions, workflows, or behaviors that can be explicitly modeled (e.g., multi-step forms, UI wizards, or media players). 3. State Representation: Redux: Represents state as a single, immutable object. State changes are determined by reducers, which are pure functions that update state based on actions. XState: Represents state as finite states with transitions. It supports hierarchical states, parallel states, and transient states, allowing for more expressive and visual state modeling. 4. Side Effects: Redux: Side effects (e.g., API calls) are handled externally using middleware like Redux Thunk or Redux Saga. XState: Supports built-in side effects through actions directly defined in state transitions, simplifying integration with external systems. 5. Learning Curve: Redux: Easier to

learn for basic use cases but can become verbose and complex as the application grows. XState: Has a steeper learning curve due to its state-machine concepts but provides a clearer and more maintainable structure for complex state management. 6. Visualization: Redux: No built-in visualization tools for state transitions. XState: Offers tools like the XState visualizer, which helps developers model and debug state transitions visually. 7. Choosing Between Redux and XState: 1. Use Redux if: - Your application requires centralized state management with global access. - State transitions are simple and don't need to be explicitly modeled. - You are already familiar with Redux or have a team experienced with it. 2. Use XState if: - Your application involves complex workflows, multi-step processes, or intricate UI states. - You want to explicitly define and visualize state transitions. - Your application requires features like hierarchical or parallel states. In summary, choose Redux for simplicity and global state management, and choose XState when complex state transitions and workflows need to be explicitly modeled and managed.

Co-Pilot, Verb Count: 41 , Adjective Count: 30

Co-Pilot Answer: When deciding between Redux and XState for state management, it's important to understand their key differences and use cases: Redux: Redux is based on the Flux architecture pattern and follows a unidirectional data flow. It uses actions, reducers, and a single store to manage the state of the application. Redux is well-suited for applications with complex state management needs and has a strong community and ecosystem. It provides tools like Redux DevTools for debugging and integrates seamlessly with libraries like React. However, handling temporal behavior and parallel states requires additional middleware like Redux-Saga or Redux-Thunk. XState: XState is a library for creating state machines and statecharts. It models states, events, and transitions in a more declarative way, making it easier to manage complex state logic. XState has built-in support for handling time-based events, parallel states, and robust error handling. It provides a visualizer for debugging and understanding state transitions. XState is particularly useful for applications with complex state transitions and temporal behavior. When to choose XState: If your application involves complex state transitions, temporal behavior, parallel states, or requires robust error handling, XState might be a better choice. It provides a more structured and formalized approach to state management. When to choose Redux: If your application has a simpler state management requirement or you prefer a well-established and widely-used solution with strong community support, Redux might be more suitable. It is also easier to integrate with libraries like React and has extensive debugging tools. Ultimately, the choice depends on the specific needs and complexity of your application. Both libraries have their strengths and can be used effectively depending on the context.

Question 4:

Tags: python, mypy, python-typing

Question: I have a function that takes a tuple of different lengths as an argument: `from typing import Tuple def process_tuple(t: Tuple[str]): # Do nasty tuple stuff process_tuple(("a",)) process_tuple(("a", "b")) process_tuple(("a", "b", "c"))` When I annotate function like mentioned above, I get these error messages `fool.py:9: error: Argument 1 to "process_tuple" has incompatible type "Tuple[str, str]"; expected "Tuple[str]" fool.py:10: error: Argument 1 to "process_tuple" has incompatible type "Tuple[str, str, str]"; expected "Tuple[str]"` `process_tuple` really works with tuples and I use them as immutable lists of variable length. I haven't found any consensus on this topic on the internet, so I wonder how should I annotate this kind of input.

Answer Score: 283

Stack Overflow answer, Verb Count: 8 , Adjective Count: 15

Stack Overflow Answer: We can annotate variable-length homogeneous tuples using the `...` literal (aka [Ellipsis](#)) like this: `def process_tuple(t: Tuple[str, ...]): ...` or for Python3.9+ `def process_tuple(t: tuple[str, ...]): ...`. After that, the errors should go away. From the docs: To specify a variable-length tuple of homogeneous type, use literal ellipsis, e.g. `Tuple[int, ...]`. A plain `Tuple` is equivalent to `Tuple[Any, ...]`, and in turn to `tuple`.

GPT answer, Verb Count: 18 , Adjective Count: 15

GPT-4o Answer: Annotating Tuples of Variable Length in Python To annotate a function parameter that accepts a tuple of variable length, you can use the `typing` module's `Tuple` type along with an ellipsis (`...`), which indicates that the tuple can contain a variable number of elements of the specified type. Here is the corrected annotation: `from typing import Tuple def process_tuple(t: Tuple[str, ...]): # Do nasty tuple stuff print(t) process_tuple(("a",)) # Valid process_tuple(("a", "b")) # Valid process_tuple(("a", "b", "c")) # Valid` Explanation: 1. The `Tuple[str, ...]` annotation specifies that the function accepts a tuple with elements of type `str`, and the tuple can have any length. 2. This avoids the type errors encountered when passing tuples of varying lengths to the function. 3. The ellipsis (`...`) is a shorthand to indicate a homogeneous tuple of indeterminate length.

Co-Pilot, Verb Count: 8 , Adjective Count: 8

Co-Pilot Answer: To handle tuples of different lengths, you should use `Tuple` with an ellipsis. This allows the function to accept tuples of any length where all elements are of the specified type. `from typing import Tuple def process_tuple(t: Tuple[str, ...]): # Do nasty tuple stuff process_tuple(("a",)) process_tuple(("a", "b")) process_tuple(("a", "b", "c"))` This way, the function will accept tuples of variable lengths with string elements.

Question 5:

Tags: javascript, reactjs, react-hooks

Question: I was going through the hooks documentation when I stumbled upon `useRef`. Looking at their example, `function TextInputWithFocusButton() { const inputEl = useRef(null); const onClick = () => { // `current` points to the mounted text input element inputEl.current.focus(); }; return (<> <input ref={inputEl} type="text" /> <button onClick={onClick}>Focus the input</button> </>); }`, it seems like `useRef` can be replaced with `createRef`. `function TextInputWithFocusButton() { const inputRef = createRef(); // what's the diff? const onClick = () => { // `current` points to the mounted text input element inputRef.current.focus(); }; return (<> <input ref={inputRef} type="text" /> <button onClick={onClick}>Focus the input</button> </>); }` Why do I need a hook for refs? Why does `useRef` exist?

Answer Score: 311

Stack Overflow answer, Verb Count: 18 , Adjective Count: 9

Stack Overflow Answer: The difference is that `createRef` will always create a new ref. In a class-based component, you would typically put the ref in an instance property during construction (e.g. `this.input = createRef()`). You don't have this option in a function component. `useRef` takes care of returning the same ref each time as on the initial rendering. Here's an example app demonstrating the difference in the behavior of these two functions: `import React, { useRef, createRef, useState } from "react"; import ReactDOM from "react-dom"; function App() { const [renderIndex, setRenderIndex] = useState(1); const refFromUseRef = useRef(); const refFromCreateRef = createRef(); if (!refFromUseRef.current) { refFromUseRef.current = renderIndex; } if (!refFromCreateRef.current) { refFromCreateRef.current = renderIndex; }`

```
return ( <div className="App"> Current render index: {renderIndex} <br />
First render index remembered within useRef.current:
{refFromUseRef.current} <br /> First render index unsuccessfully remembered
within refFromCreateRef.current: {refFromCreateRef.current} <br /> <button
onClick={() => setRenderIndex(prev => prev + 1)}> Cause re-render </button>
</div> ); } const rootElement = document.getElementById("root");
ReactDOM.render(<App />, rootElement);
```

GPT answer, Verb Count: 37 , Adjective Count: 39

GPT-4o Answer: Difference Between `useRef` and `createRef`

1. Core Difference: `useRef` is a React hook used in functional components, while `createRef` is primarily used in class components. The key difference lies in how they behave during re-renders.
2. Behavior During Re-Renders: `useRef`: The same `ref` object persists across re-renders. This makes it ideal for storing a reference to a DOM element or a mutable value that should not trigger re-renders.

```
function FunctionalComponent() {
  const inputRef = useRef(null);
  useEffect(() => { console.log("inputRef persists across renders", inputRef.current); });
  return <input ref={inputRef} />;
}
```

`createRef`: A new `ref` object is created every time the component re-renders. This makes it unsuitable for functional components where the reference needs to persist across renders.

```
class ClassComponent extends React.Component {
  constructor() {
    super();
    this.inputRef = React.createRef();
  }
  render() {
    return <input ref={this.inputRef} />;
  }
}
```
3. Why Does `useRef` Exist? `useRef` exists to provide a way to manage mutable references in functional components. It allows you to:
 1. Access a DOM element directly without triggering re-renders.
 2. Persist values across renders without storing them in state.
4. When to Use: `useRef`: Use in functional components to maintain a persistent reference or store mutable data that does not cause re-renders. `createRef`: Use in class components or in specific scenarios where a new `ref` is required for every render.

Conclusion: `useRef` is tailored for functional components, offering better performance and functionality compared to `createRef` in such cases. It simplifies `ref` handling in the modern React ecosystem.

Co-Pilot, Verb Count: 10 , Adjective Count: 15

Co-Pilot Answer: `useRef` and `createRef` serve different purposes in React. `createRef` is typically used in class components. Every time the component renders, `createRef` creates a new `ref`.

```
function TextInputWithFocusButton() {
  const inputRef = createRef(); // what's the diff?
  const onClick = () => { // current points to the mounted text input element
    inputRef.current.focus();
  };
  return ( <> Focus the input );
}
```

`useRef` is a hook and is used in functional components. It persists the `ref` across renders, which is crucial for maintaining the same `ref` object throughout the component's lifecycle.

```
function TextInputWithFocusButton() {
  const inputEl = useRef(null);
  const onClick = () => { // current points to the mounted text input element
    inputEl.current.focus();
  };
  return ( <> Focus the input );
}
```

Month: Mar-19

Question 1:

Tags: javascript, reactjs, react-hooks

Question: I am using functional component with hooks. I need to update state in parent from a child. I am using a prop function in Parent. All works fine except my prop function is getting the previous state and not the current state. My prop function gets executed before `useState` hook setting current state. How can I wait for my call back function to be executed after `useState` call. I am looking for something like `setState(state,callback)` from class based components. Here is the code snippet:

```
function Parent() {
  const [Name, setName] = useState("");
  getChildChange =
```

```

getChildChange.bind(this); function getChildChange(value) { setName(value);
} return <div> {Name} : <Child getChildChange={getChildChange} ></Child>
</div> } function Child(props) { const [Name, setName] = useState("");
handleChange = handleChange.bind(this); function handleChange(ele) {
setName(ele.target.value); props.getChildChange(collectState()); } function
collectState() { return Name; } return (<div> <input onChange={handleChange}
value={Name}></input> </div>); }

```

Answer Score: 176

Stack Overflow answer, Verb Count: 14 , **Adjective Count:** 12

Stack Overflow Answer: You can use `useEffect/useLayoutEffect` to achieve this: `const SomeComponent = () => { const [count, setCount] = React.useState(0) React.useEffect(() => { if (count > 1) { document.title = 'Threshold of over 1 reached.'; } else { document.title = 'No threshold reached.'; } }, [count]); return (<div> <p>{count}</p> <button type="button" onClick={() => setCount(count + 1)}> Increase </button> </div>); }; If you want to prevent the callback from running on first render, adjust the previous version: const SomeComponent = () => { const [count, setCount] = React.useState(0) const didMount = React.useRef(false); React.useEffect(() => { if (!didMount.current) { didMount.current = true; return; } if (count > 1) { document.title = 'Threshold of over 1 reached.'; } else { document.title = 'No threshold reached.'; } }, [count]); return (<div> <p>{count}</p> <button type="button" onClick={() => setCount(count + 1)}> Increase </button> </div>); }; More about it over here.`

GPT answer, Verb Count: 17 , **Adjective Count:** 11

GPT-4o Answer: Handling State Updates with Callback in React Functional Components In functional components with hooks, there is no direct equivalent of the `setState(state, callback)` pattern from class components. However, you can achieve the desired behavior by using the `useEffect` hook to respond to state changes. Here is the corrected code snippet: `function Parent() { const [Name, setName] = useState(""); function getChildChange(value) { setName(value); } return (<div> {Name} : <Child getChildChange={getChildChange} /> </div>); } function Child(props) { const [Name, setName] = useState(""); useEffect(() => { // Trigger the parent's callback when the state changes props.getChildChange(Name); }, [Name, props]); function handleChange(ele) { setName(ele.target.value); } return (<div> <input onChange={handleChange} value={Name} /> </div>); } Explanation: 1. The useEffect hook in the Child component watches the Name state. 2. Whenever Name changes, the parent's getChildChange function is called with the updated state value. 3. This ensures that the parent's callback is executed after the state update, addressing the issue of stale state in the callback.`

Co-Pilot, Verb Count: 11 , **Adjective Count:** 3

Co-Pilot Answer: To ensure that the parent state updates after the child state has been set, you can use a `useEffect` hook in the child component. This will call the prop function after the state update has occurred. `function Parent() { const [Name, setName] = useState(""); function getChildChange(value) { setName(value); } return {Name} : ; } function Child(props) { const [Name, setName] = useState(""); useEffect(() => { props.getChildChange(Name); }, [Name]); function handleChange(ele) { setName(ele.target.value); } return (); }`

Question 2:

Tags: javascript, automated-tests, e2e-testing, cypress

Question: I want to toggle running only one test, so that I don't have to wait for my other tests to finish to see the result of the one test I'm interested in. Currently, I comment out my other tests but this is really annoying. Is there a way to toggle only running one test in [Cypress](#)?

Answer Score: 420

Stack Overflow answer, Verb Count: 21 , **Adjective Count:** 16

Stack Overflow Answer: to run only one file `cypress run --spec path/to/file.spec.js` or using glob patterns: `cypress run --spec 'path/to/files/*.spec.js'` Note: you need to wrap your glob patterns in single quotes to avoid shell expansion! to run only one test in a file You can use a `.only` as described in the Cypress docs `it.only('only run this one', () => { // similarly use it.skip(...) to skip a test }) it('not this one', () => { })` Also, you can do the same with `describe` and `context` blocks edit: there's also a nice [VSCode](#) extension to make adding/removing `.only`'s easier with keyboard shortcuts. It's called Test Utils (install with `ext install chrisbreiding.test-utils`). It works with js, coffee, and typescript:

GPT answer, Verb Count: 21 , **Adjective Count:** 11

GPT-4o Answer: Toggling to Run Only One Test in Cypress In Cypress, you can use the `.only` method to run a specific test or suite of tests. This allows you to isolate a single test or a group of tests without commenting out others. Running a Single Test: `it.only('should run this specific test', () => { // Test code here });` Running a Single Test Suite: `describe.only('should run this specific test suite', () => { it('test 1', () => { // Test code here }); it('test 2', () => { // Test code here }); });` Important Notes: 1. Using `.only` ensures that only the marked test or suite runs, while all others are ignored. 2. Remember to remove `.only` before committing your code to avoid unintentionally skipping other tests in CI/CD pipelines.

Co-Pilot, Verb Count: 13 , **Adjective Count:** 8

Co-Pilot Answer: To run a single test in Cypress without having to comment out other tests, you can use the `--spec` flag in the command line to specify the test file you want to run. Here's how you can do it: `npx cypress run --spec "cypress/e2e/my-spec.cy.js"` Replace `"cypress/e2e/my-spec.cy.js"` with the path to your specific test file. This way, only the specified test will run, and you won't need to comment out other tests.

Question 3:

Tags: javascript, reactjs, next.js

Question: In my Next.js app I can't seem to access `window`: Unhandled Rejection (ReferenceError): window is not defined `componentWillMount() { console.log('window.innerHeight', window.innerHeight); }`

Answer Score: 102

Stack Overflow answer, Verb Count: 13 , **Adjective Count:** 17

Stack Overflow Answer: Move the code from `componentWillMount()` to `componentDidMount()`: `componentDidMount() { console.log('window.innerHeight', window.innerHeight); }` In Next.js, `componentDidMount()` is executed only on the client where `window` and other browser specific APIs will be available. From the Next.js wiki: Next.js is universal, which means it executes code first server-side, then client-side. The window object is only present client-side, so if you absolutely need to have access to it in some React component, you should put that code in `componentDidMount`. This lifecycle method will only be executed on the client. You may also want to check if there isn't some alternative universal library which may suit your needs. Along

the same lines, `componentWillMount()` will be deprecated in v17 of React, so it effectively will be potentially unsafe to use in the very near future.

GPT answer, Verb Count: 26 , Adjective Count: 14

GPT-4o Answer: Handling "window is not defined" in Next.js The error occurs because Next.js runs both on the server (during SSR) and the client. The `window` object is only available on the client side, not on the server side. To resolve this issue, you need to ensure that code accessing `window` only runs in the browser. Solution: Check if the code is running on the client side before accessing `window`. You can use `useEffect` in functional components or lifecycle methods in class components. Functional Component Example:

```
import { useEffect } from "react"; const MyComponent = () => {
  useEffect(() => { console.log("window.innerHeight", window.innerHeight); }, []);
  return <div>My Component</div>;
}; export default MyComponent;
```

 Class Component Example:

```
import React from "react"; class MyComponent extends React.Component {
  componentDidMount() { console.log("window.innerHeight", window.innerHeight); }
  render() { return <div>My Component</div>; }
} export default MyComponent;
```

 Explanation: 1. The `useEffect` hook in functional components or `componentDidMount` in class components ensures the code runs only after the component has been mounted on the client side. 2. Avoid accessing `window` or other browser-specific objects during server-side rendering to prevent the error.

Co-Pilot, Verb Count: 22 , Adjective Count: 9

Co-Pilot Answer: The issue arises because `window` is only available in the browser environment, not on the server side where Next.js also runs during the initial render. To access `window` in a Next.js application, you need to ensure your code runs only in the browser. You can achieve this by using a lifecycle method that executes in the browser or by checking if the code is running on the client side. Here's an example using `componentDidMount`:

```
import React from 'react'; class MyComponent extends React.Component {
  componentDidMount() { console.log('window.innerHeight', window.innerHeight); }
  render() { return <div>Check the console for window height</div>; }
} export default MyComponent;
```

 Alternatively, you can use a functional component with `useEffect`:

```
import React, { useEffect } from 'react'; const MyComponent = () => {
  useEffect(() => { console.log('window.innerHeight', window.innerHeight); }, []);
  return <div>Check the console for window height</div>;
}; export default MyComponent;
```

Question 4:

Tags: python, google-chrome, anaconda, jupyter-notebook

Question: I've been running a script on jupyter notebooks for about 26 hour; I haven't really been using my computer for anything else, but it needs to run this program that will take ~30 hours to complete. At about 21 hours in, it stopped saving and my terminal had this: `403 PUT /api/contents/[file.ipynb] (:::1): '_xsrftoken' argument missing from POST` where `[file.ipynb]` is the location of my jupyter notebook. It also says: `'_xsrftoken' argument missing from post` in the top right part of the notebook again. The program is still running and I don't want to restart jupyter notebook and have to run the program again, as I have a deadline, is there anything else I can do? I'm using google chrome, but I don't have the LastPass extension or any '%' characters in my code, as another post suggested. Thanks for any help!

Answer Score: 761

Stack Overflow answer, Verb Count: 8 , Adjective Count: 7

Stack Overflow Answer: The easiest way I found is this: <https://github.com/nteract/hydrogen/issues/922#issuecomment-405456346> Just open another (non-running, existing) notebook on the same kernel, and the issue is magically gone; you can again

save the notebooks that were previously showing the `_xsrf` error. If you have already closed the Jupyter home page, you can find a link to it on the terminal from which Jupyter was started.

GPT answer, Verb Count: 42 , Adjective Count: 19

GPT-4o Answer: Dealing with Jupyter Notebook "`_xsrf`" Argument Missing Error This issue typically occurs when the Jupyter Notebook server loses its session authentication due to inactivity or browser-related issues. Since your script is still running, you can take steps to resolve the issue without restarting the kernel or losing progress. Steps to Resolve: 1. Try refreshing the browser tab where your Jupyter Notebook is running. This can re-establish the session with the server. 2. If refreshing doesn't work, open the Jupyter Notebook server URL in a new browser tab or window, ensuring the URL matches the running server instance. Log back in if prompted. 3. If the problem persists, disable Cross-Site Request Forgery (XSRF) protection temporarily by restarting the Jupyter Notebook server with the following flag: `jupyter notebook --NotebookApp.disable_check_xsrf=True` Warning: Disabling XSRF protection can expose your server to security risks. Use this method only as a last resort and if your Jupyter Notebook is running on a secure, local environment. Preventive Measures: 1. Avoid long periods of browser inactivity while running long scripts by periodically interacting with the notebook. 2. Use the `nohup` command or a screen session to run long-running scripts outside the browser to avoid dependency on the browser session: `nohup jupyter notebook &` Next Steps: Once your script completes, restart the Jupyter Notebook server to re-enable XSRF protection if it was temporarily disabled.

Co-Pilot, Verb Count: 13 , Adjective Count: 5

Co-Pilot Answer: Try opening another existing notebook on the same kernel, then go back to the problematic notebook and try saving it again. If that doesn't work, visit the `/tree` endpoint (e.g., <http://localhost:8888/tree>), refresh your browser, and reopen the kernel. If the issue persists, try opening the notebook in a new browser tab or window.

Question 5:

Tags: python, keras, tensorflow, tensorflow2.0

Question: When I am executing the command `sess = tf.Session()` in Tensorflow 2.0 environment, I am getting an error message as below: `Traceback (most recent call last): File "<stdin>", line 1, in <module> AttributeError: module 'tensorflow' has no attribute 'Session'` System Information: OS Platform and Distribution: Windows 10 Python Version: 3.7.1 Tensorflow Version: 2.0.0-alpha0 (installed with pip) Steps to reproduce: Installation: `pip install --upgrade pip` `pip install tensorflow==2.0.0-alpha0` `pip install keras` `pip install numpy==1.16.2` Execution: Execute command: `import tensorflow as tf` Execute command: `sess = tf.Session()`

Answer Score: 404

Stack Overflow answer, Verb Count: 7 , Adjective Count: 6

Stack Overflow Answer: According to [TF 1:1 Symbols Map](#), in TF 2.0 you should use `tf.compat.v1.Session()` instead of `tf.Session()` https://docs.google.com/spreadsheets/d/1FLFJLzg7WNP6JHODX5q8BDgptKafq_slHpnHVbJlIteQ/edit#gid=0 To get TF 1.x like behaviour in TF 2.0 one can run `import tensorflow.compat.v1 as tf` `tf.disable_v2_behavior()` but then one cannot benefit of many improvements made in TF 2.0. For more details please refer to the migration guide <https://www.tensorflow.org/guide/migrate>

GPT answer, Verb Count: 24 , Adjective Count: 12

GPT-4o Answer: Resolving the "No Attribute 'Session'" Error in TensorFlow 2.0 The error occurs because TensorFlow 2.0 has removed the `Session` object and moved to an eager execution model by

default. This change makes it easier to work with TensorFlow without requiring session management. Solution: If you're migrating from TensorFlow 1.x and need to use code that requires `Session`, you can enable TensorFlow 1.x compatibility mode. # Enable TensorFlow 1.x compatibility mode `import tensorflow.compat.v1 as tf; tf.disable_v2_behavior()` # Now you can use `tf.Session()` `sess = tf.Session()` Recommended Approach in TensorFlow 2.x: Instead of using `Session`, take advantage of TensorFlow 2.x's eager execution, which eliminates the need for sessions. Here's an example: `import tensorflow as tf` # TensorFlow 2.x eager execution example `a = tf.constant(2)` `b = tf.constant(3)` `c = a + b` `print(c.numpy())` # Outputs: 5 Key Notes: 1. If you are using TensorFlow 2.x, it's best to adapt your code to use eager execution and avoid `Session`. 2. For legacy TensorFlow 1.x code, use the compatibility mode shown above. 3. Check TensorFlow documentation for guidance on migrating from 1.x to 2.x: <https://www.tensorflow.org/guide/migrate>

Co-Pilot, Verb Count: 11 , Adjective Count: 12

Co-Pilot Answer: In TensorFlow 2.0, the `Session` object is no longer used. TensorFlow 2.0 encourages the use of the `tf.function` decorator and eager execution, which is enabled by default. Here is how you can adapt your code for TensorFlow 2.0: `import tensorflow as tf` Define a simple computation: `a = tf.constant(2)` `b = tf.constant(3)` `c = a + b` Eager execution (default in TF 2.0): `print(c)` If you need to use `tf.function` to define a graph, you can do the following: `import tensorflow as tf` `@tf.function` `def add(a, b):` `return a + b` Now call the function `result = add(tf.constant(2), tf.constant(3))` `print(result)`

Month: Apr-19

Question 1:

Tags: javascript, reactjs, react-native, react-hooks

Question: I am following a Udemy course on how to register events with hooks, the instructor gave the below code: `const [userText, setUserText] = useState(''); const handleUserKeyPress = event => { const { key, keyCode } = event; if (keyCode === 32 || (keyCode >= 65 && keyCode <= 90)) { setUserText(`${userText}${key}`); } }; useEffect(() => { window.addEventListener('keydown', handleUserKeyPress); return () => { window.removeEventListener('keydown', handleUserKeyPress); } }; return (<div> <h1>Feel free to type!</h1> <blockquote>{userText}</blockquote> </div>);` Now it works great but I'm not convinced that this is the right way. The reason is, if I understand correctly, on each and every re-render, events will keep registering and deregistering every time and I simply don't think it is the right way to go about it. So I made a slight modification to the `useEffect` hooks to below `useEffect(() => { window.addEventListener('keydown', handleUserKeyPress); return () => { window.removeEventListener('keydown', handleUserKeyPress); } }, []);` By having an empty array as the second argument, letting the component to only run the effect once, imitating `componentDidMount`. And when I try out the result, it's weird that on every key I type, instead of appending, it's overwritten instead. I was expecting `setUserText(`${userText}${key}`);` to have new typed key append to current state and set as a new state but instead, it's forgetting the old state and rewriting with the new state. Was it really the correct way that we should register and deregister event on every re-render?

Answer Score: 183

Stack Overflow answer, Verb Count: 15 , Adjective Count: 15

Stack Overflow Answer: The best way to go about such scenarios is to see what you are doing in the

event handler. If you are simply setting `state` using previous `state`, it's best to use the callback pattern and register the event listeners only on initial mount. If you do not use the callback pattern, the listeners reference along with its lexical scope is being used by the event listener but a new function is created with updated closure on each render; hence in the handler you will not be able to access the updated state

```
const [userText, setUserText] = useState("");
const handleUserKeyPress = useCallback(event => { const { key, keyCode } = event;
if(keyCode === 32 || (keyCode >= 65 && keyCode <= 90)){
setUserText(prevUserText => `${prevUserText}${key}`); } }, []);
useEffect(() => { window.addEventListener("keydown", handleUserKeyPress); return () => {
window.removeEventListener("keydown", handleUserKeyPress); } }, [handleUserKeyPress]);
return ( <div> <h1>Feel free to type!</h1> <blockquote>{userText}</blockquote> </div> );
```

GPT answer, Verb Count: 38 , Adjective Count: 29

GPT-4o Answer: Correct Usage of Event Listeners with Hooks in React The behavior you observed is due to how closures in JavaScript interact with React state in functional components. The version of `handleUserKeyPress` being registered in the event listener is "stale" because it captures the initial value of `userText` when the `useEffect` is run. This is why appending does not work as expected when you use `useEffect` with an empty dependency array (`[]`). To resolve this, you can either:

1. Correctly Register and Deregister on Every Re-Render:

```
useEffect(() => { const handleUserKeyPress = event => { const { key, keyCode } = event; if (keyCode === 32 || (keyCode >= 65 && keyCode <= 90)) { setUserText(prevUserText => `${prevUserText}${key}`); } }; window.addEventListener('keydown', handleUserKeyPress); return () => { window.removeEventListener('keydown', handleUserKeyPress); } }, []);
```

 This ensures that the correct version of `handleUserKeyPress` is registered, as the function is redefined with the updated state on each render.
2. Use a Stable Event Listener with a Ref:

```
import { useState, useEffect, useRef } from 'react';
const App = () => { const [userText, setUserText] = useState(''); const userTextRef = useRef(userText);
useEffect(() => { userTextRef.current = userText; }, [userText]);
useEffect(() => { const handleUserKeyPress = event => { const { key, keyCode } = event; if (keyCode === 32 || (keyCode >= 65 && keyCode <= 90)) { setUserText(userTextRef.current + key); } };
window.addEventListener('keydown', handleUserKeyPress); return () => { window.removeEventListener('keydown', handleUserKeyPress); } }, []);
return ( <div> <h1>Feel free to type!</h1> <blockquote>{userText}</blockquote> </div> ); };
```

 Explanation: 1. In the first approach, the event listener is re-registered on every render, ensuring it has access to the latest state value. 2. In the second approach, a `useRef` is used to store the current value of `userText`. The event listener uses the ref, avoiding the need to re-register the listener on every render. Both approaches are valid, but the second approach can improve performance by avoiding unnecessary re-registrations.

Co-Pilot, Verb Count: 12 , Adjective Count: 17

Co-Pilot Answer: The issue arises because the `userText` state inside your event handler is stale and doesn't reflect the current state value when using `useEffect` with an empty dependency array. You should use a functional update to ensure you always have the latest state value:

```
const [userText, setUserText] = useState('');
const handleUserKeyPress = event => { const { key, keyCode } = event; if (keyCode === 32 || (keyCode >= 65 && keyCode <= 90)) {
setUserText(prevUserText => `${prevUserText}${key}`); } };
useEffect(() => { window.addEventListener('keydown', handleUserKeyPress); return () => {
window.removeEventListener('keydown', handleUserKeyPress); } }, []);
return ( <div> <h1>Feel free to type!</h1> <blockquote>{userText}</blockquote> </div> );
```

 This way, the `handleUserKeyPress` function will always use the most current state value when appending the new key.

Question 2:

Tags: javascript, reactjs, react-hooks

Question: I have a component that looks like this (very simplified version): `const component = (props: PropTypes) => { const [allResultsVisible, setAllResultsVisible] = useState(false); const renderResults = () => { return (<section> <p onClick={ setAllResultsVisible(!allResultsVisible) }> More results v </p> { allResultsVisible && <section className="entity-block--hidden-results"> ... </section> } </section>); }; return <div>{ renderResults() }</div>; }` When I load the page this component is used on, I get this error: `Uncaught Invariant Violation: Rendered more hooks than during the previous render.` I tried to find an explanation of this error, but my searching returned no results. When I modify the component slightly: `const component = (props: PropTypes) => { const [allResultsVisible, setAllResultsVisible] = useState(false); const handleToggle = () => { setAllResultsVisible(!allResultsVisible); } const renderResults = () => { return (<section> <p onClick={ handleToggle }> More results v </p> { allResultsVisible && <section className="entity-block--hidden-results"> ... </section> } </section>); }; return <div>{ renderResults() }</div>; }` I no longer get that error. Is it because I included the `useState` function within the jsx that is returned by `renderResults`? It would be great to have an explanation of why the fix works.

Answer Score: 64

Stack Overflow answer, Verb Count: 24 , **Adjective Count:** 16

Stack Overflow Answer: The fix works because the first code sample (the erroring one) invokes a function inside `onClick`, while the second (the working one) passes a function to `onClick`. The difference is those all-important parentheses, which in JavaScript mean 'invoke this code'. Think of it this way: in the first code sample, every time `component` is rendered, `renderResults` is invoked. Every time that happens, `setAllResultsVisible(!allResultsVisible)`, rather than waiting for a click, is called. Since React performs the render on its own schedule, there's no telling how many times that will happen. From the React docs: With JSX you pass a function as the event handler, rather than a string. React Handling Events Docs Note: I wasn't able to get this exact error message when running the first code sample in a sandbox. My error referred to an infinite loop. Maybe a more recent version of React produces the error described?

GPT answer, Verb Count: 32 , **Adjective Count:** 19

GPT-4o Answer: Understanding "Rendered More Hooks Than During the Previous Render" Error The error occurs because the way hooks are called is inconsistent between renders in your initial implementation. Hooks must be called in the same order on every render, and their structure must not change dynamically. Problem in the Initial Code: `<p onClick={ setAllResultsVisible(!allResultsVisible) }>` This code directly invokes the `setAllResultsVisible` function during the render phase. This means the `useState` hook is being used dynamically depending on how React processes the function, which violates the rules of hooks. Why the Fix Works: `const handleToggle = () => { setAllResultsVisible(!allResultsVisible); }` By moving the state update logic into a separate function (`handleToggle`), you ensure that the `setAllResultsVisible` function is only called when the `onClick` event occurs. This keeps the render phase pure and ensures that hooks are called consistently in every render. Key Points: 1. Hooks must always be called in the same order and at the top level of the component or custom hook. 2. Avoid calling state update functions directly inside JSX or during render. 3. Use event handlers to handle state updates or side effects. Summary: The error was caused because you were invoking the state update function (`setAllResultsVisible`) during the render phase, which violated the rules of hooks. By moving this logic to an event handler

([handleToggle](#)), you ensure consistent hook calls and fix the error.

Co-Pilot, Verb Count: 15 , Adjective Count: 13

Co-Pilot Answer: The error [Uncaught Invariant Violation: Rendered more hooks than during the previous render.](#) occurs because React Hooks must be called in the same order in every render. When you call [setAllResultsVisible\(!allResultsVisible\)](#) directly inside the JSX, it causes state updates that could potentially alter the hook order on subsequent renders, leading to this error. Your fix works because you moved the state update logic to a separate function ([handleToggle](#)) outside the JSX. This ensures that the hooks are called in the same order on every render, thus avoiding the violation.

```
const component = (props: PropTypes) => { const [allResultsVisible, setAllResultsVisible] = useState(false); const handleToggle = () => { setAllResultsVisible(!allResultsVisible); }; const renderResults = () => { return ( More results v { allResultsVisible && ... } ); }; return {renderResults()}; };
```

Question 3:

Tags: java, git, android, gitlab, android-studio

Question: I have one project on Gitlab and I worked with it for the last few days! Now i want [pull](#) project on my home PC but show me below error : [Invocation failed Unexpected Response from Server: Unauthorized java.lang.RuntimeException: Invocation failed Unexpected Response from Server: Unauthorized at org.jetbrains.git4idea.nativessh.GitNativeSshAskPassXmlRpcClient.handleInput\(GitNativeSshAskPassXmlRpcClient.java:34\) at org.jetbrains.git4idea.nativessh.GitNativeSshAskPassApp.main\(GitNativeSshAskPassApp.java:30\) Caused by: java.io.IOException: Unexpected Response from Server: Unauthorized at org.apache.xmlrpc LiteXmlRpcTransport.sendRequest\(LiteXmlRpcTransport.java:231\) at org.apache.xmlrpc LiteXmlRpcTransport.sendXmlRpc\(LiteXmlRpcTransport.java:90\) at org.apache.xmlrpc.XmlRpcClientWorker.execute\(XmlRpcClientWorker.java:72\) at org.apache.xmlrpc.XmlRpcClient.execute\(XmlRpcClient.java:194\) at org.apache.xmlrpc.XmlRpcClient.execute\(XmlRpcClient.java:185\) at org.apache.xmlrpc.XmlRpcClient.execute\(XmlRpcClient.java:178\)](#) My android studio version is 3.4 !

Answer Score: 66

Stack Overflow answer, Verb Count: 6 , Adjective Count: 3

Stack Overflow Answer: Managed to fix it like this: AndroidStudio -> Preferences -> Git -> SSH Executable and changed from [Native](#) to [Built-in](#) and it started working. Edit: this is for Android Studio 3.4, for 3.6 Volodymyr has an answer below.

GPT answer, Verb Count: 41 , Adjective Count: 33

GPT-4o Answer: Resolving "Unexpected Response from Server: Unauthorized" Error on GitLab Pull This error indicates an authentication issue between your GitLab repository and your local machine. The following steps should help resolve the problem: 1. Check Your GitLab Credentials: [Ensure you are using the correct username and personal access token \(PAT\) instead of a password. Starting from August 2021, GitLab requires a PAT for HTTPS-based operations.](#) 2. Update Your Remote URL: [# Check the current remote URL git remote -v # Update the remote URL with your username and PAT git remote set-url origin https://:@gitlab.com/.git Replace <username>, <personal_access_token>, <group>, and <repository> with your actual credentials and repository details.](#) 3. Reauthenticate in Android Studio: 1. Open Android Studio and go to "File" → "Settings" → "Version Control" → "Git". 2. Under "SSH executable", select "Native". 3. If you are

using HTTPS, clear stored credentials by going to "File" → "Settings" → "Appearance & Behavior" → "System Settings" → "Passwords". Then reauthenticate when prompted during the next pull. 4. Use SSH Authentication: If you prefer SSH over HTTPS: # Generate an SSH key (if not already created) `ssh-keygen -t rsa -b 4096 -C "your_email@example.com"` # Add the SSH key to your GitLab account `cat ~/.ssh/id_rsa.pub` Copy the output and add it to GitLab under "Settings" → "SSH Keys". Update your remote URL to use SSH: `git remote set-url origin git@gitlab.com:/.git` 5. Test the Connection: `git pull` If the error persists, double-check your credentials, access token, and SSH key configurations. Ensure your token has the required permissions for repository access (e.g., "read_repository" for pull).

Co-Pilot, Verb Count: 33 , Adjective Count: 14

Co-Pilot Answer: The error you're encountering suggests an authorization issue when attempting to pull the project from GitLab. Here are a few steps to resolve this: 1. Ensure Correct Credentials: Verify that you are using the correct username and password or personal access token for authentication. 2. Update Git Remote URL: Use the correct Git remote URL format, including the username. For example: `git remote set-url origin https://@gitlab.com/.git` 3. SSH Key Authentication: If you are using SSH, ensure that your SSH key is correctly added to your GitLab account. Generate a new SSH key if needed: `ssh-keygen -t rsa -b 4096 -C "your_email@example.com"` 4. Re-authenticate in Android Studio: You may need to re-authenticate Git credentials in Android Studio. Go to File > Settings > Version Control > Git, and update your credentials. 5. Check Git Configuration: Ensure that your Git configuration is correctly set up. Run the following commands to check: `git config --global user.name "Your Name"` `git config --global user.email "your_email@example.com"` After performing these steps, try pulling the project again.

Question 4:

Tags: javascript, string, algorithm

Question: I have to create a function which takes a string, and it should return `true` or `false` based on whether the input consists of a repeated character sequence. The length of the given string is always greater than 1 and the character sequence must have at least one repetition. "aa" // true(entirely contains two strings "a") "aaa" //true(entirely contains three string "a") "abcabcabc" //true(entirely contains three strings "abc") "aba" //false(At least there should be two same substrings and nothing more) "ababa" //false("ab" exists twice but "a" is extra so false) I have created the below function: `function check(str){ if(!(str.length && str.length - 1)) return false; let temp = ''; for(let i = 0;i<=str.length/2;i++){ temp += str[i] //console.log(str.replace(new RegExp(temp,"g"),'')) if(!str.replace(new RegExp(temp,"g"),'')) return true; } return false; }` `console.log(check('aa')) //true console.log(check('aaa')) //true console.log(check('abcabcabc')) //true console.log(check('aba')) //false console.log(check('ababa')) //false` Checking of this is part of the real problem. I can't afford a non-efficient solution like this. First of all, it's looping through half of the string. The second problem is that it is using `replace()` in each loop which makes it slow. Is there a better solution regarding performance?

Answer Score: 190

Stack Overflow answer, Verb Count: 41 , Adjective Count: 24

Stack Overflow Answer: There's a nifty little theorem about strings like these. A string consists of the same pattern repeated multiple times if and only if the string is a nontrivial rotation of itself. Here, a rotation means deleting some number of characters from the front of the string and moving them to the back. For example, the string `hello` could be rotated to form any of these strings: `hello` (the

trivial rotation) elloh llohe lohel ohell To see why this works, first, assume that a string consists of k repeated copies of a string w . Then deleting the first copy of the repeated pattern (w) from the front of the string and tacking it onto the back will give back the same string. The reverse direction is a bit trickier to prove, but the idea is that if you rotate a string and get back what you started with, you can apply that rotation repeatedly to tile the string with multiple copies of the same pattern (that pattern being the string you needed to move to the end to do the rotation). Now the question is how to check whether this is the case. For that, there's another beautiful theorem we can use: If x and y are strings of the same length, then x is a rotation of y if and only if x is a substring of yy . As an example, we can see that `lohel` is a rotation of `hello` as follows: `hellohello ^^^^^` In our case, we know that every string x will always be a substring of xx (it'll appear twice, once at each copy of x). So basically we just need to check if our string x is a substring of xx without allowing it to match at the first or halfway character. Here's a one-liner for that: `function check(str) { return (str + str).indexOf(str, 1) !== str.length; }` Assuming `indexOf` is implemented using a fast string matching algorithm, this will run in time $O(n)$, where n is the length of the input string.

GPT answer, Verb Count: 24 , Adjective Count: 18

GPT-4o Answer: Efficient Solution for Checking Repeated Character Sequences in a String You can solve this problem efficiently by using string concatenation and the properties of substrings. Here is the optimized implementation: `function check(str) { const doubled = str + str; const sliced = doubled.slice(1, doubled.length - 1); return sliced.includes(str); }` `console.log(check('aa'));` // true `console.log(check('aaa'));` // true `console.log(check('abcabcabc'));` // true `console.log(check('aba'));` // false `console.log(check('ababa'));` // false **Explanation:** 1. Concatenate the string to itself to create a "doubled" string. For example, "abc" becomes "abcabc". 2. Remove the first and last characters from the doubled string. For example, "abcabc" becomes "bcab". 3. Check if the original string exists as a substring in the sliced string. If it does, the string consists of repeated patterns. **Why This Is Efficient:** 1. This solution avoids looping through half of the string and eliminates the need for regular expressions or `replace`. 2. The time complexity is $O(n)$, where n is the length of the string, as slicing and substring checks are linear operations.

Co-Pilot, Verb Count: 14 , Adjective Count: 11

Co-Pilot Answer: To create a more efficient solution, you can use the KMP (Knuth-Morris-Pratt) algorithm to find the smallest repeating substring. Here's a more optimized version: `function isRepeatedSubstring(s) { const n = s.length; const lps = Array(n).fill(0); let i = 1, len = 0; while (i < n) { if (s[i] === s[len]) { len++; lps[i] = len; i++; } else { if (len !== 0) { len = lps[len - 1]; } else { lps[i] = 0; i++; } } } const subLen = n - lps[n - 1]; return lps[n - 1] > 0 && n % subLen === 0; }` `console.log(isRepeatedSubstring('aa'));` // true `console.log(isRepeatedSubstring('aaa'));` // true `console.log(isRepeatedSubstring('abcabcabc'));` // true `console.log(isRepeatedSubstring('aba'));` // false `console.log(isRepeatedSubstring('ababa'));` // false This approach uses the KMP algorithm to preprocess the string and find the length of the smallest repeating substring. It checks if the entire string can be constructed by repeating this substring.

Question 5:

Tags: python, numpy, keras

Question: I'm trying to implement the binary classification example using the IMDb dataset in Google Colab. I have implemented this model before. But when I tried to do it again after a few days, it returned a `value error: 'Object arrays cannot be loaded when allow_pickle=False'` for the

load_data() function. I have already tried solving this, referring to an existing answer for a similar problem: How to fix 'Object arrays cannot be loaded when allow_pickle=False' in the sketch_rnn algorithm. But it turns out that just adding an allow_pickle argument isn't sufficient. My code:

```
from keras.datasets import imdb (train_data, train_labels), (test_data,
test_labels) = imdb.load_data(num_words=10000) The error: ValueError Traceback
(most recent call last) <ipython-input-1-2ab3902db485> in <module>() 1 from
keras.datasets import imdb ----> 2 (train_data, train_labels), (test_data,
test_labels) = imdb.load_data(num_words=10000) 2 frames
/usr/local/lib/python3.6/dist-packages/keras/datasets/imdb.py in
load_data(path, num_words, skip_top, maxlen, seed, start_char, oov_char,
index_from, **kwargs) 57 file_hash='599dadb1135973df5b59232a0e9a887c') 58
with np.load(path) as f: ---> 59 x_train, labels_train = f['x_train'],
f['y_train'] 60 x_test, labels_test = f['x_test'], f['y_test'] 61
/usr/local/lib/python3.6/dist-packages/numpy/lib/npio.py in
__getitem__(self, key) 260 return format.read_array(bytes, 261
allow_pickle=self.allow_pickle, --> 262 pickle_kwargs=self.pickle_kwargs)
263 else: 264 return self.zip.read(key)
/usr/local/lib/python3.6/dist-packages/numpy/lib/format.py in read_array(fp,
allow_pickle, pickle_kwargs) 690 # The array contained Python objects. We
need to unpickle the data. 691 if not allow_pickle: --> 692 raise
ValueError("Object arrays cannot be loaded when " 693 "allow_pickle=False")
694 if pickle_kwargs is None: ValueError: Object arrays cannot be loaded
when allow_pickle=False
```

Answer Score: 159

Stack Overflow answer, Verb Count: 11 , **Adjective Count:** 9

Stack Overflow Answer: Here's a trick to force `imdb.load_data` to allow pickle by, in your notebook, replacing this line: `(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words=10000)` by this: `import numpy as np # save np.load`
`np_load_old = np.load # modify the default parameters of np.load np.load =`
`lambda *a,**k: np_load_old(*a, allow_pickle=True, **k) # call load_data with`
`allow_pickle implicitly set to true (train_data, train_labels), (test_data,`
`test_labels) = imdb.load_data(num_words=10000) # restore np.load for future`
`normal usage np.load = np_load_old`

GPT answer, Verb Count: 30 , **Adjective Count:** 25

GPT-4o Answer: Fixing "Object arrays cannot be loaded when allow_pickle=False" in IMDB Dataset
This error occurs because the `numpy.load` function used internally by `keras.datasets.imdb.load_data` requires the `allow_pickle` parameter to be set to `True`. By default, it is set to `False` for security reasons. Solution: Modify the Keras source code to allow pickling or use an alternative loading method. The easiest way is to set the `allow_pickle` parameter explicitly: `import numpy as np from keras.datasets import imdb # Manually load`
`the dataset path = keras.utils.get_file('imdb.npz',`
`origin='https://s3.amazonaws.com/text-datasets/imdb.npz',`
`file_hash='599dadb1135973df5b59232a0e9a887c') # Load using numpy with`
`allow_pickle=True with np.load(path, allow_pickle=True) as f: x_train,`
`y_train = f['x_train'], f['y_train'] x_test, y_test = f['x_test'],`
`f['y_test'] print("Data loaded successfully") Alternative Solution: If modifying the`
`source code is not an option, update your TensorFlow/Keras version, as the issue may have been fixed`
`in newer releases. Use: !pip install --upgrade tensorflow After upgrading, retry your`
`original code: from keras.datasets import imdb (train_data, train_labels),`
`(test_data, test_labels) = imdb.load_data(num_words=10000) Explanation: 1. The`

`allow_pickle` flag controls whether NumPy allows loading object arrays from .npz files. 2. Manually setting `allow_pickle=True` resolves the issue when loading the dataset. 3. Keeping libraries up-to-date ensures compatibility with newer features and bug fixes.

Co-Pilot, Verb Count: 18 , Adjective Count: 24

Co-Pilot Answer: The error occurs because the default settings for `np.load` do not allow pickled object arrays to be loaded. You need to explicitly set `allow_pickle=True` when calling `np.load` inside the `imdb.py` module from Keras. Since you cannot modify the Keras source files directly in Google Colab, you can download the IMDB data manually and load it. Here is how you can do it:

```
import numpy as np
import os
import keras
from keras.utils.data_utils import get_file

def load_imdb_data(path='imdb.npz', num_words=None, skip_top=0, maxlen=None, seed=113, start_char=1, oov_char=2, index_from=3, **kwargs):
    path = get_file(path,
        origin='https://s3.amazonaws.com/text-datasets/imdb.npz',
        file_hash='599dadb1135973df5b59232a0e9a887c')
    with np.load(path, allow_pickle=True) as f:
        x_train, labels_train = f['x_train'], f['y_train']
        x_test, labels_test = f['x_test'], f['y_test']
    np.random.seed(seed)
    np.random.shuffle(x_train)
    np.random.seed(seed)
    np.random.shuffle(labels_train)
    np.random.seed(seed * 2)
    np.random.shuffle(x_test)
    np.random.seed(seed * 2)
    np.random.shuffle(labels_test)
    xs = np.concatenate([x_train, x_test])
    labels = np.concatenate([labels_train, labels_test])
    if start_char is not None:
        xs = [[start_char] + [w + index_from for w in x] for x in xs]
    elif index_from:
        xs = [[w + index_from for w in x] for x in xs]
    if maxlen:
        xs, labels = keras.preprocessing.sequence.pad_sequences(xs, maxlen=maxlen, padding='post', truncating='post')
    labels_idxs = np.arange(len(xs))
    np.random.seed(seed)
    np.random.shuffle(labels_idxs)
    xs = np.array(xs)[labels_idxs]
    labels = np.array(labels)[labels_idxs]
    if not num_words:
        return (x_train, y_train), (x_test, y_test)
    if start_char is not None and oov_char is not None:
        num_words += 2
    elif oov_char is not None:
        num_words += 1
    elif start_char is not None:
        num_words += 1
    xs = [[w if (skip_top <= w < num_words) else oov_char for w in x] for x in xs]
    return (np.array(xs[:len(x_train)]), labels_train), (np.array(xs[len(x_train):]), labels_test)

(train_data, train_labels), (test_data, test_labels) = load_imdb_data(num_words=10000)
```

Month: May-19

Question 1:

Tags: javascript, typescript, reactjs, react-hooks

Question: Are there any benefits in using `useMemo` (e.g. for an intensive function call) instead of using a combination of `useEffect` and `useState`? Here are two custom hooks that work exactly the same on first sight, besides `useMemo`'s return value being `null` on the first render: `useEffect` & `useState`

```
import { expensiveCalculation } from "foo";
function useCalculate(someNumber: number): number | null {
    const [result, setResult] = useState<number | null>(null);
    useEffect(() => {
        setResult(expensiveCalculation(someNumber));
    }, [someNumber]);
    return result;
}
import { expensiveCalculation } from "foo";
function useCalculateWithMemo(someNumber: number): number {
    return useMemo(() => {
        return expensiveCalculation(someNumber);
    }, [someNumber]);
}
Both calculate the result each time their parameter someNumber changes, where is the memoization of useMemo kicking
```

in?

Answer Score: 227

Stack Overflow answer, Verb Count: 24 , **Adjective Count:** 39

Stack Overflow Answer: The `useEffect` and `setState` will cause extra renders on every change: the first render will "lag behind" with stale data and then it'll immediately queue up an additional render with the new data. Suppose we have: `// Maybe I'm running this on a literal potato function expensiveCalculation(x) { return x + 1; };` Lets suppose `x` is initially 0: The `useMemo` version immediately renders 1. The `useEffect` version renders `null`, then after the component renders the effect runs, changes the state, and queues up a new render with 1. Then if we change `x` to 2: The `useMemo` runs and 3 is rendered. The `useEffect` version runs, and renders 1 again, then the effect triggers and the component reruns with the correct value of 3. In terms of how often `expensiveCalculation` runs, the two have identical behavior, but the `useEffect` version is causing twice as much rendering which is bad for performance for other reasons. Plus, the `useMemo` version is just cleaner and more readable, IMO. It doesn't introduce unnecessary mutable state and has fewer moving parts. So you're better off just using `useMemo` here.

GPT answer, Verb Count: 45 , **Adjective Count:** 43

GPT-4o Answer: Benefits of `useMemo` vs. `useEffect` and `useState` Key Differences: 1. `useMemo`: Calculates the value during rendering and memoizes the result to avoid recomputation on future renders unless dependencies change. 2. `useEffect` with `useState`: Performs the calculation after rendering and updates the state, triggering a second render when the state changes. Advantages of `useMemo`: 1. Reduces Re-Renders: Avoids the second render caused by `useState` updates, which can be beneficial for performance in components with complex hierarchies. 2. Simpler Code: Directly returns the calculated value, eliminating the need for managing intermediate state and effects. 3. Immediate Availability: The computed value is available during the initial render, while `useEffect` runs after rendering, leading to a null state initially. When to Use `useMemo`: 1. Use `useMemo` for synchronously computed values that are expensive and need to be available during the render phase. 2. Use `useMemo` when the computed value does not rely on asynchronous operations or side effects. When to Use `useEffect` and `useState`: 1. Use `useEffect` and `useState` when the computation involves side effects or asynchronous operations, such as fetching data or updating DOM elements. 2. Use `useEffect` when the value does not need to be immediately available during the render phase. Memoization in `useMemo`: Memoization prevents the re-execution of the `expensiveCalculation` function if `someNumber` remains unchanged between renders. This optimization is critical for preventing unnecessary computations in performance-critical applications. Conclusion: Choose `useMemo` when you need to calculate and memoize a value during the render phase without triggering additional renders. Use `useEffect` with `useState` for computations involving side effects or when the calculated value does not need to be immediately available.

Co-Pilot, Verb Count: 28 , **Adjective Count:** 25

Co-Pilot Answer: `useMemo` and a combination of `useEffect` and `useState` can both be used to optimize performance for expensive computations, but they serve different purposes and have different benefits. Benefits of using `useMemo`: `useMemo` is specifically designed for memoizing the result of a computation. It caches the computed value and recalculates it only when its dependencies change. This can improve performance by preventing unnecessary recalculations. `import { expensiveCalculation } from "foo"; function useCalculateWithMemo(someNumber: number): number { return useMemo(() => { return expensiveCalculation(someNumber); }, [someNumber]); };` Benefits of using `useEffect` and `useState`: The combination of `useEffect` and `useState` allows you to handle side

effects and state updates separately. This approach can be useful when the computation has side effects or when you need to perform additional actions based on the computed value. `import { expensiveCalculation } from "foo"; function useCalculate(someNumber: number): number | null { const [result, setResult] = useState(null); useEffect(() => { setResult(expensiveCalculation(someNumber)); }, [someNumber]); return result; }` Memoization Benefits: The memoization benefit of `useMemo` kicks in by ensuring that the function `expensiveCalculation` is only recalculated when its dependency `someNumber` changes. This prevents unnecessary recalculations, providing a performance boost for expensive computations.

Question 2:

Tags: java, pom.xml, maven, spring-boot

Question: Getting unknown error at Line 1 in `pom.xml` in Eclipse IDE. It was working fine till yesterday, but all of a sudden after updating my project from master and after fixing merge conflicts getting "Unknown error" in `pom.xml`. Except me, none of my teammates are facing this issue. I also changed the workspace, deleted cache but still no luck. I am using the h2 database for this project, in spite of inserting values in `data.sql`, it is not picking up any values and inserting into h2 DB tables. Before this issue, it was working fine. I feel its due to the issue in `pom.xml`, I am not sure though. Please help I deleted the existing project and took the latest code from master. After that deleted `.m2` folder under users including the repository folder. I did update project and enabled Forced update of snapshots/releases, maven clean and maven build. But nothing helps. `pom.xml` `<?xml`

```
version="1.0" encoding="UTF-8"?> <project
xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion> <groupId>com.abc.roster</groupId>
<artifactId>spring-boot-roster-app</artifactId>
<version>0.0.1-SNAPSHOT</version> <packaging>jar</packaging>
<name>spring-boot-roster-app</name> <description>Demo project for Spring
Boot Roster</description> <parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.2.0.BUILD-SNAPSHOT</version> <relativePath /> <!-- lookup parent
from repository --> </parent> <properties>
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
<java.version>1.8</java.version> <!--
<start-class>com.infosys.roster.SpringBootRosterAppApplication</start-class>
--> </properties> <dependencies> <dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter</artifactId> </dependency> <dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-security</artifactId> </dependency>
<!--encryption lib --> <dependency> <groupId>org.jasypt</groupId>
<artifactId>jasypt</artifactId> <version>1.9.2</version> </dependency>
<dependency> <groupId>org.jasypt</groupId>
<artifactId>jasypt-springsecurity3</artifactId> <version>1.9.0</version>
</dependency> <dependency> <groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-test</artifactId> <scope>test</scope>
</dependency> <dependency> <groupId>org.springframework</groupId>
```

```

<artifactId>spring-web</artifactId> </dependency> <dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId> </dependency> <dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-jpa</artifactId> </dependency>
<dependency> <groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-data-rest</artifactId> </dependency>
<dependency> <groupId>com.h2database</groupId> <artifactId>h2</artifactId>
<scope>runtime</scope> </dependency> <!-- <dependency> --> <!--
<groupId>org.hibernate</groupId> --> <!--
<artifactId>hibernate-core</artifactId> --> <!--
<version>5.2.10.Final</version> --> <!-- </dependency> --> <!-- <dependency>
--> <!-- <groupId>io.springfox</groupId> --> <!--
<artifactId>springfox-swagger2</artifactId> --> <!--
<version>2.7.0</version> --> <!-- <scope>compile</scope> --> <!--
</dependency> --> <!-- <dependency> --> <!-- <groupId>io.springfox</groupId>
--> <!-- <artifactId>springfox-swagger-ui</artifactId> --> <!--
<version>2.7.0</version> --> <!-- <scope>compile</scope> --> <!--
</dependency> --> <!-- <dependency> --> <!-- <groupId>io.springfox</groupId>
--> <!-- <artifactId>springfox-data-rest</artifactId> --> <!--
<version>2.7.0</version> --> <!-- </dependency> --> </dependencies> <build>
<plugins> <plugin> <groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId> </plugin> </plugins>
</build> <repositories> <repository> <id>spring-snapshots</id> <name>Spring
Snapshots</name> <url>https://repo.spring.io/snapshot</url> <snapshots>
<enabled>true</enabled> </snapshots> </repository> <repository>
<id>spring-milestones</id> <name>Spring Milestones</name>
<url>https://repo.spring.io/milestone</url> </repository> </repositories>
<pluginRepositories> <pluginRepository> <id>spring-snapshots</id>
<name>Spring Snapshots</name> <url>https://repo.spring.io/snapshot</url>
<snapshots> <enabled>true</enabled> </snapshots> </pluginRepository>
<pluginRepository> <id>spring-milestones</id> <name>Spring Milestones</name>
<url>https://repo.spring.io/milestone</url> </pluginRepository>
</pluginRepositories> </project> application.properties server.port=
spring.h2.console.path=/h2 spring.h2.console.enabled=true
spring.datasource.url= spring.datasource.username=
spring.datasource.password= spring.datasource.driverClassName=org.h2.Driver
spring.datasource.platform=h2 spring.datasource.initialization-mode=always
spring.jpa.hibernate.ddl-auto=update
spring.datasource.continue-on-error=true spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=false
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.H2Dialect
Should remove unknown error in pom.xml and also I must be able to populate values in H2 db.

```

Answer Score: 186

Stack Overflow answer, Verb Count: 9 , **Adjective Count:** 9

Stack Overflow Answer: answer according to current status This issue got fixed: Please install the m2e connector for mavenarchiver plugin 0.17.3 from <https://download.eclipse.org/m2e-wtp/releases/1.4/> obsolete answer A less profound change than a downgrade from Spring Boot 2.1.5.RELEASE to 2.1.4.RELEASE would be downgrading only the affected Maven JAR Plugin from 3.1.2 to 3.1.1 as long as this bug exists: <properties> <!-- ... --> <maven-jar-plugin.version>3.1.1</maven-jar-plugin.version>

</properties>

GPT answer, Verb Count: 42 , Adjective Count: 29

GPT-4o Answer: Steps to Fix Unknown Error in pom.xml and H2 Database Issue

1. Resolve Unknown Error in pom.xml: The issue with the "Unknown error" in `pom.xml` is likely due to an issue with your local Maven setup or IDE configuration. Follow these steps:

1. Delete the local Maven repository: Delete the `.m2/repository` directory to remove corrupted dependencies.
2. Force Maven to update dependencies: Run the following command: `mvn clean install -U`
3. Re-import the project into Eclipse: - Right-click on your project in Eclipse. - Select Maven → Update Project.... - Check the option for Force Update of Snapshots/Releases.
4. Validate the pom.xml structure: Ensure the XML is correctly formatted and contains valid Maven dependencies and plugins.
5. Verify Internet connectivity: Ensure your machine can access Maven repositories online.

2. Fix H2 Database Issue: The H2 database not populating values from `data.sql` could be caused by the `spring.datasource.initialization-mode` setting. To fix it, do the following:

1. Ensure your `data.sql` file is in the `src/main/resources` directory.
2. Update the `application.properties` with the correct configuration:
`spring.datasource.initialization-mode=always` `spring.sql.init.mode=always`
`spring.datasource.url=jdbc:h2:mem:testdb` `spring.datasource.username=sa`
`spring.datasource.password=`
3. Verify the H2 Console: Access the H2 console at `http://localhost:[PORT]/h2` and use the correct URL, username, and password.

3. General Checks:

1. Ensure all dependencies in the `pom.xml` are resolved correctly.
2. Use compatible versions of Spring Boot and its dependencies. For instance: - Check if 2.2.0.BUILD-SNAPSHOT is valid and supported.
3. If issues persist, update to the latest Spring Boot stable version and re-test.

By following these steps, you should resolve the "Unknown error" in `pom.xml` and ensure that the H2 database initializes and populates data as expected.

Co-Pilot, Verb Count: 20 , Adjective Count: 17

Co-Pilot Answer: The "Unknown error" in your `pom.xml` could be due to several reasons, including XML parsing issues or dependency resolution problems. Here are some steps to troubleshoot and resolve the issue:

1. Check for XML Parsing Errors:

```
<?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"> <modelVersion>4.0.0</modelVersion> <!-- Rest of your POM file --> </project>
```

Ensure your `pom.xml` file starts with a valid XML declaration and project tag.
2. Correct Parent Dependency Version:

```
<parent> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-parent</artifactId> <version>2.2.0.RELEASE</version> <relativePath /> <!-- lookup parent from repository --> </parent>
```

Use a stable release version instead of a snapshot for your parent dependency.
3. Update Maven Repositories:

```
<repositories> <repository> <id>central</id> <name>Maven Central</name> <url>https://repo.maven.apache.org/maven2</url> </repository> </repositories>
```

Ensure your repositories section includes the Maven Central repository.
4. Reimport Maven Project: Right-click on the project in Eclipse > Maven > Update Project > Select Force Update of Snapshots/Releases > OK
5. Delete and Rebuild .m2 Repository: Close Eclipse > Delete the `.m2/repository` folder > Restart Eclipse > Rebuild the project
6. Ensure H2 Database Initialization: `spring.datasource.initialization-mode=always` `spring.datasource.platform=h2` Verify these settings in your `application.properties`. Following these steps should help resolve

the unknown error in your [pom.xml](#) and ensure values are populated in the H2 database.

Question 3:

Tags: java, android

Question: I am working on an Android Studio project with several activities. I am currently trying to read the output from a Java Servlet on localhost but it seems to be crashing due to a socket permission. I've made a new project, used the exact same code and worked perfectly. So I don't understand why it is not willing to work on my project. `public class LoginActivity extends`

```
AppCompatActivity { String apiUrl =
"http://10.0.2.2:8080/ProjectService/Servlet?action=login"; EditText
username; EditText password; AlertDialog dialog; Usuario session; @Override
public void onCreate(Bundle savedInstanceState) { // Inicializacion de
ventana super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login); getSupportActionBar().hide(); //
Inicializacion de componentes username = findViewById(R.id.username);
password = findViewById(R.id.password); // Inicializacion de funcionalidad
de botones Button button= (Button) findViewById(R.id.login);
button.setOnClickListener(new View.OnClickListener() { public void
onClick(View v) { UserLoginTask mAuthTask = new UserLoginTask();
mAuthTask.execute(); } }); password = findViewById(R.id.password);
createAlertDialog("Usuario o Contraseña Incorrectos"); } private void
createAlertDialog(String message){ AlertDialog.Builder builder = new
AlertDialog.Builder(this); builder.setMessage(message) .setTitle("Error");
dialog = builder.create(); } // ASYNCRONOUS NETWORK PROCESS public class
UserLoginTask extends AsyncTask<String, String, String> { @Override
protected void onPreExecute() { } @Override protected String
doInBackground(String... params) { // implement API in background and store
the response in current variable String current = ""; try { URL url;
HttpURLConnection urlConnection = null; try { url = new URL(apiUrl);
System.out.println(apiUrl); urlConnection = (HttpURLConnection) url
.openConnection(); InputStream in = urlConnection.getInputStream();
InputStreamReader isw = new InputStreamReader(in); int data = isw.read();
while (data != -1) { current += (char) data; data = isw.read();
//System.out.print(current); } System.out.print(current); // return the data
to onPostExecute method return current; } catch (Exception e) {
e.printStackTrace(); } finally { if (urlConnection != null) {
urlConnection.disconnect(); } } } catch (Exception e) { e.printStackTrace();
return "Exception: " + e.getMessage(); } return current; } } protected void
onPostExecute(String success) { Log.i(success, ""); //attemptLogin(); } } }
```

Expect it to read the data but it crashes at this line: `InputStream in =`

```
urlConnection.getInputStream(); This is the error output: java.net.SocketException:
socket failed: EPERM (Operation not permitted) at
java.net.Socket.createImpl(Socket.java:492) at
java.net.Socket.getImpl(Socket.java:552) at
java.net.Socket.setSoTimeout(Socket.java:1180) at com.android.okhttp.internal
io.RealConnection.connectSocket(RealConnection.java:143) at com.android.ok
http.internal.io.RealConnection.connect(RealConnection.java:116) at com.andr
oid.okhttp.internal.http.StreamAllocation.findConnection(StreamAllocation.ja
va:186) at com.android.okhttp.internal.http.StreamAllocation.findHealthyConn
ection(StreamAllocation.java:128) at com.android.okhttp.internal.http.Stream
Allocation.newStream(StreamAllocation.java:97) at
```

```
com.android.okhttp.internal.http.HttpEngine.connect(HttpEngine.java:289) at
com.android.okhttp.internal.http.HttpEngine.sendRequest(HttpEngine.java:232)
at com.android.okhttp.internal.huc.HttpURLConnectionImpl.execute(HttpURLConnectionImpl.java:465) at com.android.okhttp.internal.huc.HttpURLConnectionImpl.getResponse(HttpURLConnectionImpl.java:411) at com.android.okhttp.internal.huc.HttpURLConnectionImpl.getInputStream(HttpURLConnectionImpl.java:248) at com.example.controller.LoginActivity$UserLoginTask.doInBackground(LoginActivity.java:114) at com.example.controller.LoginActivity$UserLoginTask.doInBackground(LoginActivity.java:93) at android.os.AsyncTask$3.call(AsyncTask.java:378) at java.util.concurrent.FutureTask.run(FutureTask.java:266) at android.os.AsyncTask$SerialExecutor$1.run(AsyncTask.java:289) at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1167) at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:641) at java.lang.Thread.run(Thread.java:919)
```

Answer Score: 817

Stack Overflow answer, Verb Count: 10 , **Adjective Count:** 8

Stack Overflow Answer: Your app needs additional permissions and/or to be reinstalled. Add additional permissions to `AndroidManifest.xml` within the `<manifest>` section:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

To reinstall, uninstall the app from the emulator or physical connected device and then run it again. (If adding permissions, make sure to reinstall afterwards.)

GPT answer, Verb Count: 30 , **Adjective Count:** 18

GPT-4o Answer: Resolving "java.net.SocketException: socket failed: EPERM (Operation not permitted)" The error occurs because of a missing permission or incorrect configuration in your Android project. Follow the steps below to resolve the issue: 1. Add Internet Permission to Your Manifest File:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.controller"> <uses-permission
android:name="android.permission.INTERNET" /> <application ... > ...
</application> </manifest>
```

2. Use the Correct URL for Emulator: The URL `http://10.0.2.2` is correct when accessing localhost on an emulator. Ensure your servlet is running on your machine and is accessible through `http://10.0.2.2:8080`.

3. Verify Network Configuration: In Android 9 (API 28) and above, cleartext traffic (HTTP) is disabled by default. To allow HTTP requests, update your `network_security_config.xml`:

```
<network-security-config> <domain-config
cleartextTrafficPermitted="true"> <domain
includeSubdomains="true">10.0.2.2</domain> </domain-config>
</network-security-config>
```

Add this to your `AndroidManifest.xml` under the `<application>` tag:

```
android:networkSecurityConfig="@xml/network_security_config"
```

4. Check Your Localhost Server: Ensure your localhost server (e.g., Tomcat) is running and accessible on `http://10.0.2.2:8080`.

5. Debugging Tips: - Log the exception message and stack trace to identify exact issues. - Use tools like Postman or a browser to verify if the servlet endpoint is working as expected. After making these changes, your application should be able to access the servlet without encountering the socket permission error.

Co-Pilot, Verb Count: 17 , **Adjective Count:** 13

Co-Pilot Answer: The error you're encountering, `java.net.SocketException: socket`

failed: EPERM (Operation not permitted), suggests that your application is being denied network access. This typically happens due to missing permissions in your AndroidManifest.xml file. To resolve this issue, ensure you have the necessary permissions added to your AndroidManifest.xml file:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Here's an example of how your AndroidManifest.xml file should look:

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.yourapp"> <uses-permission
android:name="android.permission.INTERNET" /> <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" /> <application
android:allowBackup="true" android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round" android:supportsRtl="true"
android:theme="@style/AppTheme"> <activity android:name=".LoginActivity">
<intent-filter> <action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter> </activity> </application> </manifest>
```

 Make sure you have the INTERNET and ACCESS_NETWORK_STATE permissions declared. This should allow your application to access the network and avoid the EPERM error.

Question 4:

Tags: javascript, reactjs, react-hooks

Question: I've initialized a state that is an array, and when I update it my component does not re-render. Here is a minimal proof-of-concept:

```
function App() { const [numbers,
setNumbers] = React.useState([0, 1, 2, 3]); console.log("rendering...");
return ( <div className="App"> {numbers.map(number => ( <p>{number}</p> ))}
<input type="text" value={numbers[0].toString()} onChange={newText => { let
old = numbers; old[0] = 1; setNumbers(old); }} /> </div> ); }
```

 Based on this code, it seems that the input should contain the number 0 to start, and any time it is changed, the state should change too. After entering "02" in the input, the App component does not re-render. However, if I add a setTimeout in the onChange function which executes after 5 seconds, it shows that numbers has indeed been updated. Any thoughts on why the component doesn't update? Here is a CodeSandbox with the proof of concept.

Answer Score: 561

Stack Overflow answer, Verb Count: 12 , **Adjective Count:** 8

Stack Overflow Answer: You're calling `setNumbers` and passing it the array it already has. You've changed one of its values but it's still the same array, and I suspect React doesn't see any reason to re-render because state hasn't changed; the new array is the old array. One easy way to avoid this is by spreading the array into a new array: `setNumbers([...old])`

GPT answer, Verb Count: 39 , **Adjective Count:** 12

GPT-4o Answer: Why the Component Does Not Re-Render The issue occurs because React detects state updates using shallow comparison. In your code, you are mutating the original array instead of creating a new one. React doesn't detect the change because the reference to the state object hasn't changed. Explanation: 1. When you execute `let old = numbers;`, you are creating a reference to the same array object in memory. 2. Modifying `old[0]` directly mutates the existing array without changing its reference. 3. React's state comparison mechanism does not detect changes to the contents of the array, only to the reference itself. Solution: Create a new array when updating the state to ensure React detects the change and re-renders the component.

```
function App() { const
```

```
[numbers, setNumbers] = React.useState([0, 1, 2, 3]);
console.log("rendering..."); return ( <div className="App">
{numbers.map(number => ( <p>{number}</p> ))} <input type="text"
value={numbers[0].toString()} onChange={(e) => { const newNumbers =
[...numbers]; // Create a new array newNumbers[0] = parseInt(e.target.value,
10) || 0; // Update the value setNumbers(newNumbers); // Update the state }}
/> </div> ); } Key Points: 1. Always use immutable operations when updating
state in React. 2. Use [...numbers] to create a new array and update it. 3.
This ensures that React can detect the change in state and trigger a
re-render.
```

Co-Pilot, Verb Count: 20 , Adjective Count: 12

Co-Pilot Answer: The issue is that you are mutating the existing array, which React does not recognize as a state change. React relies on referential equality to determine if a state update has occurred. When you mutate the array, the reference remains the same, so React does not trigger a re-render. To fix this, you need to create a new array with the updated value: `function App() { const [numbers, setNumbers] = React.useState([0, 1, 2, 3]); console.log("rendering..."); return ({numbers.map(number => ({number}))} type="text" value={numbers[0].toString()} onChange={newText => { const newNumbers = [...numbers]; newNumbers[0] = 1; setNumbers(newNumbers); }} />); }` In this code, `const newNumbers = [...numbers];` creates a new array with the same elements as the original array. Updating `newNumbers[0]` does not mutate the original array. By passing `newNumbers` to `setNumbers`, you trigger a re-render.

Question 5:

Tags: python, math, floating-point, hash, pi

Question: The hash of infinity in Python has digits matching pi: `>>> inf = float('inf') >>> hash(inf) 314159 >>> int(math.pi*1e5) 314159` Is that just a coincidence or is it intentional?

Answer Score: 50

Stack Overflow answer, Verb Count: 8 , Adjective Count: 6

Stack Overflow Answer: `_PyHASH_INF` is defined as a constant equal to `314159`. I can't find any discussion about this, or comments giving a reason. I think it was chosen more or less arbitrarily. I imagine that as long as they don't use the same meaningful value for other hashes, it shouldn't matter.

GPT answer, Verb Count: 13 , Adjective Count: 25

GPT-4o Answer: Explanation of the Hash of Infinity The hash value of infinity in Python is not a coincidence; it is intentional. Python's `hash` function for floating-point numbers is designed to provide consistent and deterministic results for special floating-point values like `float('inf')`, `float('-inf')`, and `float('nan')`. The value `314159` is derived from an intentional choice in the Python implementation. Specifically, the hash of positive infinity is defined as: `hash(float('inf')) = 314159` This value corresponds to `int(math.pi * 1e5)`, making it a deliberate nod to the mathematical constant pi. The reason for this choice is largely arbitrary but serves as a meaningful and recognizable constant for the hash of infinity. Similarly, the hash of negative infinity and NaN are also predefined: `hash(float('-inf')) = -314159` `hash(float('nan')) = 0` Conclusion: The matching digits between `hash(float('inf'))` and `int(math.pi * 1e5)` are intentional, chosen as a recognizable constant in the Python implementation.

Co-Pilot, Verb Count: 2 , Adjective Count: 4

Co-Pilot Answer: It's an intentional design choice by Python's developers. The hash value of positive

infinity is set to 314159, which is the first six digits of pi. This design was inspired by pi's significance in mathematics and as a playful nod by the developers.

Month: Jun-19

Question 1:

Tags: javascript, unit-testing, reactjs, jestjs, react-testing-library

Question: Here is my code for a tooltip that toggles the CSS property `display: block` on `MouseOver` and on `Mouse Out display: none`. `it('should show and hide the message using onMouseOver and onMouseOut events respectively', () => { const { queryById, queryByText } = render(<Tooltip id="test" message="test" />,) fireEvent.mouseOver(queryById('tooltip')) expect(queryByText('test')).toBeInTheDocument() fireEvent.mouseOut(queryById('tooltip')) expect(queryByText('test')).not.toBeInTheDocument() cleanup() })` I keep getting the error `TypeError: expect(...).toBeInTheDocument is not a function` Has anyone got any ideas why this is happening? My other tests to render and snapshot the component all work as expected. As do the `queryById` and `queryById`.

Answer Score: 464

Stack Overflow answer, Verb Count: 4 , Adjective Count: 2

Stack Overflow Answer: `toBeInTheDocument` is not part of RTL. You need to install `jest-dom` to enable it. And then import it in your test files by: `import '@testing-library/jest-dom'`

GPT answer, Verb Count: 20 , Adjective Count: 15

GPT-4o Answer: Reason for the Error and Solution Reason: The error `TypeError: expect(...).toBeInTheDocument is not a function` occurs because the `toBeInTheDocument` matcher is not included by default in Jest. It is provided by the `@testing-library/jest-dom` package, which needs to be explicitly imported or set up in your test environment. Solution: 1. Install the `@testing-library/jest-dom` package: `npm install --save-dev @testing-library/jest-dom` 2. Import `@testing-library/jest-dom` into your test file: `import '@testing-library/jest-dom'`; Alternatively, you can configure Jest to load `@testing-library/jest-dom` globally for all tests by adding it to your Jest setup file (e.g., `setupTests.js`): `// setupTests.js import '@testing-library/jest-dom'`; Then, ensure your Jest configuration includes the setup file: `"jest": { "setupFilesAfterEnv": ["<rootDir>/setupTests.js"] }` 3. After importing `@testing-library/jest-dom`, the `toBeInTheDocument` matcher will work as expected in your test: `it('should show and hide the message using onMouseOver and onMouseOut events respectively', () => { const { queryById, queryByText } = render(<Tooltip id="test" message="test" />,); fireEvent.mouseOver(queryById('tooltip')); expect(queryByText('test')).toBeInTheDocument(); // No error fireEvent.mouseOut(queryById('tooltip')); expect(queryByText('test')).not.toBeInTheDocument(); // No error cleanup(); });`

Co-Pilot, Verb Count: 8 , Adjective Count: 8

Co-Pilot Answer: The error `TypeError: expect(...).toBeInTheDocument is not a function` typically occurs when the `toBeInTheDocument` matcher is not properly imported. The matcher `toBeInTheDocument` is part of the `@testing-library/jest-dom` library. You need to import this library in your test file to use the `toBeInTheDocument` matcher. Add the following import

statement at the top of your test file: `import '@testing-library/jest-dom'`; With this import, the matchers from `@testing-library/jest-dom` will be available in your test file, and the error should be resolved.

Question 2:

Tags: python, matplotlib, pycharm

Question: I am trying to plot a simple graph using pyplot, e.g.: `import matplotlib.pyplot as plt plt.plot([1,2,3],[5,7,4]) plt.show()` but the figure does not appear and I get the following message: `UserWarning: Matplotlib is currently using agg, which is a non-GUI backend, so cannot show the figure.` I found and tried some advice to re-configure the "backend" mentioned in that warning, like so: `import matplotlib matplotlib.use('TkAgg') import matplotlib.pyplot as plt` but this gives me an error message: `ModuleNotFoundError: No module named 'tkinter'` I assumed that I had to install this module separately, but `pip install tkinter` does not work: `Collecting tkinter Could not find a version that satisfies the requirement tkinter (from versions:) No matching distribution found for tkinter` How can I make Matplotlib display the graph? See also: Why does tkinter (or turtle) seem to be missing or broken? Shouldn't it be part of the standard library? . This question is not a duplicate, because the answers discuss other backends besides the Tkinter one. Also see `_tkinter.TclError: no display name and no $DISPLAY environment variable` for issues with attempts to use Matplotlib remotely.

Answer Score: 605

Stack Overflow answer, Verb Count: 24 , **Adjective Count:** 15

Stack Overflow Answer: Solution 1: is to install the GUI backend `tk` I found a solution to my problem (thanks to the help of ImportanceOfBeingErnest). All I had to do was to install `tkinter` through the Linux bash terminal using the following command: `sudo apt-get install python3-tk` instead of installing it with `pip` or directly in the virtual environment in Pycharm. Solution 2: install any of the `matplotlib` supported GUI backends solution 1 works fine because you get a GUI backend... in this case the `TkAgg` however you can also fix the issue by installing any of the matplotlib GUI backends like `Qt5Agg`, `GTKAgg`, `Qt4Agg`, etc for example `pip install pyqt5` will fix the issue also NOTE: usually this error appears when you pip install matplotlib and you are trying to display a plot in a GUI window and you do not have a python module for GUI display. The authors of `matplotlib` made the pypi software deps not depend on any GUI backend because some people need `matplotlib` without any GUI backend.

GPT answer, Verb Count: 43 , **Adjective Count:** 19

GPT-4o Answer: Solution to Display Matplotlib Graphs Reason: The error occurs because your Python environment does not have a GUI backend available for Matplotlib to render and display the plot. The default "agg" backend does not support GUI windows, and the `tkinter` module required for the "TkAgg" backend is not installed in your system. Steps to Fix: 1. Install the Required GUI Backend: If you are using a Linux-based system, install `tkinter` using your system package manager: # On Ubuntu/Debian `sudo apt-get install python3-tk` # On Fedora `sudo dnf install python3-tkinter` If you are using Windows, ensure you installed Python with the option to include `tkinter` selected during installation. Reinstall Python if necessary, enabling this option. 2. Specify a Supported Backend: If you want to use a different backend instead of "TkAgg," specify a backend that works for your system: `import matplotlib matplotlib.use('Qt5Agg')` # Requires PyQt5 or PySide2 `import matplotlib.pyplot as plt` You may need to install PyQt5 or PySide2: `pip install PyQt5` 3. Run in an Interactive Environment: If you are using Jupyter Notebook, enable inline plotting: `%matplotlib inline` For JupyterLab, you may need: `%matplotlib widget` 4. Plot and Display: Once the GUI backend is installed and configured, the following code should display the graph: `import matplotlib.pyplot as plt plt.plot([1, 2, 3], [5,`


```
7, 4]) plt.show()
```

Additional Notes: If you are working on a headless server or remote environment (e.g., without a display), use the "Agg" backend and save the plot to a file: `import matplotlib matplotlib.use('Agg') import matplotlib.pyplot as plt plt.plot([1, 2, 3], [5, 7, 4]) plt.savefig('output.png')`

Co-Pilot, Verb Count: 23 , Adjective Count: 12

Co-Pilot Answer: To display the graph using Matplotlib, you need to install the `tkinter` module, which is the GUI toolkit used by Matplotlib's `TkAgg` backend. Since `pip install tkinter` does not work, it suggests that `tkinter` is not available via pip and must be installed through your package manager. On Windows: `python -m pip install python-tk` On Ubuntu or Debian-based systems: `sudo apt-get install python3-tk` On macOS (using Homebrew): `brew install python-tk` After installing `tkinter`, you can set the backend and plot your graph: `import matplotlib matplotlib.use('TkAgg') import matplotlib.pyplot as plt plt.plot([1,2,3],[5,7,4]) plt.show()` If you still encounter issues, you can try using another GUI backend like `Qt5Agg`: `import matplotlib matplotlib.use('Qt5Agg') import matplotlib.pyplot as plt plt.plot([1,2,3],[5,7,4]) plt.show()` This should allow you to display the graph using Matplotlib.

Question 3:

Tags: python, conda, miniconda

Question: I have `miniconda3` installed and since I would like to have an environment with python version 3.3.0, I create it via `conda create -n "myenv" python=3.3.0` However when I activate the environment via `conda activate myenv` python has version 2.7.15 and path `/usr/bin/python` and ipython has python version 3.6.8 and path `/home/myname/.local/bin/ipython` I can access the correct python with `python3` which is at `/home/myname/miniconda3/envs/myenv/bin/python3` however, `ipython3` has python version 3.6.8 again. `conda install python=3.3.0` left the situation unchanged. A solution would be to open IPython via `python3 -m IPython` however, while this works fine for `python` here I get the error message `/home/myname/miniconda3/envs/myenv/bin/python3: No module named IPython` Is it possible to access with the commands `python` and `ipython` both python version 3.3.0 in that specific environment, i.e. not by setting an alias in the `.bashrc`? EDIT: Turns out that this problem does not occur if you select version 3.3 instead of 3.3.0 together with @ilmarinen's answer `conda create -n "myenv" python=3.3 ipython` everything works fine and `python` as well as `ipython` result to version python 3.3.5.

Answer Score: 267

Stack Overflow answer, Verb Count: 16 , Adjective Count: 3

Stack Overflow Answer: You need to install ipython as well into your given environment `conda create -n "myenv" python=3.3.0 ipython` The conda environments are prepended to your PATH variable, so when you are trying to run the executable "ipython", Linux will not find "ipython" in your activated environment (since it doesn't exist there), but it will continue searching for it, and eventually find it wherever you have it installed.

GPT answer, Verb Count: 34 , Adjective Count: 29

GPT-4o Answer: Solution to Using Python 3.3.0 and IPython in Conda Environment To correctly set up a Conda environment with Python 3.3.0 and ensure that both `python` and `ipython` use the desired version, follow these steps: 1. Create the Environment: `conda create -n myenv python=3.3 ipython` This ensures that the `ipython` package compatible with Python 3.3 is installed during environment creation. 2. Activate the Environment: `conda activate myenv` This sets the paths for `python` and `ipython` to point to the versions installed in your Conda environment. 3. Verify the Versions: `python --version ipython --version` Both should report versions compatible with

Python 3.3.x. 4. Install Missing Modules: If `ipython` is missing modules when run, install them explicitly within the environment: `conda install ipython pip install ipython` (if `conda` does not resolve the issue). 5. Alternative for Precision: If you must use `python3.3.0` specifically, try the following: `conda create -n myenv python=3.3.0 conda activate myenv` Then install `ipython` manually using: `conda install ipython` 6. Avoid Global Conflicts: Ensure your global Python or IPython installations are not conflicting. Use: `which python which ipython` to confirm the correct paths within the Conda environment. 7. Additional Notes: If the problem persists, delete and recreate the environment with: `conda remove -n myenv --all conda create -n myenv python=3.3 ipython`

Co-Pilot, Verb Count: 19 , **Adjective Count:** 11

Co-Pilot Answer: To ensure you are using the correct version of Python and IPython in your Conda environment, follow these steps: `conda create -n "myenv" python=3.3 ipython conda activate myenv` This will create a Conda environment with Python 3.3 and IPython installed. Once you activate the environment, both `python` and `ipython` should point to the versions installed in the environment. If you still face issues, ensure your PATH is set correctly by checking the following: `which python which ipython` These commands should return paths within your Conda environment, such as `/home/myname/miniconda3/envs/myenv/bin/python` and `/home/myname/miniconda3/envs/myenv/bin/ipython`. If needed, reinstall IPython in the environment: `conda install ipython`

Question 4:

Tags: javascript, typescript

Question: I have some vanilla javascript code that takes a string input, splits the string into characters, and then matches those characters to a key on an object. `DNATranscriber = { "G": "C", "C": "G", "T": "A", "A": "U" } function toRna(sequence){ const sequenceArray = [...sequence]; const transcriptionArray = sequenceArray.map(character =>{ return this.DNATranscriber[character]; }); return transcriptionArray.join(""); } console.log(toRna("ACGTGGTCTTAA")); //Returns UGCCACGAAUU` This works as expected. I'd now like to convert this to typescript. `class Transcriptor { DNATranscriber = { G: "C", C: "G", T: "A", A: "U" } toRna(sequence: string) { const sequenceArray = [...sequence]; const transcriptionArray = sequenceArray.map(character =>{ return this.DNATranscriber[character]; }); } } export default Transcriptor` But I'm getting the following error. Element implicitly has an 'any' type because expression of type 'string' >can't be used to index type '{ "A": string; }'. No index signature with a parameter of type 'string' was found on type >{ "A": string; }.ts(7053) I thought that the issue was that I needed my object key to be a string. But converting them to strings didn't work. `DNATranscriber = { "G": "C", "C": "G", "T": "A", "A": "U" }` I'm quite confused by this. It says that no index signature with a type of string exists on my object. But I'm sure that it does. What am I doing wrong? Edit - I solved this by giving the DNATranscriber object a type of any. `DNATranscriber: any = { "G": "C", "C": "G", "T": "A", "A": "U" }`

Answer Score: 127

Stack Overflow answer, Verb Count: 40 , **Adjective Count:** 17

Stack Overflow Answer: You can fix the errors by validating your input, which is something you should do regardless of course. The following typechecks correctly, via type guarding validations `const DNATranscriber = { G: 'C', C: 'G', T: 'A', A: 'U' }; export default class Transcriptor { toRna(dna: string) { const codons = [...dna]; if (!isValidSequence(codons)) { throw Error('invalid sequence'); } const transcribedRNA = codons.map(codon => DNATranscriber[codon]); return`

```
transcribedRNA; } } function isValidSequence(values: string[]): values is
Array<keyof typeof DNATranscriber> { return values.every(isValidCodon); }
function isValidCodon(value: string): value is keyof typeof DNATranscriber {
return value in DNATranscriber; } Playground Link It is worth mentioning that you seem to
be under the misapprehension that converting JavaScript to TypeScript involves using classes. In the
following, more idiomatic version, we leverage TypeScript to improve clarity and gain stronger typing of
base pair mappings without changing the implementation. We use a function, just like the original,
because it makes sense. This is important! Converting JavaScript to TypeScript has nothing to do with
classes, it has to do with static types. const DNATranscriber = { G: 'C', C: 'G', T: 'A',
A: 'U' }; export default function toRna(dna: string) { const codons =
[...dna]; if (!isValidSequence(codons)) { throw Error('invalid sequence'); }
const transcribedRNA = codons.map(codon => DNATranscriber[codon]); return
transcribedRNA; } function isValidSequence(values: string[]): values is
Array<keyof typeof DNATranscriber> { return values.every(isValidCodon); }
function isValidCodon(value: string): value is keyof typeof DNATranscriber {
return value in DNATranscriber; } Playground Link Update: Since TypeScript 3.7, we can
write this more expressively, formalizing the correspondence between input validation and its type
implication using assertion signatures. const DNATranscriber = { G: 'C', C: 'G', T:
'A', A: 'U' } as const; type DNACodon = keyof typeof DNATranscriber; type
RNACodon = typeof DNATranscriber[DNACodon]; export default function
toRna(dna: string): RNACodon[] { const codons = [...dna];
validateSequence(codons); const transcribedRNA = codons.map(codon =>
DNATranscriber[codon]); return transcribedRNA; } function
validateSequence(values: string[]): asserts values is DNACodon[] { if
(!values.every(isValidCodon)) { throw Error('invalid sequence'); } }
function isValidCodon(value: string): value is DNACodon { return value in
DNATranscriber; } Playground Link You can read more about assertion signatures in the
TypeScript 3.7 release notes.
```

GPT answer, Verb Count: 21 , Adjective Count: 17

GPT-4o Answer: Solution for TypeScript Error in Transcription Code The error occurs because TypeScript requires explicit typing when indexing an object. Using [any](#) works but defeats the purpose of TypeScript's type safety. The proper way to resolve this is to define an explicit type for the keys and values of `DNATranscriber`. Correct Implementation:

```
type DNATranscriberMap = { [key: string]: string }; class Transcriptor { DNATranscriber: DNATranscriberMap = { G: "C", C: "G", T: "A", A: "U" }; toRna(sequence: string): string { const sequenceArray = [...sequence]; const transcriptionArray = sequenceArray.map(character => { const transcription = this.DNATranscriber[character]; if (!transcription) { throw new Error(`Invalid character: ${character}`); } return transcription; }); return transcriptionArray.join(""); } } export default Transcriptor;
```

 Key Points: 1. Define a [type](#) or [interface](#) for `DNATranscriber` to specify valid keys and values. 2. Handle invalid characters gracefully by checking for missing keys and throwing an error if necessary. 3. This approach ensures that TypeScript enforces the correct mapping and prevents runtime errors.

Co-Pilot, Verb Count: 9 , Adjective Count: 4

Co-Pilot Answer: To properly type the `DNATranscriber` object in TypeScript and avoid using the [any](#) type, you should define an interface or type that describes the structure of your object. Here's how you can do it:

```
interface DNATranscriberType { [key: string]: string; } class Transcriptor { DNATranscriber: DNATranscriberType = { G: "C", C: "G", T: "A", A: "U" }; toRna(sequence: string): string { const sequenceArray = [...sequence]; const transcriptionArray = sequenceArray.map(character => { return this.DNATranscriber[character]; }); return
```

```
transcriptionArray.join(""); } } export default Transcripor;
```

Question 5:

Tags: javascript, reactjs, material-ui, react-hooks

Question: I want to show some records in a table using React but I got this error: Invalid hook call. Hooks can only be called inside of the body of a function component. This could happen for one of the following reasons: You might have mismatching versions of React and the renderer (such as React DOM) You might be breaking the Rules of Hooks You might have more than one copy of React in the same app See for tips about how to debug and fix this problem.

```
import React, { Component } from 'react'; import { makeStyles } from '@material-ui/core/styles'; import Table from '@material-ui/core/Table'; import TableBody from '@material-ui/core/TableBody'; import TableCell from '@material-ui/core/TableCell'; import TableHead from '@material-ui/core/TableHead'; import TableRow from '@material-ui/core/TableRow'; import Paper from '@material-ui/core/Paper'; const useStyles = makeStyles(theme => ({ root: { width: '100%', marginTop: theme.spacing(3), overflowX: 'auto' }, table: { minWidth: 650 } })); class allowance extends Component { constructor() { super(); this.state = { allowances: [] }; } componentWillMount() { fetch('http://127.0.0.1:8000/allowances') .then(data => { return data.json(); }) .then(data => { this.setState({ allowances: data }); console.log('allowance state', this.state.allowances); }); } render() { const classes = useStyles(); return (<Paper className={classes.root}> <Table className={classes.table}> <TableHead> <TableRow> <TableCell>Allow ID</TableCell> <TableCell align="right">Description</TableCell> <TableCell align="right">Allow Amount</TableCell> <TableCell align="right">AllowType</TableCell> </TableRow> </TableHead> <TableBody> { this.state.allowances.map(row => (<TableRow key={row.id}> <TableCell component="th" scope="row">{row.AllowID}</TableCell> <TableCell align="right">{row.AllowDesc}</TableCell> <TableCell align="right">{row.AllowAmt}</TableCell> <TableCell align="right">{row.AllowType}</TableCell> </TableRow> )) } </TableBody> </Table> </Paper> ); } } export default allowance;
```

Answer Score: 114

Stack Overflow answer, Verb Count: 8 , Adjective Count: 4

Stack Overflow Answer: You can only call hooks from React functions. Read more here. Just convert the Allowance class component to a functional component. Working CodeSandbox demo.

```
const Allowance = () => { const [allowances, setAllowances] = useState([]); useEffect(() => { fetch('http://127.0.0.1:8000/allowances') .then(data => { return data.json(); }) .then(data => { setAllowances(data); }) .catch(err => { console.log(123123); }); }, []); const classes = useStyles(); return (<Paper className={classes.root}> <Table className={classes.table}> <TableHead> <TableRow> <TableCell> Allow ID </TableCell> <TableCell align="right"> Description </TableCell> <TableCell align="right"> Allow Amount </TableCell> <TableCell align="right"> AllowType </TableCell> </TableRow> </TableHead> <TableBody>{ allowances.map(row => (<TableRow key={row.id}> <TableCell component="th" scope="row">{row.AllowID}</TableCell> <TableCell align="right"> {row.AllowDesc}</TableCell> <TableCell align="right"> {row.AllowAmt}</TableCell> <TableCell
```

```
align="right">{row.AllowType}</TableCell> </TableRow> )) } </TableBody>
</Table> </Paper> ); }; export default Allowance;
```

GPT answer, Verb Count: 25 , Adjective Count: 11

GPT-4o Answer: Solution to "Invalid Hook Call" Error in React Component The error occurs because `useStyles` is a React Hook, and React Hooks can only be used inside function components, not class components. Your component is a class component, which violates the Rules of Hooks. To fix this issue, you can refactor the `allowance` component into a functional component or move the styles logic outside the component. Below is a refactored solution: Refactored Functional Component

```
import React, { useState, useEffect } from 'react'; import { makeStyles } from '@material-ui/core/styles'; import Table from '@material-ui/core/Table'; import TableBody from '@material-ui/core/TableBody'; import TableCell from '@material-ui/core/TableCell'; import TableHead from '@material-ui/core/TableHead'; import TableRow from '@material-ui/core/TableRow'; import Paper from '@material-ui/core/Paper';
const useStyles = makeStyles(theme => ({ root: { width: '100%', marginTop: theme.spacing(3), overflowX: 'auto' }, table: { minWidth: 650 } }));
const Allowance = () => { const classes = useStyles(); const [allowances, setAllowances] = useState([]);
useEffect(() => { fetch('http://127.0.0.1:8000/allowances') .then(response => response.json()) .then(data => { setAllowances(data); console.log('allowance state', data); }) .catch(error => console.error('Error fetching data:', error)); }, []);
return ( <table> <thead> <tr> <th>Allow ID</th> <th>Description</th> <th>Allow Amount</th> <th>Allow Type</th> </tr> </thead> <tbody> {allowances.map(row => ( <tr> <td>{row.AllowID}</td> <td>{row.AllowDesc}</td> <td>{row.AllowAmt}</td> <td>{row.AllowType}</td> </tr> ))} </tbody> </table> ); };
export default Allowance;
```

Key Changes: 1. Converted the class component into a functional component to use hooks. 2. Used `useState` for state management. 3. Used `useEffect` to fetch data when the component mounts. 4. Ensured `useStyles` is called within a functional component, adhering to the Rules of Hooks.

Co-Pilot, Verb Count: 17 , Adjective Count: 6

Co-Pilot Answer: The error you're seeing, `Invalid hook call`. Hooks can only be called inside of the body of a function component, occurs because `useStyles` is a hook, and hooks can only be used within function components, not class components. To fix this, you need to convert your class component to a function component. Here's how you can do it:

```
import React, { useState, useEffect } from 'react'; import { makeStyles } from '@material-ui/core/styles'; import Table from '@material-ui/core/Table'; import TableBody from '@material-ui/core/TableBody'; import TableCell from '@material-ui/core/TableCell'; import TableHead from '@material-ui/core/TableHead'; import TableRow from '@material-ui/core/TableRow'; import Paper from '@material-ui/core/Paper';
const useStyles = makeStyles(theme => ({ root: { width: '100%', marginTop: theme.spacing(3), overflowX: 'auto' }, table: { minWidth: 650 } }));
const Allowance = () => { const classes = useStyles(); const [allowances, setAllowances] = useState([]);
useEffect(() => { fetch('http://127.0.0.1:8000/allowances') .then(data => data.json()) .then(data => setAllowances(data)); }, []);
return ( <table> <thead> <tr> <th>Allow ID</th> <th>Description</th> <th>Allow Amount</th> <th>Allow Type</th> </tr> </thead> <tbody> {allowances.map(row => ( <tr> <td>{row.AllowID}</td> <td>{row.AllowDesc}</td> <td>{row.AllowAmt}</td> <td>{row.AllowType}</td> </tr> ))} </tbody> </table> ); };
export default Allowance;
```

Month: Jul-19

Question 1:

Tags: python, printing, f-string

Question: I am reading through a python script that takes an input of XML files and outputs an XML file. However, I do not understand the printing syntax. Can someone please explain what `f` in `print(f"...")` does? `args = parser.parser_args() print(f"Input directory: {args.input_directory}") print(f"Output directory: {args.output_directory}")`

Answer Score: 142

Stack Overflow answer, Verb Count: 21 , Adjective Count: 16

Stack Overflow Answer: The `f` means Formatted string literals and it's new in [Python 3.6](#). A formatted string literal or f-string is a string literal that is prefixed with `f` or `F`. These strings may contain replacement fields, which are expressions delimited by curly braces `{}`. While other string literals always have a constant value, formatted strings are really expressions evaluated at run time. Some examples of formatted string literals: `>>> name = "Fred" >>> f"He said his name is {name}." "He said his name is Fred." >>> name = "Fred" >>> f"He said his name is {name!r}." "He said his name is Fred." >>> f"He said his name is {repr(name)}." # repr() is equivalent to !r "He said his name is Fred." >>> width = 10 >>> precision = 4 >>> value = decimal.Decimal("12.34567") >>> f"result: {value:{width}.{precision}}" # nested fields result: 12.35 >>> today = datetime(year=2023, month=1, day=27) >>> f"{today:%B %d, %Y}" # using date format specifier January 27, 2023 >>> number = 1024 >>> f"{number:#0x}" # using integer format specifier 0x400`

GPT answer, Verb Count: 12 , Adjective Count: 17

GPT-4o Answer: The `f` in `print(f"...")` denotes an f-string, which stands for "formatted string literal" in Python. It is a concise way to include variables or expressions inside a string by placing them within curly braces `{}`. In your example: `args = parser.parser_args() print(f"Input directory: {args.input_directory}") print(f"Output directory: {args.output_directory}")` Explanation: Here, `f"Input directory: {args.input_directory}"` dynamically inserts the value of `args.input_directory` into the string. Similarly, `args.output_directory` is inserted into the second string. The `f` ensures that these variables are evaluated and their values are embedded in the output string. For example, if `args.input_directory = "/path/to/input"` and `args.output_directory = "/path/to/output"`, the output will be: `Input directory: /path/to/input Output directory: /path/to/output`

Co-Pilot, Verb Count: 14 , Adjective Count: 10

Co-Pilot Answer: The `f` in `print(f"...")` denotes an f-string, also known as a formatted string literal. It allows you to embed expressions inside string literals using curly braces `{}`. The expressions inside the braces are evaluated at runtime and formatted using the specified format. This feature was introduced in Python 3.6. `args = parser.parser_args() print(f"Input directory: {args.input_directory}") print(f"Output directory: {args.output_directory}")` In this example, `args.input_directory` and `args.output_directory` are evaluated and their values are inserted into the string at runtime.

Question 2:

Tags: javascript, angular

Question: Oke, I upgraded Angular from 6 to 8. But I stil get errors. I found on internet a solution that for a lot of users helped. But in this case it doesn't helped me. So my package.json file looks like this: `{ "name": "vital10-frontend", "version": "0.55.0", "license": "Unlicensed",`


```

"scripts": { "ng": "ng", "start": "ng serve", "hmr": "ng serve
--configuration hmr", "build": "ng build", "build:prod": "npm run sass:prod
&& npm run vitl0prod", "build:acc": "npm run sass:prod && npm run vitl0acc",
"build:test": "npm run sass:prod && npm run vitl0test", "build:dev": "npm
run sass:prod && npm run vitl0dev", "test": "ng test", "test:cover": "ng
test --code-coverage", "lint": "ng lint", "e2e": "ng e2e", "local": "ng
serve -c local", "analyze-bundle": "webpack-bundle-analyzer
dist/stats.json", "postinstall": "node patch-webpack.js", "sass:watch":
"sass --watch sass:src\\assets", "sass:prod": "sass --update -f --style
compressed --sourcemap=none sass:src\\assets", "vitl0dev": "ng build --prod
-c vitl0dev", "vitl0test": "ng build --prod -c vitl0test", "vitl0acc": "ng
build --prod -c vitl0acc", "vitl0prod": "ng build --prod -c production" },
"private": true, "dependencies": { "@angular/animations": "^8.1.2",
"@angular/cdk": "^7.3.7", "@angular/common": "^8.1.2", "@angular/compiler":
"^8.1.2", "@angular/core": "^8.1.2", "@angular/forms": "^8.1.2",
"@angular/http": "^7.2.15", "@angular/platform-browser": "^8.1.2",
"@angular/platform-browser-dynamic": "^8.1.2", "@angular/platform-server":
"^8.1.2", "@angular/router": "^8.1.2", "@types/chart.js": "^2.7.54",
"buffer": "^5.2.1", "chart.js": "^2.7.2", "core-js": "^2.5.7", "events":
"^3.0.0", "fs": "0.0.1-security", "jasmine": "^3.1.0", "ng2-charts":
"^1.6.0", "ng2-dragula": "^2.1.1", "ng2-pdf-viewer": "^5.0.1", "node-sass":
"^4.11.0", "oidc-client": "^1.5.1", "path": "^0.12.7", "pdfjs-dist":
"^2.0.489", "rxjs": "^6.5.2", "rxjs-compat": "^6.5.2", "stream": "0.0.2",
"timers": "^0.1.1", "web-animations-js": "^2.3.1", "zone.js": "^0.9.1" },
"devDependencies": { "@angular-devkit/build-angular": "^0.801.2",
"@angular/cli": "^8.1.2", "@angular/compiler-cli": "^8.1.2",
"@angular/language-service": "^8.1.2", "@angularclass/hmr": "^2.1.3",
"@compodoc/compodoc": "^1.1.10", "@types/jasmine": "^3.3.14", "@types/node":
"^12.6.8", "chai": "^4.1.2", "codelyzer": "^5.1.0", "convert-csv-to-json":
"0.0.13", "cucumber": "^4.2.1", "cucumber-html-reporter": "^5.0.0",
"cucumber-junit": "^1.7.1", "express-static": "^1.2.5", "jasmine-core":
"^3.4.0", "jasmine-reporters": "^2.3.1", "jasmine-spec-reporter": "^4.2.1",
"karma": "^4.2.0", "karma-chrome-launcher": "^3.0.0", "karma-cli": "^2.0.0",
"karma-coverage-istanbul-reporter": "^1.2.1", "karma-html-reporter":
"^0.2.7", "karma-jasmine": "^2.0.1", "karma-jasmine-html-reporter":
"^1.4.2", "karma-junit-reporter": "^1.2.0", "karma-teamcity-reporter":
"^1.1.0", "moment": "^2.22.1", "prettier": "1.13.4", "protractor": "^5.4.2",
"protractor-beautiful-reporter": "^1.2.1", "protractor-cucumber-framework":
"^5.0.0", "puppeteer": "^1.14.0", "request": "^2.87.0",
"request-promise-native": "^1.0.5", "ts-node": "^3.3.0", "tslint":
"^5.18.0", "tslint-angular": "^1.1.2", "typescript": "3.4.5",
"webpack-bundle-analyzer": "^3.3.2", "xlsx": "^0.12.13" }, "browser": {
"fs": false, "path": false, "os": false, "crypto": false, "stream": false,
"http": false, "tls": false, "zlib": false, "https": false, "net": false } }

```

And my **b** tsconfig.json file looks like this: { "compileOnSave": false,
"compilerOptions": { "outDir": "./dist/out-tsc", "baseUrl": "src",
"sourceMap": true, "declaration": false, "moduleResolution": "node",
"emitDecoratorMetadata": true, "experimentalDecorators": true, "target":
"es5", "typeRoots": ["node_modules/@types"], "lib": ["es2016", "dom"] },
"angularCompilerOptions": { "enableIvy": false } } But I still get these errors: ERROR
in ./node_modules/resolve/lib/async.js Module not found: Error: Can't
resolve 'fs' in 'C:\Source\nien\vital10-frontend\node_modules\resolve\lib'
ERROR in ./node_modules/resolve/lib/sync.js Module not found: Error: Can't

```

resolve 'fs' in 'C:\Source\nien\vital10-frontend\node_modules\resolve\lib'
ERROR in ./node_modules/resolve/lib/node-modules-paths.js Module not found:
Error: Can't resolve 'fs' in
'C:\Source\nien\vital10-frontend\node_modules\resolve\lib' ERROR in
./node_modules/tslint/lib/utils.js Module not found: Error: Can't resolve
'fs' in 'C:\Source\nien\vital10-frontend\node_modules\tslint\lib' I found a
possible solution, that you have to put some configuration in one of the node_modules. But I think that
is not a good solution. Because if you remove all the imports and then install again the imports your
configurations are gone. Thank you Oke, I removed the "fs": "0.0.1-security", I deleted all
the packages(node_modules) and did a npm install. But now I get this: (node:13880)
MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11
SIGINT listeners added. Use emitter.setMaxListeners() to increase limit
Binary found at C:\Source\nien\vital10-frontend\node_modules\node-sass\vendor\win32-x64-64\binding.node Testing binary Binary is fine >
vital10-frontend@0.55.0 postinstall C:\Source\nien\vital10-frontend > node
patch-webpack.js internal/modules/cjs/loader.js:638 throw err; ^ Error:
Cannot find module 'C:\Source\nien\vital10-frontend\patch-webpack.js' at
Function.Module._resolveFilename (internal/modules/cjs/loader.js:636:15) at
Function.Module._load (internal/modules/cjs/loader.js:562:25) at
Function.Module.runMain (internal/modules/cjs/loader.js:829:12) at startup
(internal/bootstrap/node.js:283:19) at bootstrapNodeJSCore
(internal/bootstrap/node.js:622:3) npm ERR! code ELIFECYCLE npm ERR! errno 1
npm ERR! vital10-frontend@0.55.0 postinstall: `node patch-webpack.js` npm
ERR! Exit status 1 npm ERR! npm ERR! Failed at the vital10-frontend@0.55.0
postinstall script. npm ERR! This is probably not a problem with npm. There
is likely additional logging output above. npm ERR! A complete log of this
run can be found in: npm ERR! C:\Users\nien\AppData\Roaming\npm-cache\_logs\
2019-07-23T10_25_46_184Z-debug.log PS C:\Source\nien\vital10-frontend> and
my package.json file looks like this now: { "name": "vital10-frontend", "version":
"0.55.0", "license": "Unlicensed", "scripts": { "ng": "ng", "start": "ng
serve", "hmr": "ng serve --configuration hmr", "build": "ng build",
"build:prod": "npm run sass:prod && npm run vit10prod", "build:acc": "npm
run sass:prod && npm run vit10acc", "build:test": "npm run sass:prod && npm
run vit10test", "build:dev": "npm run sass:prod && npm run vit10dev",
"test": "ng test", "test:cover": "ng test --code-coverage", "lint": "ng
lint", "e2e": "ng e2e", "local": "ng serve -c local", "analyze-bundle":
"webpack-bundle-analyzer dist/stats.json", "postinstall": "node
patch-webpack.js", "sass:watch": "sass --watch sass:src\\assets",
"sass:prod": "sass --update -f --style compressed --sourcemap=none
sass:src\\assets", "vit10dev": "ng build --prod -c vit10dev", "vit10test":
"ng build --prod -c vit10test", "vit10acc": "ng build --prod -c vit10acc",
"vit10prod": "ng build --prod -c production" }, "private": true,
"dependencies": { "@angular/animations": "^8.1.2", "@angular/cdk": "^7.3.7",
"@angular/common": "^8.1.2", "@angular/compiler": "^8.1.2", "@angular/core":
"^8.1.2", "@angular/forms": "^8.1.2", "@angular/http": "^7.2.15",
"@angular/platform-browser": "^8.1.2", "@angular/platform-browser-dynamic":
"^8.1.2", "@angular/platform-server": "^8.1.2", "@angular/router": "^8.1.2",
"@types/chart.js": "^2.7.54", "buffer": "^5.2.1", "chart.js": "^2.7.2",
"core-js": "^2.5.7", "events": "^3.0.0", "jasmine": "^3.1.0", "ng2-charts":
"^1.6.0", "ng2-dragula": "^2.1.1", "ng2-pdf-viewer": "^5.0.1", "node-sass":
"^4.11.0", "oidc-client": "^1.5.1", "path": "^0.12.7", "pdfjs-dist":
"^2.0.489", "rxjs": "^6.5.2", "rxjs-compat": "^6.5.2", "stream": "0.0.2",
"timers": "^0.1.1", "web-animations-js": "^2.3.1", "zone.js": "^0.9.1" },

```

```
"devDependencies": { "@angular-devkit/build-angular": "^0.801.2",
"@angular/cli": "^8.1.2", "@angular/compiler-cli": "^8.1.2",
"@angular/language-service": "^8.1.2", "@angularclass/hmr": "^2.1.3",
"@compodoc/compodoc": "^1.1.10", "@types/jasmine": "^3.3.14", "@types/node":
"^12.6.8", "chai": "^4.1.2", "codelyzer": "^5.1.0", "convert-csv-to-json":
"0.0.13", "cucumber": "^4.2.1", "cucumber-html-reporter": "^5.0.0",
"cucumber-junit": "^1.7.1", "express-static": "^1.2.5", "jasmine-core":
"^3.4.0", "jasmine-reporters": "^2.3.1", "jasmine-spec-reporter": "^4.2.1",
"karma": "^4.2.0", "karma-chrome-launcher": "^3.0.0", "karma-cli": "^2.0.0",
"karma-coverage-istanbul-reporter": "^1.2.1", "karma-html-reporter":
"^0.2.7", "karma-jasmine": "^2.0.1", "karma-jasmine-html-reporter":
"^1.4.2", "karma-junit-reporter": "^1.2.0", "karma-teamcity-reporter":
"^1.1.0", "moment": "^2.22.1", "prettier": "1.13.4", "protractor": "^5.4.2",
"protractor-beautiful-reporter": "^1.2.1", "protractor-cucumber-framework":
"^5.0.0", "puppeteer": "^1.14.0", "request": "^2.87.0",
"request-promise-native": "^1.0.5", "ts-node": "^3.3.0", "tslint":
"^5.18.0", "tslint-angular": "^1.1.2", "typescript": "3.4.5",
"webpack-bundle-analyzer": "^3.3.2", "xlsx": "^0.12.13" }, "browser": {
"fs": false, "path": false, "os": false, "crypto": false, "stream": false,
"http": false, "tls": false, "zlib": false, "https": false, "net": false } }
```

Still get this errors: (node:15956) MaxListenersExceededWarning: Possible EventEmmitter memory leak detected. 11 SIGINT listeners added. Use emitter.setMaxListeners() to increase limit

```
internal/modules/cjs/loader.js:638 throw err; ^ Error: Cannot find module
'C:\Source\nien\vital10-frontend\patch-webpack.js' at
Function.Module._resolveFilename (internal/modules/cjs/loader.js:636:15) at
Function.Module._load (internal/modules/cjs/loader.js:562:25) at
Function.Module.runMain (internal/modules/cjs/loader.js:829:12) at startup
(internal/bootstrap/node.js:283:19) at bootstrapNodeJSCore
(internal/bootstrap/node.js:622:3) npm ERR! code ELIFECYCLE npm ERR! errno 1
npm ERR! vital10-frontend@0.55.0 postinstall: `node patch-webpack.js` npm
ERR! Exit status 1 npm ERR! npm ERR! Failed at the vital10-frontend@0.55.0
postinstall script. npm ERR! This is probably not a problem with npm. There
is likely additional logging output above. npm ERR! A complete log of this
run can be found in: npm ERR! C:\Users\nien\AppData\Roaming\npm-cache\_logs\
2019-07-23T10_59_04_891Z-debug.log PS C:\Source\nien\vital10-frontend> i Still
get this three erros: ERROR in ./node_modules/resolve/lib/async.js Module not
found: Error: Can't resolve 'fs' in
'C:\Source\nien\vital10-frontend\node_modules\resolve\lib' ERROR in
./node_modules/resolve/lib/sync.js Module not found: Error: Can't resolve
'fs' in 'C:\Source\nien\vital10-frontend\node_modules\resolve\lib' ERROR in
./node_modules/tslint/lib/utils.js Module not found: Error: Can't resolve
'fs' in 'C:\Source\nien\vital10-frontend\node_modules\tslint\lib' i
ÔΩçwdmÔΩ£: Failed to compile.
```

Answer Score: 229

Stack Overflow answer, Verb Count: 3 , **Adjective Count:** 12

Stack Overflow Answer: The error is because of [angular-cli](#) does not support modules in node like "fs" and "path". (Issue) Add the following to the root of the "package.json" file. `"browser": { "fs": false, "path": false, "os": false }` Angular apps such as the ones Angular CLI creates are browser apps. fs isn't available on the browser, it's only available on node apps.

<https://github.com/angular/angular-cli/issues/8272#issuecomment-341428996>

GPT answer, Verb Count: 53 , Adjective Count: 35

GPT-4o Answer: To address the issues you are facing after upgrading Angular, follow these steps: 1. Resolve the "Cannot find module 'fs'" Errors The 'fs' module is a Node.js module and not available in browser environments. These errors suggest that some dependencies are trying to use Node.js modules in your front-end project. To resolve this: `npm install browserify-fs --save` Then, update your `webpack.config.js` (or create one if it does not exist) to include: `resolve: { fallback: { fs: false, path: require.resolve("path-browserify"), os: require.resolve("os-browserify/browser") } }` If you do not have direct control over webpack configuration (e.g., in an Angular CLI project), ensure the offending packages are compatible with front-end use or consider replacing them with alternatives. 2. Fix the Missing `patch-webpack.js` Script The error indicates that the `patch-webpack.js` file is missing. Either restore this file if it was deleted or remove the `postinstall` script in your `package.json`: `"postinstall": "node patch-webpack.js"` Then, remove it with: `npm install` 3. Update TSLint Configuration The TSLint-related errors indicate that some of your TSLint rules or dependencies are incompatible. Update TSLint to the latest version: `npm install tslint@latest --save-dev` Also, consider migrating from TSLint to ESLint since TSLint is deprecated. 4. MaxListenersExceededWarning This warning is caused by adding too many event listeners. Increase the limit using: `require('events').EventEmitter.defaultMaxListeners = 20;` Add this line to the top of your `webpack.config.js` or other relevant scripts. 5. Reinstall Dependencies Clear your `node_modules` folder and reinstall dependencies: `rm -rf node_modules package-lock.json npm install` 6. Ensure Angular CLI Compatibility Make sure your Angular CLI version matches your Angular version: `npm install @angular/cli@8.1.2 --save-dev` 7. Verify Your TypeScript Version Angular 8 requires TypeScript 3.4.x or later. Check and update TypeScript: `npm install typescript@3.4.5 --save-dev` Once all these steps are completed, run your project again with: `npm start`

Co-Pilot, Verb Count: 1 , Adjective Count: 0

Co-Pilot Answer: Too long no Answer given.

Question 3:

Tags: python, file-permissions, python-3.x, git-bash, windows-store-apps

Question: Seems as though an update on Windows 10 overnight broke Python. Just trying to run `python --version` returned a "Permission Denied" error. None of the three updates; KB4507453, KB4506991, or KB4509096 look like they'd be the culprit but the timing of the issue is suspicious. Rather than messing with rolling back, I'm hoping there's a simpler fix that I'm missing. The permissions on python are "-rwxr-xr-x" and I haven't changed anything besides letting the Windows update reboot machine after installing last night's patches. According to the System Information, I'm running 10.0.18362 Should also note that this is happening whether I (try) to execute Python from git-bash using "run as administrator" or not, and if I try using PowerShell, it just opens the Windows store as if the app isn't installed so I'm thinking it can't see the contents of my `/c/Users/david/AppData/Local/Microsoft/WindowsApps/` folder for some reason. I've also tried to reinstall Python 3.7.4, but that didn't help either. Is there something else I should be looking at?

Answer Score: 343

Stack Overflow answer, Verb Count: 16 , Adjective Count: 9

Stack Overflow Answer: As far as I can tell, this was caused by a conflict with the version of Python 3.7 that was recently added into the Windows Store. It looks like this added two "stubs" called `python.exe` and `python3.exe` into the `%USERPROFILE%\AppData\Local\Microsoft\WindowsApps` folder, and in my case, this was inserted before my existing Python executable's entry in the `PATH`. Moving this entry below the correct

Python folder (partially) corrected the issue. The second part of correcting it is to type `manage app execution aliases` into the Windows search prompt and disable the store versions of Python altogether. It's possible that you'll only need to do the second part, but on my system I made both changes and everything is back to normal now.

GPT answer, Verb Count: 47 , Adjective Count: 21

GPT-4o Answer: The issue is likely related to the Windows update modifying or conflicting with the PATH environment variable, or an issue with the Microsoft Store redirection. Here's how you can resolve the problem: 1. Verify Python Installation Path Ensure Python is installed in the correct directory. The typical location is:

`C:\Users\[YourUsername]\AppData\Local\Programs\Python\Python37` If Python is installed elsewhere, note the path for updating the environment variable. 2. Check PATH Environment Variable Confirm the PATH variable includes the Python installation directory and its Scripts folder: `C:\Users\[YourUsername]\AppData\Local\Programs\Python\Python37\` `C:\Users\[YourUsername]\AppData\Local\Programs\Python\Python37\Scripts\` Steps to update PATH: 1. Press Windows + S and search for "Environment Variables." 2. Select "Edit the system environment variables." 3. In the System Properties dialog, click "Environment Variables." 4. Under "System variables" or "User variables," find and edit the PATH variable. 5. Add the Python paths if missing and save the changes. 3. Disable App Execution Aliases Windows 10 may be redirecting Python commands to the Microsoft Store. To disable this: 1. Open "Settings" (Windows + I). 2. Go to "Apps" > "Apps & features." 3. Click "App execution aliases" in the right-hand menu. 4. Turn off the toggle for "App Installer" Python aliases. 4. Reinstall Python and Update PATH Automatically If the above steps do not resolve the issue, uninstall Python completely: 1. Uninstall Python via "Add or Remove Programs." 2. Reinstall Python, ensuring the "Add Python to PATH" option is checked during installation. 5. Check Permissions on Python Executables Ensure the Python executables in the installation directory have the correct permissions: 1. Navigate to the Python installation folder. 2. Right-click the Python executable (e.g., `python.exe`) and select "Properties." 3. Go to the "Security" tab and ensure your user account has "Read & Execute" permissions. 6. Test Python Execution Open a new terminal or PowerShell and run: `python --version` If the issue persists, consider removing recent Windows updates or resetting the PATH variable to a known working state.

Co-Pilot, Verb Count: 40 , Adjective Count: 14

Co-Pilot Answer: It sounds like the Windows update might have affected the permissions or paths related to Python. Here are a few steps you can try to resolve the issue: 1. Check Python Path: Ensure that the Python executable is in your system's PATH environment variable. You can do this by searching for "Environment Variables" in the Windows Start menu, and then adding the Python installation directory (e.g., `C:\Users\david\AppData\Local\Programs\Python\Python37`) to the PATH variable if it's not already there. 2. Run as Administrator: Try running the command prompt or PowerShell as an administrator and then running `python --version` again. 3. Check WindowsApps Folder: Since you mentioned the Windows Store, it might be interfering. You can try renaming the `WindowsApps` folder to something else (e.g., `WindowsApps_backup`) to see if that resolves the issue. 4. Reinstall Python: Although you mentioned you already tried reinstalling Python, ensure you completely uninstall Python first, then reinstall it from the official website. 5. Check Permissions: Verify the permissions on the Python directory and files. Right-click on the Python folder, select "Properties," go to the "Security" tab, and ensure your user account has full control. 6. Use Full Path: When running Python scripts, use the full path to the Python executable to avoid any path issues. If none of these steps work, you might need to consider rolling back the Windows update, but that should be a last resort.

Question 4:

Tags: python, matplotlib, heatmap, seaborn

Question: When plotting heatmaps with seaborn (and correlation matrices with matplotlib) the first and the last row is cut in half. This happens also when I run this minimal code example which I found online. `import pandas as pd import seaborn as sns import matplotlib.pyplot as plt data = pd.read_csv('https://raw.githubusercontent.com/resbaz/r-novice-gapminder-files/master/data/gapminder-FiveYearData.csv') plt.figure(figsize=(10,5)) sns.heatmap(data.corr()) plt.show()` The labels at the y axis are on the correct spot, but the rows aren't completely there. A few days ago, it work as intended. Since then, I installed texlive-xetex so I removed it again but it didn't solve my problem. Any ideas what I could be missing?

Answer Score: 122

Stack Overflow answer, Verb Count: 7 , Adjective Count: 5

Stack Overflow Answer: Unfortunately matplotlib 3.1.1 broke seaborn heatmaps; and in general inverted axes with fixed ticks. This is fixed in the current development version; you may hence revert to matplotlib 3.1.0 use matplotlib 3.1.2 or higher set the heatmap limits manually (`ax.set_ylim(bottom, top)` # set the ylim to bottom, top)

GPT answer, Verb Count: 16 , Adjective Count: 16

GPT-4o Answer: The issue with the first and last rows of the heatmap being cut in half is typically caused by the default settings for the axis limits or aspect ratio in Matplotlib. You can fix this by explicitly adjusting the axis limits or using the `cbar_kws` and `xticklabels/yticklabels` parameters in Seaborn. Here's the corrected code: `import pandas as pd import seaborn as sns import matplotlib.pyplot as plt # Load data data = pd.read_csv('https://raw.githubusercontent.com/resbaz/r-novice-gapminder-files/master/data/gapminder-FiveYearData.csv') # Create the heatmap with proper adjustments plt.figure(figsize=(10, 5)) sns.heatmap(data.corr(), annot=True, fmt=".2f", cbar_kws={'shrink': 0.8}, xticklabels=True, yticklabels=True) # Adjust the axis to prevent truncation plt.gca().set_ylim(len(data.corr()), -0.5) plt.show()` Explanation of changes: 1. `cbar_kws={'shrink': 0.8}`: Ensures the color bar does not interfere with the plot. 2. `plt.gca().set_ylim(len(data.corr()), -0.5)`: Fixes the y-axis limits to ensure the full rows are visible. This should resolve the truncation issue and display the heatmap properly.

Co-Pilot, Verb Count: 12 , Adjective Count: 9

Co-Pilot Answer: This issue can often be resolved by adjusting the aspect ratio or the layout of the plot to ensure all rows and labels are fully visible. Here are a few approaches you can try: 1. Adjust the Aspect Ratio: `plt.figure(figsize=(12, 6)) sns.heatmap(data.corr(), aspect=0.5) plt.show()` 2. Use `tight_layout()` to Adjust Spacing: `plt.figure(figsize=(10, 5)) sns.heatmap(data.corr()) plt.tight_layout() plt.show()` 3. Pad the Axes: `plt.figure(figsize=(10, 5)) sns.heatmap(data.corr()) plt.gca().set_xlim(-0.5, data.corr().shape[1] - 0.5) plt.gca().set_ylim(-0.5, data.corr().shape[0] - 0.5) plt.show()` These adjustments should help ensure that the entire heatmap is displayed correctly, including the first and last rows.

Question 5:

Tags: javascript, rename, object, destructuring, ecma-script-6

Question: `const a = { b: { c: 'Hi!' } }; const { b: { c } } = a;` Is it possible

Answer Score: 61

Stack Overflow answer, Verb Count: 18 , Adjective Count: 8

Stack Overflow Answer: Writing my comment as an answer: The problem is that you're calling `Event.Event()` directly, even though it is a react component. That causes react to treat the hook calls inside the function as part of `Container`, even though you meant for them to be part of `Event`. The solution is to use JSX: `return isToday && <Event.Event dayHeight={dayHeight} event={e} />` Why this works is clearer when you replace the JSX with the resulting JS code: `return isToday && React.createElement(Event.Event, { dayHeight, event: e })` See <https://reactjs.org/docs/react-api.html#createelement>. You never want to call the function components directly, how react works is that you always hand a reference the component to react and let it call the function at the correct time.

GPT answer, Verb Count: 26 , Adjective Count: 14

GPT-4o Answer: The issue arises from how React hooks must always be invoked in the same order during every render. The warning indicates that the order of hooks is changing between renders in your `Container` component. The problem lies in this part of the `Container`: `const getDaysEvents = (day: Event.Time, events: Array<Event.Event>) => { return events.map(e => { const isToday = e.startTime.hasSame(day, "day"); return isToday && Event.Event({ dayHeight, event: e }); }); }` Although it may seem like you are not conditionally invoking hooks, the `Event.Event` function may be using a hook internally. This, combined with your mapping over the `events` array, could lead to a change in the number or order of hooks called depending on the contents of the array. To fix this issue, ensure that: 1. All hooks are invoked unconditionally and always in the same order for every render. 2. Move any hook invocation out of loops, conditionals, or nested calls within `Event.Event` or `useDrop`. For example: `export const Container: FunctionComponent<Props> = () => { let events = useContext(State.StateContext); const days = Array.from({ length: 7 }, (_, i) => { const day = DateTime.today().plus({ days: i }); const dayEvents = events.filter(e => e.startTime.hasSame(day, "day")); return (<Day key={day.toISO()} height={dayHeight} date={day}> {dayEvents.map(event => (<Event key={event.id} dayHeight={dayHeight} event={event} />))} </Day>); }); return <div className="Container">{days}</div>; }` Ensure that any hooks inside `Day` or `Event` are also declared at the top level of those components and are not dynamically conditionally invoked.

Co-Pilot, Verb Count: 19 , Adjective Count: 13

Co-Pilot Answer: The error occurs because React hooks must be called in the exact same order on every render. If a hook is conditionally called or called in a different order on subsequent renders, it will result in the warning you're seeing. Ensure that all hooks are called unconditionally and in the same order. Here is an adjusted version of your `Container` component: `export const Container: FunctionComponent<Props> = () => { const events = useContext(State.StateContext); const getDaysEvents = (day: Event.Time, events: Array<Event.Event>) => { return events.map(e => { const isToday = e.startTime.hasSame(day, "day"); return isToday && Event.Event({ dayHeight, event: e }); }); }; const days = Array.from({ length: 7 }, (_, i) => { const day = DateTime.today().plus({ days: i }); return (<Day key={day.toISO()} height={dayHeight} date={day}> {getDaysEvents(day, events)} </Day>); }); return <div className="Container">{days}</div>; }` Ensure that the hooks in your `Day` component are also called at the top level and not conditionally: `const Day: FunctionComponent<DayProps> = ({ date, height, children }) => { const dispatch = useContext(State.DispatchContext); const [{ isOver, offset }, dropRef] = useDrop({ // ... }); // ... }` This should resolve the issue with the order of hooks changing between renders.

Question 2:

Tags: javascript, node.js, mjs

Question: I have started working on an existing project based on Node.js. I was just trying to understand the flow of execution, where I encountered with some `*.mjs` files. I have searched the web where I found that these are module based JS-files. I want to know how is it different from `*.js` files (how does it benefit)?

Answer Score: 830

Stack Overflow answer, Verb Count: 10 , **Adjective Count:** 12

Stack Overflow Answer: It indicates an ES6 module file. Node.js's original module system is CommonJS (which uses `require` and `module.exports`). Since Node.js was created, the ECMAScript module system (which uses `import` and `export`) has become standard and Node.js has added support for it. Node.js will treat `.cjs` files as CommonJS modules and `.mjs` files as ECMAScript modules. It will treat `.js` files as whatever the default module system for the project is (which is CommonJS unless `package.json` says `"type": "module",`). See also: Differences between ES6 module system and CommonJs

GPT answer, Verb Count: 24 , **Adjective Count:** 33

GPT-4o Answer: The `*.mjs` files are JavaScript files explicitly marked as ES Modules, whereas `*.js` files can be treated as either CommonJS or ES Modules, depending on the `type` field in the project's `package.json` file or the file's extension. Key differences and benefits: 1. Explicit Module Syntax: The `*.mjs` extension ensures the file is always treated as an ES Module regardless of the `package.json` configuration. This makes it explicit and avoids ambiguity. 2. Import/Export Syntax: ES Modules (in `*.mjs`) use the `import` and `export` syntax: `import { something } from './module.mjs'; export const value = 42;` CommonJS (typically in `*.js`) uses `require` and `module.exports`: `const something = require('./module.js'); module.exports = { value: 42 };` 3. Benefits of ES Modules: - Static structure: ES Modules are statically analyzable, meaning imports and exports can be resolved at compile time, enabling better optimization and tree-shaking by bundlers. - Improved scope management: Variables and functions in ES Modules have their own scope, reducing the risk of name collisions. - Support for asynchronous loading: ES Modules natively support `import()` for dynamic imports, enabling lazy loading of modules. 4. Compatibility: Node.js treats `*.mjs` files as ES Modules by default, ensuring compatibility with modern JavaScript standards. However, for `*.js` files, behavior depends on the `type` field in `package.json` (set to `"module"` for ES Modules or omitted/defaulted to `"commonjs"`).

Co-Pilot, Verb Count: 32 , **Adjective Count:** 27

Co-Pilot Answer: The `.mjs` file extension stands for "module JavaScript" and is used to denote ES6 (ECMAScript 2015) modules, which enable the use of `import` and `export` statements for modular code. The main differences and benefits compared to `.js` files are: 1. Strict Mode: `*.mjs` files are automatically in strict mode, which helps catch common coding mistakes and improve security. 2. Explicit Imports and Exports: ES6 modules use `import` and `export` statements to explicitly declare dependencies, making code more modular and easier to manage. `import { myFunction } from './myModule.mjs'; export const myVariable = 42;` 3. Asynchronous Loading: ES6 modules can be loaded asynchronously, improving performance by allowing parallel loading of dependencies. 4. Scoped Variables: Variables and functions declared in ES6 modules are scoped to the module, reducing the risk of naming conflicts. 5. Tree Shaking: Modern build tools can perform tree shaking on ES6 modules to remove unused code, reducing the final bundle size. `export function unusedFunction() { // This function will be included if used } function usedFunction() { // This function will be removed if not used }` In contrast, `*.js` files can still use CommonJS modules (e.g., `require` and `module.exports`), but they do not

automatically enforce strict mode, and their scope is less restrictive.

Question 3:

Tags: javascript, typescript, reactjs

Question: I have the following React component that generates an HTML Table from an array of objects. The columns that should be displayed are defined through the `tableColumns` property. When looping through `items` and displaying the correct columns I have to use the `key` property from the `tableColumn` object (`{item[column.key]}`) but typescript is generating the following error: Element implicitly has an 'any' type because expression of type 'string' can't be used to index type '{}'. No index signature with a parameter of type 'string' was found on type '{}'. What could I do to fix this? I'm lost How I call the component: `<TableGridView items=[{ id: 1, name: 'John Doe', email: 'john@doe.de' }, { id: 2, name: 'Lorem ipsum', email: 'lorem@ipsum.com', }] tableColumns=[{ key: 'id', label: 'ID', }, { key: 'name', label: 'Name', }] />` My Component: `export type TableColumn = { key: string, label: string, }; export type TableGridViewProps = { items: object[], tableColumns: TableColumn[] }; const TableGridView: React.FC<TableGridViewProps> = ({ tableColumns, items }) => { return (<table> <tbody> {items.map(item => { return (<tr> {tableColumns.map((column, index) => { return (<td key={column.key} className="lorem ipsum" > {item[column.key]} // error thrown here </td>); }}} </tr>); }}} </tbody> </table>); }`

Answer Score: 152

Stack Overflow answer, Verb Count: 21 , Adjective Count: 15

Stack Overflow Answer: `items: object[]`, While technically it is a JavaScript object, the type can be better. For Typescript to correctly help you identify mistakes when accessing objects properties, you need to tell it the exact shape of the object. If you type it as `object`, typescript cannot help you with that. Instead you could tell it the exact properties and datatypes the object has: `let assistance: { safe: string } = { safe: 1 /* typescript can now tell this is wrong */ }; assistance.unknown; // typescript can tell this wont really work too` Now in the case that the object can contain any sort of key / value pair, you can at least tell typescript what type the values (and the keys) have, by using an object index type: `items: { [key: string]: number | string, }[]` That would be the accurate type in the case given.

GPT answer, Verb Count: 23 , Adjective Count: 16

GPT-4o Answer: The error occurs because TypeScript is treating `item` as an object with no defined properties, so accessing `item[column.key]` results in a type error. You can resolve this issue by providing a more specific type for `items` and `tableColumns`. Here's the updated code: `export type TableColumn = { key: keyof Item, // Use keyof to reference properties of the item label: string, }; export type Item = { [key: string]: any; // Define a generic structure for the items }; export type TableGridViewProps = { items: Item[], // Specify the type for items tableColumns: TableColumn[] }; const TableGridView: React.FC = ({ tableColumns, items }) => { return (<table> <tbody> {items.map((item, rowIndex) => (<tr key={rowIndex}> { /* Add a key for each row */ } {tableColumns.map((column, colIndex) => (<td key={colIndex} { /* Add a key for each cell */ } className="lorem ipsum" > {item[column.key]} { /* No more error */ } </td>))) </tr>))) </tbody> </table>); }; Changes made: - Defined a generic Item type for items that allows any key-value structure. - Updated TableColumn.key to use keyof Item, ensuring type safety between columns and items. - Added key attributes for rows and cells to prevent React warnings about missing keys.`

Co-Pilot, Verb Count: 9 , Adjective Count: 1

Co-Pilot Answer: To resolve the TypeScript error, you need to ensure that TypeScript understands the structure of your items array. You can define a type for the items and use it in the TableGridViewProps.

```
Here is the updated component: type Item = { id: number; name: string; email: string; }; export type TableColumn = { key: keyof Item; label: string; }; export type TableGridViewProps = { items: Item[]; tableColumns: TableColumn[]; }; const TableGridView: React.FC<TableGridViewProps> = ({ tableColumns, items }) => { return ( <table> <tbody> {items.map(item => ( <tr key={item.id}> {tableColumns.map((column) => ( <td key={column.key} className="lorem ipsum"> {item[column.key]} </td> ))} </tr> ))} </tbody> </table> ); }; export default TableGridView;
```

Question 4:

Tags: javascript, type-conversion

Question: I was practicing some JavaScript when one of my friends came across this JavaScript code: `document.write(('b' + 'a' + + 'a' + 'a').toLowerCase());` The above code answers "banana"! Can anyone explain why?

Answer Score: 629

Stack Overflow answer, Verb Count: 14 , Adjective Count: 23

Stack Overflow Answer: `+'a'` resolves to `NaN` ("Not a Number") because it coerces a string to a number, while the character `a` cannot be parsed as a number. `document.write(+'a');` To lowercase it becomes `nan`. Adding `NaN` to `"ba"` turns `NaN` into the string `"NaN"` due to type conversion, gives `baNaN`. And then there is an `a` behind, giving `baNaNNa`. The space between `+` `+` is to make the first one string concatenation and the second one a unary plus (i.e. "positive") operator. You have the same result if you use `'ba'+(+ 'a')+'a'`, resolved as `'ba'+NaN+'a'`, which is equivalent to `'ba'+NaN+'a'` due to type juggling. `document.write('ba'+(+ 'a')+'a');`

GPT answer, Verb Count: 10 , Adjective Count: 9

GPT-4o Answer: The result of the code `('b' + 'a' + + 'a' + 'a').toLowerCase()` is "banana" because of the following steps: `'b' + 'a'` // Results in the string "ba" + `'a'` // The unary plus (+) operator tries to convert 'a' to a number. Since 'a' is not a valid number, this results in NaN. `'ba' + NaN` // Results in the string "baNaN" (string concatenation). `'baNaN' + 'a'` // Adds 'a' to the end, resulting in "baNaNNa". `'baNaNNa'.toLowerCase()` // Converts the string to lowercase, giving "banana". The key part is `+'a'`, which converts the string 'a' into `NaN` (Not-a-Number) due to the unary plus operator.

Co-Pilot, Verb Count: 9 , Adjective Count: 3

Co-Pilot Answer: The code `document.write(('b' + 'a' + + 'a' + 'a').toLowerCase());` evaluates to "banana" due to the following steps: `'b' + 'a'` results in the string "ba". `+'a'` is interpreted as a unary plus operator applied to the string 'a', which converts it to `NaN` (Not-a-Number). `"ba" + NaN` results in the string "baNaN". `"baNaN" + 'a'` results in the string "baNaNNa". `"baNaNNa".toLowerCase()` converts the string to "banana".

Question 5:

Tags: python, numpy

Question: I'm facing an issue with allocating huge arrays in numpy on Ubuntu 18 while not facing the same issue on MacOS. I am trying to allocate memory for a numpy array with shape `(156816, 36,`

the array may still fail during runtime if the system runs out of memory when actually accessing elements. 3. Strict Allocation on Ubuntu: On Ubuntu, numpy likely attempts to allocate all the memory upfront, resulting in the `MemoryError`. This behavior can be stricter, ensuring that resources are actually available before proceeding. 4. Workaround Solutions: // Option 1: Use smaller chunks `array_shape = (156816, 36, 53806)` `chunk_size = 1000` `result = []` `for i in range(0, array_shape[0], chunk_size):`
`result.append(np.zeros((min(chunk_size, array_shape[0] - i), array_shape[1], array_shape[2]), dtype='uint8'))` // Option 2: Use memory-mapped arrays `result = np.memmap('large_array.dat', dtype='uint8', mode='w+', shape=(156816, 36, 53806))` // Option 3: Reduce the dimensions `result = np.zeros((156816, 36, 1000), dtype='uint8')` # Adjust the size based on your use case 5. System Configuration: Check for limits imposed by the operating system, such as `ulimit` on Linux, which may restrict the maximum amount of memory that can be allocated. These solutions ensure you work within the system's memory constraints while handling large datasets effectively.

Co-Pilot, Verb Count: 33 , Adjective Count: 19

Co-Pilot Answer: The issue you're facing is likely due to differences in how memory allocation is handled on Ubuntu and MacOS. Here are a few steps you can take to resolve the `MemoryError` on Ubuntu: Increase Virtual Memory (Swap Space): Ensure that your Ubuntu system has enough swap space allocated. You can increase the swap space by editing the `/etc/fstab` file and adding a swap file or partition. Overcommit Memory: You can enable memory overcommitment on Ubuntu by running the following command: `sudo sysctl vm.overcommit_memory=1` This allows the system to allocate more memory than is physically available. Use a Smaller Data Type: If possible, use a smaller data type for your array, such as `np.float32` instead of `np.uint8`. This can reduce the memory footprint. Optimize Your Code: Ensure that you're not holding onto unnecessary references to large objects. Use `del` to delete large arrays when they are no longer needed. These steps should help you manage memory allocation more effectively on Ubuntu. If the issue persists, you might need to consider upgrading your hardware or using a more memory-efficient approach for your computations.

Month: Sep-19

Question 1:

Tags: javascript, typescript, reactjs

Question: I am currently migrating a React application to TypeScript. So far, this works pretty well, but I have a problem with the return types of my `render` functions, specifically in my functional components. I have always used `JSX.Element` as the return type, now this doesn't work any more if a component decides to not render anything, i.e. returns `null`, since `null` is not a valid value for `JSX.Element`. This was the beginning of my journey. I searched the web and found that you should use `ReactNode` instead, which includes `null` and a few other things that can happen. However, when creating a functional component, TypeScript complains about the `ReactNode` type. Again, after some searching I found, that for functional components you should use `ReactElement` instead. However, if I do so, the compatibility issue is gone, but now TypeScript again complains about `null` not being a valid value. To cut a long story short, I have three questions: What is the difference between `JSX.Element`, `ReactNode`, and `ReactElement`? Why do the `render` methods of class components return `ReactNode` but functional components return `ReactElement`? How do I solve this with respect to `null`?

Answer Score: 875

Stack Overflow answer, Verb Count: 17 , Adjective Count: 44

Stack Overflow Answer: What is the difference between `JSX.Element`, `ReactNode`, and `ReactElement`? A `ReactElement` is an object with `type`, `props`, and `key` properties: `interface ReactElement< P = any, T extends | string | JSXElementConstructor<any> = string | JSXElementConstructor<any>, > { type: T; props: P; key: string | null; }` A `JSX.Element` is a `ReactElement<any, any>`. It exists as various libraries can implement `JSX` in their own way: `declare global { // ,Ä namespace JSX { // ,Ä interface Element extends React.ReactElement<any, any> {} // ,Ä } // ,Ä }` A `ReactPortal` is a `ReactElement` with a `children` property: `interface ReactPortal extends ReactElement { children: ReactNode; }` A `ReactNode` is a `ReactElement`, `string`, `number`, `Iterable<ReactNode>`, `ReactPortal`, `boolean`, `null`, or `undefined`: `type ReactNode = | ReactElement | string | number | Iterable<ReactNode> | ReactPortal | boolean | null | undefined;` Example: `<div> // <- ReactElement <Component> // <- ReactElement { condition && 'text' } // <- ReactNode </Component> </div>` Why do the render methods of class components return `ReactNode`, but function components return `ReactElement`? This is due to historical reasons. A `Component.render` returns a `ReactNode`: `class Component<P, S> { // ,Ä render(): ReactNode; // ,Ä }` A `FunctionComponent` returns a `ReactElement<any, any> | null`: `interface FunctionComponent<P = {}> { (props: PropsWithChildren<P>, context?: any): ReactElement<any, any> | null; propTypes?: WeakValidationMap<P> | undefined; contextTypes?: ValidationMap<any> | undefined; defaultProps?: Partial<P> | undefined; displayName?: string | undefined; }` How do I solve this with respect to `null`? Type it as `ReactElement | null` just as React does. Or let TypeScript infer the type.

GPT answer, Verb Count: 30 , Adjective Count: 52

GPT-4o Answer: 1. What is the difference between `JSX.Element`, `ReactNode`, and `ReactElement`? `JSX.Element`: This is the type returned by JSX expressions. It represents an object created by `React.createElement` and is a subset of `ReactElement`. It does not include `null`, strings, or numbers. `ReactElement`: This is the type of an element created by `React.createElement`. It includes elements created with JSX and represents the lowest-level building block of React's virtual DOM. Like `JSX.Element`, it does not include `null`, strings, or numbers. `ReactNode`: This is a broader type that includes anything renderable by React, such as `ReactElement`, strings, numbers, fragments, arrays, booleans (treated as `null`), and `null`. It is used when a component can render multiple types of content. 2. Why do the `render` methods of class components return `ReactNode` but functional components return `ReactElement`? The return type of `render` methods in class components is `ReactNode` because class components can return `null`, strings, arrays, or other React-renderable types in addition to `ReactElement`. Functional components typically return `ReactElement` because they are designed to render a single element or fragment. However, if a functional component also needs to return `null`, it must use `ReactNode` as its return type to encompass all possibilities. 3. How do I solve this with respect to `null`? To allow functional components to return `null`, you should use `ReactNode` as the return type instead of `ReactElement`. Here's an example: `import React, { ReactNode } from 'react'; const MyComponent: React.FC = (): ReactNode => { const shouldRender = false; if (!shouldRender) { return null; } return <div>Hello, World!</div>; }; export default MyComponent;` Using `ReactNode` as the return type ensures that both `null` and valid JSX elements are acceptable.

Co-Pilot, Verb Count: 29 , Adjective Count: 46

Co-Pilot Answer: 1. Difference between `JSX.Element`, `ReactNode`, and `ReactElement`: `JSX.Element` is a type representing the value returned by JSX expressions. It corresponds to React elements generated by JSX syntax and does not include `null` or other types. `const element: JSX.Element = <div>Hello</div>;` `ReactNode` is a more general type that represents anything React can render, including strings, numbers, `ReactElements`, arrays of `ReactNodes`, `null`, and `undefined`. `type ReactNode = ReactElement | string | number | boolean | null |`

`undefined | ReactNode[]`; `ReactElement` is a type representing an element created by React. It is the return type of `React.createElement` and does not include `null` or other primitive types.

```
const element: React.ReactElement = React.createElement('div', {}, 'Hello');
```

2. Why do the `render` methods of class components return `ReactNode` but functional components return `ReactElement`? Class components' `render` method can return various types, including `JSX.Element`, `ReactElement`, strings, arrays, and `null`. Therefore, `ReactNode` is used to cover all possible return types.

```
class MyComponent extends React.Component { render(): React.ReactNode { return this.props.condition ? <div>Hello</div> : null; } }
```

Functional components typically return `JSX.Element` or `ReactElement`, but they can also return `null`. Therefore, using `ReactNode` for the return type ensures compatibility.

```
const MyComponent: React.FC = () => { return condition ? <div>Hello</div> : null; };
```

3. How to solve this with respect to `null`? To handle the possibility of `null` in functional components, you should use `ReactNode` as the return type.

```
const MyComponent: React.FC = (): React.ReactNode => { return condition ? <div>Hello</div> : null; };
```

Question 2:

Tags: javascript, mongoose, mongodb, node.js, typescript

Question: I am using Mongoose with my Node.js app and this is my configuration:

```
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true,
useUnifiedTopology: true, useCreateIndex: true, useFindAndModify: false
}).then(()=>{ console.log(`connection to database established`)
}).catch(err=>{ console.log(`db error ${err.message}`); process.exit(-1) })
```

but in the console it still gives me the warning: `DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.` What is the problem? I was not using `useUnifiedTopology` before but now it shows up in the console. I added it to the config but it still gives me this warning, why? I do not even use `MongoClient`. Edit As Felipe Plets answered there was a problem in Mongoose and they fixed this bug in later versions. So you can solve problem by updating mongoose version.

Answer Score: 273

Stack Overflow answer, Verb Count: 30 , **Adjective Count:** 17

Stack Overflow Answer: Update Mongoose 5.7.1 was release and seems to fix the issue, so setting up the `useUnifiedTopology` option work as expected.

```
mongoose.connect(mongoConnectionString, {useNewUrlParser: true,
useUnifiedTopology: true});
```

Original answer I was facing the same issue and decided to deep dive on Mongoose code: https://github.com/Automattic/mongoose/search?q=useUnifiedTopology&unscooped_q=useUnifiedTopology Seems to be an option added on version 5.7 of Mongoose and not well documented yet. I could not even find it mentioned in the library history <https://github.com/Automattic/mongoose/blob/master/History.md> According to a comment in the code: `@param {Boolean} [options.useUnifiedTopology=false]` False by default. Set to `true` to opt in to the MongoDB driver's replica set and sharded cluster monitoring engine. There is also an issue on the project GitHub about this error: <https://github.com/Automattic/mongoose/issues/8156> In my case I don't use Mongoose in a replica set or sharded cluster and though the option should be false. But if false it complains the setting should be true. Once is true it still don't work, probably because my database does not run on a replica set or sharded cluster. I've downgraded to 5.6.13 and my project is back working fine. So the only option I see for now is to downgrade it and wait for the fix to update for a newer version.

GPT answer, Verb Count: 28 , **Adjective Count:** 15

GPT-4o Answer: The issue arises because the version of Mongoose you are using does not properly

handle the `useUnifiedTopology` option internally, even though you have explicitly set it in your configuration. To resolve this problem, you need to update Mongoose to a version where this bug has been fixed. Run the following command to update Mongoose: `npm install mongoose@latest` After updating Mongoose, the `useUnifiedTopology` option will work as expected, and the deprecation warning should disappear. Ensure your updated configuration looks like this:

```
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true,
useUnifiedTopology: true, useCreateIndex: true, useFindAndModify: false
}).then(() => { console.log(`connection to database established`);
}).catch(err => { console.log(`db error ${err.message}`); process.exit(-1);
});
```

If the issue persists, verify the installed Mongoose version by running: `npm list mongoose` Ensure it reflects the latest version. For example, Mongoose version 5.7.1 or above includes proper handling of `useUnifiedTopology`.

Co-Pilot, Verb Count: 15 , Adjective Count: 8

Co-Pilot Answer: The warning you're encountering is due to a deprecation notice for the old Server Discovery and Monitoring engine. To resolve this, you need to make sure you're using the latest version of Mongoose, as the issue has been fixed in recent updates. Here's how you can do it: `npm install mongoose@latest` After updating Mongoose, your configuration should correctly use the new Server Discovery and Monitoring engine without showing the deprecation warning. If the issue persists, ensure that your Node.js environment variables are correctly set and that no old versions of Mongoose are being used in your project.

Question 3:

Tags: javascript, powershell, angular-cli, angular

Question: I have started to learn Angular but I note that powershell in Windows gives me an error whenever I make an angular command like: `ng new new-app` or `ng serve` this is the error what I got: `ng : File C:\Users\< username >\AppData\Roaming\npm\ng.ps1 cannot be loaded because running scripts is disabled on this system. For more information, see about_Execution_Policies at https://go.microsoft.com/fwlink/?LinkID=135170. At line:1 char:1 + ng serve + ~~ + CategoryInfo : SecurityError: (:) [], PSSecurityException + FullyQualifiedErrorId : UnauthorizedAccess` P.S. I try these commands in cmd and it works.

Answer Score: 905

Stack Overflow answer, Verb Count: 4 , Adjective Count: 3

Stack Overflow Answer: Remove `ng.ps1` from the directory `C:\Users\%username%\AppData\Roaming\npm\` then try clearing the npm cache at `C:\Users\%username%\AppData\Roaming\npm-cache\`

GPT answer, Verb Count: 28 , Adjective Count: 15

GPT-4o Answer: The error occurs because PowerShell has script execution policies in place, and the `ng.ps1` script cannot be executed due to the current execution policy. To resolve this, you need to change the PowerShell execution policy to allow running scripts. Follow these steps: 1. Open PowerShell as Administrator: Right-click the PowerShell icon and select "Run as Administrator." 2. Set the Execution Policy: Run the following command to set the execution policy to allow running scripts: `Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned` This command sets the policy for the current user to allow locally created scripts and those from trusted sources. 3. Confirm the Change: You will be prompted to confirm. Type `Y` and press Enter. 4. Retry Angular Commands: Now, you should be able to use Angular CLI commands in PowerShell without encountering the error: `ng new new-app` `ng serve` If you still face issues, ensure that the Node.js

and Angular CLI are correctly installed and available in your system's PATH.

Co-Pilot, Verb Count: 10 , Adjective Count: 4

Co-Pilot Answer: Open PowerShell as an Administrator and run the following command:

`Set-ExecutionPolicy RemoteSigned -Scope CurrentUser` This will allow you to run scripts that you create locally while still blocking potentially harmful scripts downloaded from the internet. After running this command, try executing the Angular commands again.

Question 4:

Tags: javascript, async-await, reactjs, material-ui, react-functional-component

Question: I'm just beginning to use React for a project, and am really struggling with incorporating async/await functionality into one of my components. I have an asynchronous function called `fetchKey` that goes and gets an access key from an API I am serving via AWS API Gateway: `const fetchKey = async authProps => { try { const headers = { Authorization: authProps.idToken // using Cognito authorizer }; const response = await axios.post("https://MY_ENDPOINT.execute-api.us-east-1.amazonaws.com/v1/", API_GATEWAY_POST_PAYLOAD_TEMPLATE, { headers: headers }); return response.data.access_token; } catch (e) { console.log(`Axios request failed! : ${e}`); return e; } };` I am using React's Material UI theme, and wanted to make use of one of its Dashboard templates. Unfortunately, the Dashboard template uses a functional stateless component: `const Dashboard = props => { const classes = useStyles(); const token = fetchKey(props.auth); console.log(token); return (... rest of the functional component's code` The result of my `console.log(token)` is a Promise, which is expected, but the screenshot in my Google Chrome browser is somewhat contradictory - is it pending, or is it resolved? Second, if I try instead `token.then((data, error)=> console.log(data, error))`, I get `undefined` for both variables. This seems to indicate to me that the function has not yet completed, and therefore has not resolved any values for `data` or `error`. Yet, if I try to place a `const Dashboard = async props => { const classes = useStyles(); const token = await fetchKey(props.auth);` React complains mightily: `> react-dom.development.js:57 Uncaught Invariant Violation: Objects are > not valid as a React child (found: [object Promise]). If you meant to > render a collection of children, use an array instead. > in Dashboard (at App.js:89) > in Route (at App.js:86) > in Switch (at App.js:80) > in div (at App.js:78) > in Router (created by BrowserRouter) > in BrowserRouter (at App.js:77) > in div (at App.js:76) > in ThemeProvider (at App.js:75)` Now, I'll be the first to state I don't have enough experience to understand what is going on with this error message. If this was a traditional React class component, I'd use the `this.setState` method to set some state, and then go on my merry way. However, I don't have that option in this functional component. How do I incorporate async/await logic into my functional React component? Edit: So I will just say I'm an idiot. The actual response object that is returned is not `response.data.access_token`. It was `response.data.Item.access_token`. Doh! That's why the result was being returned as undefined, even though the actual promise was resolved.

Answer Score: 185

Stack Overflow answer, Verb Count: 22 , Adjective Count: 10

Stack Overflow Answer: You will have to make sure two things `useEffect` is similar to `componentDidMount` and `componentDidUpdate`, so if you use `setState` here then you need to restrict the code execution at some point when used as `componentDidUpdate` as shown below: `function Dashboard() { const [token, setToken] = useState(''); useEffect(() => { // React advises to declare the async function directly inside useEffect async function getToken() { const headers = { Authorization:`

```
authProps.idToken // using Cognito authorizer }; const response = await
axios.post( "https://MY_ENDPOINT.execute-api.us-east-1.amazonaws.com/v1/",
API_GATEWAY_POST_PAYLOAD_TEMPLATE, { headers } ); const data = await
response.json(); setToken(data.access_token); }; // You need to restrict it
at some point // This is just dummy code and should be replaced by actual if
(!token) { getToken(); } }, []); return < /*Rendering code*/>; }
```

GPT answer, Verb Count: 33 , Adjective Count: 21

GPT-4o Answer: To incorporate `async/await` logic into a functional React component, you need to use the `useState` and `useEffect` hooks. Here's how you can do it: `import React, {`

```
useState, useEffect } from 'react'; import axios from 'axios'; const
fetchKey = async (authProps) => { try { const headers = { Authorization:
authProps.idToken, // using Cognito authorizer }; const response = await
axios.post( "https://MY_ENDPOINT.execute-api.us-east-1.amazonaws.com/v1/",
API_GATEWAY_POST_PAYLOAD_TEMPLATE, { headers: headers, } ); return
response.data.Item.access_token; // Adjusted key based on the Edit } catch
(e) { console.error(`Axios request failed: ${e}`); return null; } }; const
Dashboard = (props) => { const [token, setToken] = useState(null);
useEffect(() => { const getToken = async () => { const fetchedToken = await
fetchKey(props.auth); setToken(fetchedToken); }; getToken(); },
[props.auth]); if (!token) { return Loading...; // Render a loading state
while waiting for the token } return ( Access Token: {token} { /* Rest of the
functional component code */ } ); }; export default Dashboard; Explanation: 1.
```

State Management: The `useState` hook is used to manage the state of the token. 2. Side Effects: The `useEffect` hook is used to perform the asynchronous operation when the component mounts or when `props.auth` changes. 3. Loading State: While the token is being fetched, a loading message is displayed. This approach allows you to handle asynchronous logic properly in a functional component without directly rendering promises or causing React to throw errors.

Co-Pilot, Verb Count: 32 , Adjective Count: 14

Co-Pilot Answer: To incorporate `async/await` logic into your functional React component, you can use the `useEffect` hook to perform the asynchronous operation and store the result in the component's state using the `useState` hook. Here is how you can do it: `import React, { useState,`

```
useEffect } from 'react'; import axios from 'axios'; import { makeStyles }
from '@material-ui/core/styles'; const fetchKey = async authProps => { try {
const headers = { Authorization: authProps.idToken // using Cognito
authorizer }; const response = await axios.post(
"https://MY_ENDPOINT.execute-api.us-east-1.amazonaws.com/v1/",
API_GATEWAY_POST_PAYLOAD_TEMPLATE, { headers: headers } ); return
response.data.Item.access_token; } catch (e) { console.log(Axios request
failed! : ${e}); return e; } }; const useStyles = makeStyles(theme => ({ //
Your styles here })); const Dashboard = props => { const classes =
useStyles(); const [token, setToken] = useState(null); useEffect(() => {
const getToken = async () => { const fetchedToken = await
fetchKey(props.auth); setToken(fetchedToken); }; getToken(); },
[props.auth]); console.log(token); return ( <div className={classes.root}>
{token ? ( // Check if token is available before rendering dependent content
<div>Token: {token}</div> ) : ( <div>Loading...</div> ) } </div> ); }; export
default Dashboard; This approach ensures that the fetchKey function is called when the
component mounts, and the result is stored in the component's state, which triggers a re-render with
the new data.
```


Question 5:

Tags: javascript, reactjs, react-hooks

Question: We use a third party library (over which there is limited control) that takes a callback as argument to a function. What is the correct way to provide that callback with the latest state? In class components, this would be done through the use of `this`. In React hooks, due to the way state is encapsulated in the functions of `React.useState()`, if a callback gets the state through `React.useState()`, it will be stale (the value when the callback was setup). But if it sets the state, it will have access to the latest state through the passed argument. This means we can potentially get the latest state in such a callback with React hooks by setting the state to be the same as it was. This works, but is counter-intuitive. With the following code, every time the callback accesses it, it's back at its default value. The console will keep printing `Count is: 0` no matter how many times I click.

```
function Card(title) { const [count, setCount] = React.useState(0) const
[callbackSetup, setCallbackSetup] = React.useState(false) function
setupConsoleCallback(callback) { console.log("Setting up callback")
setInterval(callback, 3000) } function clickHandler() { setCount(count+1);
if (!callbackSetup) { setupConsoleCallback(() => {console.log(`Count is:
${count}`)}) setCallbackSetup(true) } } return (<div> Active count {count}
<br/> <button onClick={clickHandler}>Increment</button> </div>); } const el
= document.querySelector("#root"); ReactDOM.render(<Card title='Example
Component' />, el);
```

CodePen I've had no problem setting state within a callback, only in accessing the latest state. If I was to take a guess, I'd think that any change of state creates a new instance of the Card function. And that the callback is referring to the old one. Based on the documentation at <https://reactjs.org/docs/hooks-reference.html#functional-updates>, I had an idea to take the approach of calling `setCount` in the callback, and passing a function to `setCount`, to see if I could access the current state from within `setCount`. Replacing `setupConsoleCallback(() => {console.log(`Count is: ${count}`)})` with `setupConsoleCallback(() => {setCount(prevCount => {console.log(`Count is: ${prevCount}`); return prevCount})})` CodePen Does work. I need to call `setCount` to access the previous state. Even though I have no intention of setting the state. How can I access the latest state information from within a callback?

Answer Score: 297

Stack Overflow answer, Verb Count: 22 , Adjective Count: 18

Stack Overflow Answer: For your scenario (where you cannot keep creating new callbacks and passing them to your 3rd party library), you can use `useRef` to keep a mutable object with the current state. Like so:

```
function Card(title) { const [count, setCount] =
React.useState(0) const [callbackSetup, setCallbackSetup] =
React.useState(false) const stateRef = useRef(); // make stateRef always
have the current count // your "fixed" callbacks can refer to this object
whenever // they need the current value. Note: the callbacks will not // be
reactive - they will not re-run the instant state changes, // but they
*will* see the current value whenever they do run stateRef.current = count;
function setupConsoleCallback(callback) { console.log("Setting up callback")
setInterval(callback, 3000) } function clickHandler() { setCount(count+1);
if (!callbackSetup) { setupConsoleCallback(() => {console.log(`Count is:
${stateRef.current}`)}) setCallbackSetup(true) } } return (<div> Active
count {count} <br/> <button onClick={clickHandler}>Increment</button>
</div>); }
```

Your callback can refer to the mutable object to "read" the current state. It will capture the mutable object in its closure, and every render the mutable object will be updated with the current state value.

GPT answer, Verb Count: 23 , Adjective Count: 16

GPT-4o Answer: To access the latest state within a callback in React, you can use a ref. Refs allow you to persist a mutable value across renders without triggering a re-render. Here's how you can refactor your code:

```
function Card(title) { const [count, setCount] = React.useState(0); const [callbackSetup, setCallbackSetup] = React.useState(false); const countRef = React.useRef(count); React.useEffect(() => { countRef.current = count; // Update the ref whenever count changes }, [count]); function setupConsoleCallback(callback) { console.log("Setting up callback"); setInterval(callback, 3000); } function clickHandler() { setCount(count + 1); if (!callbackSetup) { setupConsoleCallback(() => { console.log(`Count is: ${countRef.current}`); // Access the latest state via the ref }); setCallbackSetup(true); } } return ( Active count {count} Increment ); } const el = document.querySelector("#root"); ReactDOM.render(, el);
```

Explanation: 1. `countRef` is created using `React.useRef()` and initialized with the current `count`. 2. A `useEffect` is used to update `countRef.current` whenever `count` changes. 3. Inside the callback, you access the latest state value through `countRef.current`. This approach avoids the need to call `setState` unnecessarily and provides access to the most up-to-date state inside a callback.

Co-Pilot, Verb Count: 20 , Adjective Count: 16

Co-Pilot Answer: To access the latest state within a callback in a functional React component, you can use the `useRef` hook to keep track of the current state. This ensures that the callback always has access to the most recent state value. Here is an example of how you can do this:

```
function Card(title) { const [count, setCount] = React.useState(0); const [callbackSetup, setCallbackSetup] = React.useState(false); const countRef = React.useRef(count); // Update the ref whenever count changes React.useEffect(() => { countRef.current = count; }, [count]); function setupConsoleCallback(callback) { console.log("Setting up callback"); setInterval(callback, 3000); } function clickHandler() { setCount(count + 1); if (!callbackSetup) { setupConsoleCallback(() => { console.log(`Count is: ${countRef.current}`); }); setCallbackSetup(true); } } return ( <div> Active count {count} <br/> <button onClick={clickHandler}>Increment</button> </div> ); } const el = document.querySelector("#root"); ReactDOM.render(<Card title='Example Component' />, el);
```

In this example, `countRef` is used to store the current value of `count`. The `useEffect` hook updates `countRef.current` whenever `count` changes. This way, the callback can always access the latest state value through `countRef.current`.

Month: Oct-19

Question 1:

Tags: javascript, reactjs, react-native

Question: After upgrading to React Native 0.61, I get a lot of warnings like this: `VirtualizedLists should never be nested inside plain ScrollViews with the same orientation - use another VirtualizedList-backed container instead.` What is the other `VirtualizedList-backed container` that I should use, and why is it now advised not to use it like that?

Answer Score: 172

Stack Overflow answer, Verb Count: 6 , Adjective Count: 2

Stack Overflow Answer: If someone's still looking for a suggestion to the problem that Ponleu and David Schilling have described here (regarding content that goes above the `FlatList`), then this is the

approach I took: `<SafeAreaView style={{flex: 1}}> <FlatList data={data}
ListHeaderComponent={ContentThatGoesAboveTheFlatList}
ListFooterComponent={ContentThatGoesBelowTheFlatList} /> </SafeAreaView>` You can read more about this in `ListHeaderComponent`.

GPT answer, Verb Count: 29 , Adjective Count: 29

GPT-4o Answer: The warning indicates that nesting a `VirtualizedList` (e.g., `FlatList` or `SectionList`) inside a `ScrollView` with the same scroll direction can lead to performance and usability issues, such as improper scrolling behavior or unnecessary rendering. The recommended `VirtualizedList-backed container` you can use is another `FlatList` or `SectionList`. These are optimized for rendering large lists by rendering only the visible items, thus avoiding performance issues associated with `ScrollView`, which renders all its children at once. To fix the issue, you can replace your parent `ScrollView` with a `FlatList` or refactor your code to avoid nesting. Here's an example: `import React from 'react'; import { FlatList, Text, View } from 'react-native'; const App = () => { const data = Array.from({ length: 50 }, (_, i) => `Item ${i + 1}`); return (index.toString()) renderItem={({ item }) => ({item})} />); }; export default App;` Why is this change advised? Nesting `ScrollView` with `VirtualizedLists` can lead to: 1. Over-rendering: `ScrollView` loads all its children into memory, which can be costly for large lists. 2. Scroll conflicts: Both the `ScrollView` and the nested list may handle scroll events, causing issues with touch gestures and scrolling behavior. Using a single `VirtualizedList` ensures better performance and avoids these problems.

Co-Pilot, Verb Count: 12 , Adjective Count: 12

Co-Pilot Answer: The other `VirtualizedList-backed container` you should use is `FlatList` or `SectionList`. Nesting `VirtualizedLists` inside plain `ScrollViews` can break windowing and other functionalities, leading to performance issues. Instead, use `FlatList` or `SectionList` directly, as they are optimized for rendering large lists efficiently. This ensures that only the visible items are rendered, improving performance and avoiding the warning.

Question 2:

Tags: python, code-formatting, python-black

Question: I would like to ignore a specific multi-line code by `black` python formatter. Particularly, this is used for `np.array` or matrix construction which turned ugly when formatted. Below is the example. `np.array([[1, 0, 0, 0], [0, -1, 0, 0], [0, 0, 1, 0], [0, 0, 0, -1],]) # Will be formatted to np.array([[1, 0, 0, 0], [0, -1, 0, 0], [0, 0, 1, 0], [0, 0, 0, -1]])` I found this issue in `black` github, but that only works for inline command, which is not what I have here. Is there anything I can do to achieve this for a multi-line code?

Answer Score: 467

Stack Overflow answer, Verb Count: 10 , Adjective Count: 4

Stack Overflow Answer: You can use `#fmt: on/off` (docs) as explained in the issue linked. Here, it would look like: `# fmt: off np.array([[1, 0, 0, 0], [0, -1, 0, 0], [0, 0, 1, 0], [0, 0, 0, -1],]) # fmt: on # fmt: off` disables formatting for all following lines until re-activated with `# fmt: on`.

GPT answer, Verb Count: 10 , Adjective Count: 14

GPT-4o Answer: You can use `# fmt: off` and `# fmt: on` comments to disable and re-enable `black` formatting around the specific multi-line code. Here's how you can do it: `# fmt: off np.array([[1, 0, 0, 0], [0, -1, 0, 0], [0, 0, 1, 0], [0, 0, 0, -1],]) # fmt: on` With this approach, `black` will leave the specified section untouched, preserving the

multi-line formatting as desired.

Co-Pilot, Verb Count: 7 , Adjective Count: 10

Co-Pilot Answer: To ignore a specific multi-line code block with Black, you can use the `# fmt: off` and `# fmt: on` comments. Here's how you can apply it to your example: `fmt: off np.array([[1, 0, 0, 0], [0, -1, 0, 0], [0, 0, 1, 0], [0, 0, 0, -1],]) fmt: on` This will prevent Black from reformatting the multi-line array construction, keeping it as you intended.

Question 3:

Tags: python, performance-testing, keras, tensorflow, tensorflow2.0

Question: It's been cited by many users as the reason for switching to Pytorch, but I've yet to find a justification/explanation for sacrificing the most important practical quality, speed, for eager execution. Below is code benchmarking performance, TF1 vs. TF2 - with TF1 running anywhere from 47% to 276% faster. My question is: what is it, at the graph or hardware level, that yields such a significant slowdown? Looking for a detailed answer - am already familiar with broad concepts. Relevant Git Specs: CUDA 10.0.130, cuDNN 7.4.2, Python 3.7.4, Windows 10, GTX 1070 Benchmark results: UPDATE: Disabling Eager Execution per below code does not help. The behavior, however, is inconsistent: sometimes running in graph mode helps considerably, other times it runs slower relative to Eager. Benchmark code: `# use tensorflow.keras... to benchmark tf.keras; used GPU for all above benchmarks from keras.layers import Input, Dense, LSTM, Bidirectional, Conv1D from keras.layers import Flatten, Dropout from keras.models import Model from keras.optimizers import Adam import keras.backend as K import numpy as np from time import time batch_shape = (32, 400, 16) X, y = make_data(batch_shape) model_small = make_small_model(batch_shape) model_small.train_on_batch(X, y) # skip first iteration which builds graph timeit(model_small.train_on_batch, 200, X, y) K.clear_session() # in my testing, kernel was restarted instead model_medium = make_medium_model(batch_shape) model_medium.train_on_batch(X, y) # skip first iteration which builds graph timeit(model_medium.train_on_batch, 10, X, y) Functions used: def timeit(func, iterations, *args): t0 = time() for _ in range(iterations): func(*args) print("Time/iter: %.4f sec" % ((time() - t0) / iterations)) def make_small_model(batch_shape): ipt = Input(batch_shape=batch_shape) x = Conv1D(128, 400, strides=4, padding='same')(ipt) x = Flatten()(x) x = Dropout(0.5)(x) x = Dense(64, activation='relu')(x) out = Dense(1, activation='sigmoid')(x) model = Model(ipt, out) model.compile(Adam(lr=1e-4), 'binary_crossentropy') return model def make_medium_model(batch_shape): ipt = Input(batch_shape=batch_shape) x = Bidirectional(LSTM(512, activation='relu', return_sequences=True))(ipt) x = LSTM(512, activation='relu', return_sequences=True)(x) x = Conv1D(128, 400, strides=4, padding='same')(x) x = Flatten()(x) x = Dense(256, activation='relu')(x) x = Dropout(0.5)(x) x = Dense(128, activation='relu')(x) x = Dense(64, activation='relu')(x) out = Dense(1, activation='sigmoid')(x) model = Model(ipt, out) model.compile(Adam(lr=1e-4), 'binary_crossentropy') return model def make_data(batch_shape): return np.random.randn(*batch_shape), np.random.randint(0, 2, (batch_shape[0], 1))`

Answer Score: 129

Stack Overflow answer, Verb Count: 116 , Adjective Count: 155

Stack Overflow Answer: UPDATE 8/1730/2020: TF 2.3 has finally done it: all cases run as fast, or notably faster, than any previous version. Further, my previous update was unfair to TF; my GPU was

to blame, has been overheating lately. If you see a rising stem plot of iteration times, it's a reliable symptom. Lastly, see a dev's note on Eager vs Graph. This might be my last update on this answer. The true stats on your model's speed can only be found by you, on your device. UPDATE 5/19/2020: TF 2.2, using same tests: only a minor improvement in Eager speed. Plots for Large-Large Numpy [train_on_batch](#) case below, x-axis is successive fit iterations; my GPU isn't near its full capacity, so doubt it's throttling, but iterations do get slower over time. Per above, Graph and Eager are 1.56x and 1.97x slower than their TF1 counterparts, respectively. Unsure I'll debug this further, as I'm considering switching to Pytorch per TensorFlow's poor support for custom / low-level functionality. I did, however, open an Issue to get devs' feedback. UPDATE 2/18/2020: I've benched 2.1 and 2.1-nightly; the results are mixed. All but one configs (model & data size) are as fast as or much faster than the best of TF2 & TF1. The one that's slower, and slower dramatically, is Large-Large - esp. in Graph execution (1.6x to 2.5x slower). Furthermore, there are extreme reproducibility differences between Graph and Eager for a large model I tested - one not explainable via randomness/compute-parallelism. I can't currently present reproducible code for these claims per time constraints, so instead I strongly recommend testing this for your own models. Haven't opened a Git issue on these yet, but I did comment on the original - no response yet. I'll update the answer(s) once progress is made. VERDICT: it isn't, IF you know what you're doing. But if you don't, it could cost you, lots - by a few GPU upgrades on average, and by multiple GPUs worst-case. THIS ANSWER: aims to provide a high-level description of the issue, as well as guidelines for how to decide on the training configuration specific to your needs. For a detailed, low-level description, which includes all benchmarking results + code used, see my other answer. I'll be updating my answer(s) w/ more info if I learn any - can bookmark / "star" this question for reference. ISSUE SUMMARY: as confirmed by a TensorFlow developer, Q. Scott Zhu, TF2 focused development on Eager execution & tight integration w/ Keras, which involved sweeping changes in TF source - including at graph-level. Benefits: greatly expanded processing, distribution, debug, and deployment capabilities. The cost of some of these, however, is speed. The matter, however, is fairly more complex. It isn't just TF1 vs. TF2 - factors yielding significant differences in train speed include: TF2 vs. TF1 Eager vs. Graph mode [keras](#) vs. [tf.keras.numpy](#) vs. [tf.data.Dataset](#) vs. ... [train_on_batch\(\)](#) vs. [fit\(\)](#) GPU vs. CPU [model\(x\)](#) vs. [model.predict\(x\)](#) vs. ... Unfortunately, almost none of the above are independent of the other, and each can at least double execution time relative to another. Fortunately, you can determine what'll work best systematically, and with a few shortcuts - as I'll be showing. WHAT SHOULD I DO? Currently, the only way is - experiment for your specific model, data, and hardware. No single configuration will always work best - but there are do's and don't's to simplify your search: >> DO: [train_on_batch\(\)](#) + [numpy](#) + [tf.keras](#) + TF1 + Eager/Graph [train_on_batch\(\)](#) + [numpy](#) + [tf.keras](#) + TF2 + Graph [fit\(\)](#) + [numpy](#) + [tf.keras](#) + TF1/TF2 + Graph + large model & data >> DON'T: [fit\(\)](#) + [numpy](#) + [keras](#) for small & medium models and data [fit\(\)](#) + [numpy](#) + [tf.keras](#) + TF1/TF2 + Eager [train_on_batch\(\)](#) + [numpy](#) + [keras](#) + TF1 + Eager [Major] [tf.python.keras](#); it can run 10-100x slower, and w/ plenty of bugs; more info This includes [layers](#), [models](#), [optimizers](#), & related "out-of-box" usage imports; ops, utils, & related 'private' imports are fine - but to be sure, check for alts, & whether they're used in [tf.keras](#) Refer to code at bottom of my other answer for an example benchmarking setup. The list above is based mainly on the "BENCHMARKS" tables in the other answer. LIMITATIONS of the above DO's & DON'T's: This question's titled "Why is TF2 much slower than TF1?", and while its body concerns training explicitly, the matter isn't limited to it; inference, too, is subject to major speed differences, even within the same TF version, import, data format, etc. - see this answer. RNNs are likely to notably change the data grid in the other answer, as they've been improved in TF2 Models primarily used [Conv1D](#) and [Dense](#) - no RNNs, sparse data/targets, 4/5D inputs, & other configs Input data limited to [numpy](#) and [tf.data.Dataset](#), while many other formats exist; see other answer GPU was used; results will differ on a CPU. In fact, when I asked the question, my CUDA wasn't properly configured, and some of the results were CPU-based. Why did TF2 sacrifice the most practical quality, speed, for eager execution? It hasn't, clearly - graph is still available. But if the question is "why eager at all": Superior debugging: you've likely come across multitudes of questions asking "how do I get intermediate layer outputs" or "how do I inspect weights"; with eager, it's (almost) as simple as [.__dict__](#). Graph, in contrast, requires familiarity with special backend functions - greatly

complicating the entire process of debugging & introspection. Faster prototyping: per ideas similar to above; faster understanding = more time left for actual DL. HOW TO ENABLE/DISABLE EAGER? `tf.enable_eager_execution()` # TF1; must be done before any model/tensor creation `tf.compat.v1.disable_eager_execution()` # TF2; above holds Misleading in TF2; see here. ADDITIONAL INFO: Careful with `_on_batch()` methods in TF2; according to the TF dev, they still use a slower implementation, but not intentionally - i.e. it's to be fixed. See other answer for details. REQUESTS TO TENSORFLOW DEVS: Please fix `train_on_batch()`, and the performance aspect of calling `fit()` iteratively; custom train loops are important to many, especially to me. Add documentation / docstring mention of these performance differences for users' knowledge. Improve general execution speed to keep peeps from hopping to Pytorch. ACKNOWLEDGEMENTS: Thanks to Q. Scott Zhu, TensorFlow developer, for his detailed clarification on the matter. P. Andrey for sharing useful testing, and discussion. UPDATES: 11/14/19 - found a model (in my real application) that that runs slower on TF2 for all* configurations w/ Numpy input data. Differences ranged 13-19%, averaging 17%. Differences between `keras` and `tf.keras`, however, were more dramatic: 18-40%, avg. 32% (both TF1 & 2). (* - except Eager, for which TF2 OOM'd) 11/17/19 - devs updated `on_batch()` methods in a recent commit, stating to have improved speed - to be released in TF 2.1, or available now as `tf-nightly`. As I'm unable to get latter running, will delay benching until 2.1. 2/20/20 - prediction performance is also worth benching; in TF2, for example, CPU prediction times can involve periodic spikes

GPT answer, Verb Count: 46 , Adjective Count: 43

GPT-4o Answer: The significant slowdown in TensorFlow 2 (TF2) compared to TensorFlow 1 (TF1) can be attributed to several architectural and execution model changes: 1. Eager Execution Default In TensorFlow 2, eager execution is enabled by default. Unlike TF1's static computational graph approach, eager execution evaluates operations immediately, providing better debugging and intuitive APIs but at the cost of optimization opportunities. In contrast, TF1's graph mode allows aggressive graph-level optimizations like operation fusion, memory reuse, and parallel execution that can significantly enhance performance. 2. Mixed Graph and Eager Modes Even when disabling eager execution in TF2, the framework may not fully revert to the optimized static graph mode of TF1. Certain operations may still execute in a hybrid mode, introducing overhead. Additionally, TF2's focus on compatibility with eager execution can reduce graph-level optimizations. 3. API Overheads TF2's high-level Keras APIs introduce abstraction layers that simplify user experience but can add computational overhead compared to TF1's low-level APIs. These abstractions may introduce redundant operations or inefficiencies that affect performance. 4. Backward Compatibility Layers TF2 includes compatibility features to support TF1 codebases, such as ``tf.compat.v1``. These layers introduce additional complexity and overhead, potentially affecting runtime performance, even when using graph mode. 5. Model Compilation Changes TF2 redefined the model compilation and execution processes to align with eager execution. The runtime may execute more checks or intermediate conversions to support eager execution, leading to additional overhead. 6. Python Overheads Eager execution incurs Python-level overhead because operations are executed step-by-step in Python rather than as a single compiled graph. This is particularly impactful for small operations with frequent context switching between Python and the TensorFlow runtime. Potential Solutions to Mitigate Performance Issues: # Disable eager execution explicitly `import tensorflow as tf`
`tf.compat.v1.disable_eager_execution()` # Use `tf.function` to optimize functions in TF2 `@tf.function` def `optimized_function(...)`: ... # Use low-level APIs where possible # Avoid unnecessary abstractions for performance-critical components In conclusion, the slowdown arises from TF2's prioritization of usability and eager execution, which sacrifices some of the performance benefits of TF1's graph mode. Depending on your use case, disabling eager execution, optimizing critical paths with ``tf.function``, or using TF1-style workflows via ``tf.compat.v1`` can help regain performance.

Co-Pilot, Verb Count: 30 , Adjective Count: 22

Co-Pilot Answer: TensorFlow 2.x introduced Eager Execution by default, which allows operations to

execute immediately as they are called, rather than being added to a computation graph to be run later. This makes debugging and development more intuitive but can lead to performance slowdowns compared to TensorFlow 1.x, which relies on static computation graphs for optimized performance. Several factors contribute to the performance differences: 1. Graph Optimization: TensorFlow 1.x uses static graphs, which allow for numerous optimizations such as operation fusion, constant folding, and layout optimizations. These optimizations are harder to achieve with dynamic execution. 2. Overhead of Dynamic Execution: Eager Execution introduces overhead due to the dynamic nature of Python function calls, memory allocation, and garbage collection, which can impact performance. 3. Reduced Batch Processing Efficiency: Static graphs in TensorFlow 1.x can batch process more efficiently, reducing the number of calls to the GPU, thus minimizing context switching and overhead. 4. Inconsistent Behavior: Eager Execution can have inconsistent performance due to the variability in dynamic execution paths, making it harder to achieve consistent optimizations compared to static graphs. To mitigate these performance issues in TensorFlow 2.x, you can disable Eager Execution and use the `tf.function` decorator to convert Python functions into graph operations: `import tensorflow as tf @tf.function def train_step(inputs): # Your training step implementation` This approach allows you to benefit from the performance optimizations of static graphs while still using the TensorFlow 2.x API.

Question 4:

Tags: javascript, node.js, node-modules, package.json, npm-package

Question: I've got an ApolloServer project that's giving me trouble, so I thought I might update it and ran into issues when using the latest Babel. My "index.js" is: `require('dotenv').config() import {startServer} from './server' startServer()` And when I run it I get the error `SyntaxError: Cannot use import statement outside a module` First I tried doing things to convince TPTB* that this was a module (with no success). So I changed the "import" to a "require" and this worked. But now I have about two dozen "imports" in other files giving me the same error. *I'm sure the root of my problem is that I'm not even sure what's complaining about the issue. I sort of assumed it was Babel 7 (since I'm coming from Babel 6 and I had to change the presets) but I'm not 100% sure. Most of what I've found for solutions don't seem to apply to straight Node. Like this one here: `ES6 module Import` giving `"Uncaught SyntaxError: Unexpected identifier"` Says it was resolved by adding `"type=module"` but this would typically go in the HTML, of which I have none. I've also tried using my project's old presets: `"presets": ["es2015", "stage-2"], "plugins": []` But that gets me another error: `"Error: Plugin/Preset files are not allowed to export objects, only functions."` Here are the dependencies I started with: `"dependencies": { "@babel/polyfill": "^7.6.0", "apollo-link-error": "^1.1.12", "apollo-link-http": "^1.5.16", "apollo-server": "^2.9.6", "babel-preset-es2015": "^6.24.1",`

Answer Score: 809

Stack Overflow answer, Verb Count: 13 , **Adjective Count:** 18

Stack Overflow Answer: Verify that you have the latest version of Node.js installed (or, at least 13.2.0+). Then do one of the following, as described in the documentation: Option 1 In the nearest parent `package.json` file, add the top-level `"type"` field with a value of `"module"`. This will ensure that all `.js` and `.mjs` files are interpreted as ES modules. You can interpret individual files as CommonJS by using the `.cjs` extension. `// package.json { "type": "module" }` Option 2 Explicitly name files with the `.mjs` extension. All other files, such as `.js` will be interpreted as CommonJS, which is the default if `type` is not defined in `package.json`.

GPT answer, Verb Count: 44 , **Adjective Count:** 31

GPT-4o Answer: The error occurs because Node.js does not support ES6 `import` syntax by default unless you enable module support or transpile your code using a tool like Babel. Here's how to resolve the issue: 1. Use Node.js Native ES Modules If you're using a recent version of Node.js (`>= 12.17.0` or

>= 14.0.0), you can enable ES module support by adding the following to your `package.json`: `{ "type": "module" }` With this setting, you can use `import` and `export` syntax directly in your Node.js files. However, ensure all your files have the `.mjs` extension or `.js` if specified with `type: "module"`. 2. Transpile Using Babel If you prefer to use Babel, set up Babel to transpile ES6+ syntax to CommonJS. Update your Babel configuration and dependencies as follows: // Install necessary Babel dependencies `npm install @babel/core @babel/node @babel/preset-env --save-dev` Create a `.babelrc` file: `{ "presets": ["@babel/preset-env"] }` Modify your start script in `package.json`: `"scripts": { "start": "babel-node index.js" }` Now, Babel will transpile your ES6+ code, allowing you to use `import` syntax. 3. Temporary Workaround Using CommonJS If you prefer not to change your project setup immediately, replace `import` with `require` statements throughout your codebase: `const { startServer } = require('./server'); startServer();` 4. Ensure Consistency in Babel Dependencies The error regarding presets ("Plugin/Preset files are not allowed to export objects...") suggests that older Babel presets are incompatible with Babel 7. Update your Babel dependencies: `npm install --save-dev @babel/preset-env` Then, update your `.babelrc` file: `{ "presets": ["@babel/preset-env"] }` These steps should resolve the issue and allow you to use ES6 imports seamlessly in your project.

Co-Pilot, Verb Count: 20 , Adjective Count: 17

Co-Pilot Answer: The error `SyntaxError: Cannot use import statement outside a module` indicates that your Node.js environment is not recognizing your file as a module. To resolve this, you need to ensure that your project is set up to use ES6 modules. Here's how you can do it: 1. Update Babel Configuration: Replace your Babel presets with the latest ones compatible with Babel 7. Create a `.babelrc` file with the following content: `{ "presets": ["@babel/preset-env"] }` 2. Use `type: module` in `package.json`: Update your `package.json` to include: `{ "type": "module", "scripts": { "start": "node -r dotenv/config index.js" } }` 3. Ensure Proper Imports: Update your `index.js` to use ES6 import syntax: `import 'dotenv/config'; import { startServer } from './server'; startServer();` With these changes, your Node.js environment should correctly recognize and run your ES6 modules.

Question 5:

Tags: javascript, cookies, google-chrome, samesite

Question: Since the last update, I'm having an error with cookies, related with SameSite attribute. The cookies are from third party developers (Fontawesome, jQuery, Google Analytics, Google reCaptcha, Google Fonts, etc.) The errors in the Chrome console are like this. A cookie associated with a cross-site resource at <URL> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <URL> and <URL>. (index):1 A cookie associated with a cross-site resource at `http://jquery.com/` was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at `https://www.chromestatus.com/feature/5088147346030592` and `https://www.chromestatus.com/feature/5633521622188032`. (index):1 A cookie associated with a cross-site resource at `http://fontawesome.com/` was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at

<https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>. (index):1 A cookie associated with a cross-site resource at <http://google.com/> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>. (index):1 A cookie associated with a cross-site resource at <https://google.com/> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>. (index):1 A cookie associated with a cross-site resource at <https://www.google.com/> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>. (index):1 A cookie associated with a cross-site resource at <http://www.google.com/> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>. (index):1 A cookie associated with a cross-site resource at <http://gstatic.com/> was set without the `SameSite` attribute. A future release of Chrome will only deliver cookies with cross-site requests if they are set with `SameSite=None` and `Secure`. You can review cookies in developer tools under Application>Storage>Cookies and see more details at <https://www.chromestatus.com/feature/5088147346030592> and <https://www.chromestatus.com/feature/5633521622188032>. Is there anything I need to do in my local machine or server or is just some feature they should implement in future releases of their libraries?

Answer Score: 185

Stack Overflow answer, Verb Count: 24 , **Adjective Count:** 22

Stack Overflow Answer: This console warning is not an error or an actual problem, Chrome is just spreading the word about this new standard to increase developer adoption. It has nothing to do with your code. It is something their web servers will have to support. Release date for a fix is February 4, 2020 per: <https://www.chromium.org/updates/same-site> February, 2020: Enforcement rollout for Chrome 80 Stable: The SameSite-by-default and SameSite=None-requires-Secure behaviors will begin rolling out to Chrome 80 Stable for an initial limited population starting the week of February 17, 2020, excluding the US President, Donald Trump's Day holiday on Monday. We will be closely monitoring and evaluating ecosystem impact from this initial limited phase through gradually increasing rollouts. For the full Chrome release schedule, see here. I solved same problem by adding in response header `response.setHeader("Set-Cookie", "HttpOnly;Secure;SameSite=Strict");`

[SameSite](#) prevents the browser from sending the cookie along with cross-site requests. The main goal is mitigating the risk of cross-origin information leakage. It also provides some protection against cross-site request forgery attacks. Possible values for the flag are Lax or Strict. SameSite cookies explained here Please refer this before applying any option.

GPT answer, Verb Count: 43 , Adjective Count: 31

GPT-4o Answer: The warnings about the [SameSite](#) attribute in cookies are related to a recent security update in Chrome and other browsers. The [SameSite](#) attribute determines whether cookies are sent with cross-site requests. To resolve this issue, here's what you need to know: 1. No Action Required for Third-Party Libraries If the cookies are being set by third-party libraries (e.g., FontAwesome, jQuery, Google Analytics, etc.), the responsibility to update the [SameSite](#) and [Secure](#) attributes lies with the respective library providers. You don't need to take any action for those libraries other than ensuring you are using their latest versions, as they will likely update their implementations to comply with the new browser requirements. 2. Update Your Own Server Cookies (If Applicable) If your server sets cookies that are used in cross-site requests, you should explicitly set the [SameSite](#) and [Secure](#) attributes. For example: `Set-Cookie: myCookie=value; SameSite=None; Secure` The [SameSite=None](#) attribute ensures the cookie will be sent with cross-site requests, and the [Secure](#) attribute ensures it will only be sent over HTTPS connections. 3. Local Development Environment If you are testing locally and do not use HTTPS, you may encounter issues because the [Secure](#) attribute requires an HTTPS connection. Use a self-signed certificate or tools like [mkcert](#) to enable HTTPS in your local environment. 4. Browser Feature Flag As a temporary workaround for local testing, you can disable this enforcement in Chrome:

`chrome://flags/#same-site-by-default-cookies` However, this is not recommended for production environments and should only be used for debugging purposes. 5. Monitoring Updates Keep an eye on the documentation and updates of the third-party libraries you use. Ensure you regularly update their versions to benefit from compliance with browser security policies.

Co-Pilot, Verb Count: 33 , Adjective Count: 14

Co-Pilot Answer: It seems like the issue is related to the SameSite attribute in cookies, which is a security feature implemented by browsers to prevent cross-site request forgery (CSRF) attacks. Since the latest updates, browsers require cookies to have the SameSite attribute set to either Lax or None if they are to be used in cross-site requests. Additionally, if SameSite is set to None, the cookie must also be marked as Secure, meaning it can only be sent over HTTPS. To fix this, you need to update the cookies set by third-party developers to include the SameSite attribute and ensure they are secure. Here's how you can do it: 1. Update your server-side code to set the SameSite attribute for cookies. For example, if you are using Node.js, you can set the SameSite attribute like this:

```
res.cookie('cookieName', 'cookieValue', { sameSite: 'None', secure: true });
```

2. Contact the third-party developers to ensure their libraries or services are updated to include the SameSite attribute in their cookies. 3. Check your browser settings to ensure that cookies are being handled correctly. By making these changes, you should be able to resolve the SameSite attribute errors in your ApolloServer project. Does this help with your issue?

Month: Nov-19

Question 1:

Tags: java, android, android-asynctask, kotlin, kotlin-coroutines

Question: Google is deprecating Android AsyncTask API in Android 11 and suggesting to use `java.util.concurrent` instead. you can check out the commit here * * [@deprecated Use the standard <code>java.util.concurrent</code> or * Kotlin concurrency utilities instead. */ @Deprecated public abstract class](#)

`AsyncTask<Params, Progress, Result> {` If you're maintaining an older codebase with asynchronous tasks in Android, you're likely going to have to change it in future. My question is that what should be proper replacement of the code snippet shown below using `java.util.concurrent`. It is a static inner class of an Activity. I am looking for something that will work with `minSdkVersion 16`

```
private static class LongRunningTask extends AsyncTask<String, Void, MyPojo> {
    private static final String TAG =
    MyActivity.LongRunningTask.class.getSimpleName();
    private WeakReference<MyActivity> activityReference;
    LongRunningTask(MyActivity context) {
        activityReference = new WeakReference<>(context);
    }
    @Override
    protected MyPojo doInBackground(String... params) {
        // Some long running task
    }
    @Override
    protected void onPostExecute(MyPojo data) {
        MyActivity activity = activityReference.get();
        activity.progressBar.setVisibility(View.GONE);
        populateData(activity, data);
    }
}
```

Answer Score: 128

Stack Overflow answer, Verb Count: 20 , **Adjective Count:** 40

Stack Overflow Answer: `private WeakReference<MyActivity> activityReference;`
 Good riddance that it's deprecated, because the `WeakReference<Context>` was always a hack, and not a proper solution. Now people will have the opportunity to sanitize their code.

`AsyncTask<String, Void, MyPojo>` Based on this code, `Progress` is actually not needed, and there is a `String` input + `MyPojo` output. This is actually quite easy to accomplish without any use of `AsyncTask`.

```
public class TaskRunner {
    private final Executor executor = Executors.newSingleThreadExecutor(); // change according to your requirements
    private final Handler handler = new Handler(Looper.getMainLooper());

    public interface Callback<R> {
        void onComplete(R result);
    }

    public <R> void executeAsync(Callable<R> callable, Callback<R> callback) {
        executor.execute(() -> {
            final R result = callable.call();
            handler.post(() -> { callback.onComplete(result); });
        });
    }

    // How to pass in the String? Like so:
    class LongRunningTask implements Callable<MyPojo> {
        private final String input;

        public LongRunningTask(String input) {
            this.input = input;
        }

        @Override
        public MyPojo call() {
            // Some long running task
            return myPojo;
        }
    }

    // And // in ViewModel
    taskRunner.executeAsync(new LongRunningTask(input), (data) -> {
        // MyActivity activity = activityReference.get(); //
        activity.progressBar.setVisibility(View.GONE); // populateData(activity, data);
        loadingLiveData.setValue(false);
        dataLiveData.setValue(data);
    });

    // in Activity
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_activity);
        viewModel = ViewModelProviders.of(this).get(MyViewModel.class);
        viewModel.loadingLiveData.observe(this, (loading) -> {
            if (loading) {
                progressBar.setVisibility(View.VISIBLE);
            } else {
                progressBar.setVisibility(View.GONE);
            }
        });
        viewModel.dataLiveData.observe(this, (data) -> {
            populateData(data);
        });
    }
}
```

This example used a single-threaded pool which is good for DB writes (or serialized network requests), but if you want something for DB reads or multiple requests, you can consider the following Executor configuration:

```
private static final Executor THREAD_POOL_EXECUTOR = new
ThreadPoolExecutor(5, 128, 1, TimeUnit.SECONDS, new
LinkedBlockingQueue<Runnable>());
```

GPT answer, Verb Count: 18 , **Adjective Count:** 34

GPT-4o Answer: To replace `AsyncTask` with `java.util.concurrent` for `minSdkVersion 16`,

you can use [ExecutorService](#) along with [Callable](#) and [Future](#). Below is the equivalent code for your [LongRunningTask](#):

```
private static class LongRunningTask { private static
final String TAG = MyActivity.LongRunningTask.class.getSimpleName(); private
WeakReference activityReference; private ExecutorService executorService;
private Handler mainHandler; LongRunningTask(MyActivity context) {
activityReference = new WeakReference<>(context); executorService =
Executors.newSingleThreadExecutor(); mainHandler = new
Handler(Looper.getMainLooper()); } void execute(String... params) {
executorService.submit(() -> { MyPojo result = doInBackground(params); //
Post result to the main thread mainHandler.post(() ->
onPostExecute(result)); }); } private MyPojo doInBackground(String...
params) { // Some long running task return new MyPojo(); // Replace with
actual implementation } private void onPostExecute(MyPojo data) { MyActivity
activity = activityReference.get(); if (activity != null) {
activity.progressBar.setVisibility(View.GONE); populateData(activity, data);
} } void shutdown() { executorService.shutdown(); } } Usage: LongRunningTask
task = new LongRunningTask(this); task.execute("param1", "param2"); Notes: -
The Handler ensures that onPostExecute runs on the main thread. - Use shutdown() to properly
shut down the ExecutorService when it's no longer needed. - This approach maintains the
functionality of AsyncTask while adhering to modern recommendations.
```

Co-Pilot, Verb Count: 18 , Adjective Count: 29

Co-Pilot Answer: To replace the deprecated [AsyncTask](#) with [java.util.concurrent](#), you can use an [ExecutorService](#) for executing background tasks. Here's how you can rewrite the [LongRunningTask](#) class using [ExecutorService](#) and a [Handler](#) for updating the UI:

```
import
java.lang.ref.WeakReference; import java.util.concurrent.Callable; import
java.util.concurrent.ExecutionException; import
java.util.concurrent.ExecutorService; import java.util.concurrent.Executors;
import java.util.concurrent.Future; import android.os.Handler; import
android.os.Looper; private static class LongRunningTask { private static
final String TAG = MyActivity.LongRunningTask.class.getSimpleName(); private
WeakReference<MyActivity> activityReference; private ExecutorService
executor; private Handler handler; LongRunningTask(MyActivity context) {
activityReference = new WeakReference<>(context); executor =
Executors.newSingleThreadExecutor(); handler = new
Handler(Looper.getMainLooper()); } void execute(String... params) {
executor.submit(new Callable<MyPojo>() { @Override public MyPojo call()
throws Exception { return doInBackground(params); } }).addListener(() -> {
try { MyPojo result = future.get(); handler.post(() ->
onPostExecute(result)); } catch (InterruptedException | ExecutionException
e) { e.printStackTrace(); } }, executor); } private MyPojo
doInBackground(String... params) { // Some long running task return null; }
private void onPostExecute(MyPojo data) { MyActivity activity =
activityReference.get(); if (activity != null) {
activity.progressBar.setVisibility(View.GONE); populateData(activity, data);
} } } This implementation uses an ExecutorService to handle background tasks and a Handler
to update the UI on the main thread. It achieves the same functionality as the original AsyncTask while
being compatible with minSdkVersion 16.
```

Question 2:

Tags: javascript, typescript

Question: Here's the code: `const test = Array.from(document.getElementsByClassName('mat-form-field-infix')); test.forEach((element) => { element.outerHTML = '<div class="good-day-today" style="width: 0px;"></div>'; // Please note that this line works fine! element.style.padding = '10px'; element.style.borderTop = '0'; });` Error I get when compiled: ERROR in src/app//.component.ts(101,21): error TS2339: Property 'style' does not exist on type 'Element'. src/app//.component.ts(102,21): error TS2339: Property 'style' does not exist on type 'Element'. How can I fix it? I tried to remove the `Array.from...` part, tried to use `for of` and `for in`, tried `as any`, but above is the way I have to do it.

Answer Score: 172

Stack Overflow answer, Verb Count: 8 , Adjective Count: 10

Stack Overflow Answer: You need a typecast:

`Array.from(document.getElementsByClassName('mat-form-field-infix') as HTMLCollectionOf<HTMLElement>)` That's because `getElementsByClassName` only returns `HTMLCollection<Element>`, and `Element` does not have a `style` property. The `HTMLElement` however does implement it via its `ElementCSSInlineStyle` extended interface. Note that this typecast is typesafe in the way that every `Element` is either a `HTMLElement` or an `SVGElement`, and I hope that your SVG Elements don't have a class.

GPT answer, Verb Count: 15 , Adjective Count: 11

GPT-4o Answer: To fix the error, you need to explicitly cast the elements retrieved from `document.getElementsByClassName` to `HTMLElement`, as the `Element` type does not include the `style` property. Here's how you can update your code: `const test = Array.from(document.getElementsByClassName('mat-form-field-infix')) as HTMLElement[]; test.forEach((element) => { element.outerHTML = '<div class="good-day-today" style="width: 0px;"></div>'; element.style.padding = '10px'; element.style.borderTop = '0'; });` Explanation: - The `Array.from` creates an array from the `HTMLCollection` returned by `document.getElementsByClassName`. - The `as HTMLElement[]` ensures TypeScript treats the elements as `HTMLElement`, which has the `style` property. - This resolves the TypeScript errors related to the `style` property.

Co-Pilot, Verb Count: 14 , Adjective Count: 11

Co-Pilot Answer: The error you're encountering is due to TypeScript being strict about the types. The `document.getElementsByClassName` method returns an `HTMLCollection` of `Element` objects, which do not have a `style` property. To fix this, you can cast the elements to `HTMLElement`, which does have a `style` property. `const test = Array.from(document.getElementsByClassName('mat-form-field-infix')) as HTMLElement[]; test.forEach((element) => { element.outerHTML = '<div class="good-day-today" style="width: 0px;"></div>'; // Please note that this line works fine! element.style.padding = '10px'; element.style.borderTop = '0'; });` By casting the elements to `HTMLElement`, TypeScript recognizes that the `style` property exists on these elements, resolving the error.

Question 3:

Tags: python, machine-learning, python-3.x, deep-learning, pytorch

Question: This: `device = torch.device("cuda" if torch.cuda.is_available() else "cpu") model.to(device) for data in dataloader: inputs, labels = data outputs = model(inputs)` Gives the error: `RuntimeError: Input type (torch.FloatTensor) and weight type (torch.cuda.FloatTensor) should be the same`

Answer Score: 315

Stack Overflow answer, Verb Count: 11 , **Adjective Count:** 10

Stack Overflow Answer: You get this error because your model is on the GPU, but your data is on the CPU. So, you need to send your input tensors to the GPU. `inputs, labels = data # this is what you had inputs, labels = inputs.cuda(), labels.cuda() # add this line` Or like this, to stay consistent with the rest of your code: `device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu") inputs, labels = inputs.to(device), labels.to(device)` The same error will be raised if your input tensors are on the GPU but your model weights aren't. In this case, you need to send your model weights to the GPU. `model = MyModel() if torch.cuda.is_available(): model.cuda()` See the documentation for `cuda()`, and its opposite, `cpu()`.

GPT answer, Verb Count: 12 , **Adjective Count:** 9

GPT-4o Answer: The error occurs because the model is moved to the GPU using `model.to(device)`, but the inputs are still on the CPU. You need to ensure that the inputs are also moved to the same device as the model. Here's how you can fix it: `device = torch.device("cuda" if torch.cuda.is_available() else "cpu") model.to(device) for data in dataloader: inputs, labels = data inputs, labels = inputs.to(device), labels.to(device) # Move inputs and labels to the same device outputs = model(inputs)` Explanation: - `inputs.to(device)` moves the input tensors to the GPU or CPU, depending on the device being used. - Similarly, `labels.to(device)` ensures the labels are on the same device as the model for any subsequent operations. - This ensures compatibility between the input data and the model's weights.

Co-Pilot, Verb Count: 8 , **Adjective Count:** 5

Co-Pilot Answer: This error occurs because the model has been moved to the GPU using `model.to(device)`, but the input data is still on the CPU. You need to move the input data to the same device as the model. Here is the corrected code: `device = torch.device("cuda" if torch.cuda.is_available() else "cpu") model.to(device) for data in dataloader: inputs, labels = data inputs, labels = inputs.to(device), labels.to(device) outputs = model(inputs)` By moving both the inputs and labels to the same device as the model, the error will be resolved.

Question 4:

Tags: python, sublimetext3, windows-10

Question: Today when I tried to run simple code on Sublime Text 3, the following message appeared: Python was not found but can be installed from the Microsoft Store: <https://go.microsoft.com/fwlink?linkID=2082640> And when I type Python in CMD, it opens the Windows Store for me to download Python 3.7. This problem started today for no good reason. I didn't change or download anything about Python and already tried reinstalling Python, and the Path environment variable is correct.

Answer Score: 765

Stack Overflow answer, Verb Count: 76 , **Adjective Count:** 26

Stack Overflow Answer: Use the Windows search bar to find "Manage app execution aliases". There should be two aliases for Python. Unselect them, and this will allow the usual Python aliases "python" and "python3". See the image below. I think we have this problem when installing Python because in a new Windows installation the aliases are in the ON position as in image below. When turned on, Windows puts an empty or fake file named `python.exe` and `python3.exe` in the directory named `%USERPROFILE%\AppData\Local\Microsoft\WindowsApps`. This is the alias. Then Microsoft put that

directory at the top of the list in the "Path" environment variables. When you enter "python" in cmd, it searches the directories listed in your "Path" environment variables page from top to bottom. So if you installed Python after a new Windows 10 install then get redirected to the Windows Store, it's because there are two python.exe's: The alias in the App Execution Alias page, and the real one wherever you installed Python. But cmd finds the App execution, alias python.exe, first because that directory is at the top of the Path. I think the easiest solution is to just check the python.exe and python3.exe to OFF as I suggested before, which deletes the fake EXE file files. The first time I ran into this problem, I manually deleted the python.exe and python3.exe files but when I restarted the files regenerated. That prompted me to search for the App Execution Aliases page and uncheck the box, which solved it for me, by not allowing the files to regenerate. Based on this Microsoft Devblog, they stated they created this system partially for new Python users, specifically kids learning Python in school that had trouble installing it, and focus on learning to code. I think Windows probably deletes those aliases if you install Python from the Windows App Store. We are noticing that they do not get deleted if you manually install from another source. (Also, the empty/fake python.exe is not really empty. It says 0 KB in the screenshot, but entering "start ms-windows-store:" in cmd opens the Windows App Store, so it probably just has a line with that and a way to direct it to the Python page.) One alternative, as Chipjust suggested, you can create a new alias for Python using something like DOSKEY as explained in this article for example: How to set aliases for the command prompt in Windows Another alternative is to delete the user path environment variable that points to the alias files, %USERPROFILE%\AppData\Local\Microsoft\WindowsApps, but the App Execution Aliases handle more apps than just python, and deleting the path from environment variables breaks all the other apps that have execution aliases in that directory; which on my PC includes notepad, xbox game bar, spotify, monitoring software for my motherboard, paint, windows subsystem for android, to name a few. Also if you think about it, the average Windows user is unfamiliar editing environment variables and on school and business owned computers requires administrative access. So deleting the path to ...\\WindowsApps, from the path environment variable, is not ideal.

GPT answer, Verb Count: 31 , Adjective Count: 11

GPT-4o Answer: The issue occurs because Windows is defaulting to the Microsoft Store for Python when it doesn't recognize a valid Python installation. To fix this issue, follow these steps: 1. [Ensure Python is installed:](#) - Reinstall Python from the official website: <https://www.python.org>. - During installation, make sure to check the box for "Add Python to PATH". 2. [Verify the Path environment variable:](#) - Go to "System Properties" -> "Environment Variables". - Under "System Variables", find the variable named "Path". - Ensure that the Python installation paths are added: - C:\\PythonXX\\ (Replace XX with your version, e.g., 39 for Python 3.9). - C:\\PythonXX\\Scripts\\. 3. [Prioritize the correct Python executable:](#) - Open Command Prompt. - Type where python. - If multiple paths appear, ensure the correct Python installation is listed first. - If the wrong path is listed, edit the Path variable to move the correct Python path to the top. 4. [Test Python:](#) - Restart the Command Prompt or Sublime Text. - Run python --version to confirm that the correct Python version is being used. These steps should resolve the issue and prevent Windows from redirecting to the Microsoft Store.

Co-Pilot, Verb Count: 32 , Adjective Count: 14

Co-Pilot Answer: It sounds like Sublime Text 3 is trying to use the stub version of Python that comes with Windows, which redirects you to the Microsoft Store - Sublime Forum](<https://forum.sublimetext.com/t/cant-get-editor-to-recognize-python-windows/55461>). To fix this, you need to ensure Sublime Text is using the correct Python interpreter. Here's how you can do it: 1. [Open Sublime Text 3 - Sublime Forum\]\(https://forum.sublimetext.com/t/cant-get-editor-to-recognize-python-windows/55461\)](#). 2. Go to Tools > Build System > New Build System - Sublime Forum](<https://forum.sublimetext.com/t/cant-get-editor-to-recognize-python-windows/55461>). 3. In the new file that

opens, add the following code: { "cmd": ["C:\\Users\\YourUsername\\AppData\\Local\\Programs\\Python\\Python37\\python.exe", "-u", "\$file"], "file_regex": "^[]*File \"(...?)\"\\", line ([0-9]*)", "selector": "source.python" } 4. Save the file as Python.sublime-build - Sublime Forum](<https://forum.sublimetext.com/t/cant-get-editor-to-recognize-python-windows/55461>). 5. Go to Tools > Build System and select Python - Sublime Forum](<https://forum.sublimetext.com/t/cant-get-editor-to-recognize-python-windows/55461>). Replace YourUsername with your actual username and adjust the path to the correct Python interpreter if it's installed in a different location. This should resolve the issue and allow Sublime Text to use the correct Python interpreter.

Question 5:

Tags: javascript, typescript, optional-chaining

Question: TypeScript 3.7 now supports the optional chaining operator. Hence, you can write code such as: `const value = a?.b?.c;` I.e., you can use this operator to access properties of an object, where the object itself may be `null` or `undefined`. Now what I would like to do is basically the same, but the property names are dynamic: `const value = a[b]?.c;` However, there I get a syntax error: error TS1005: ':' expected. What am I doing wrong here? Is this even possible? The proposal seems to imply that this is not possible (but maybe I get the syntax examples wrong).

Answer Score: 317

Stack Overflow answer, Verb Count: 11 , **Adjective Count:** 10

Stack Overflow Answer: When accessing a property using bracket notation and optional chaining, you need to use a dot in addition to the brackets: `const value = a?.[b]?.c;` This is the syntax that was adopted by the TC39 proposal, because otherwise it's hard for the parser to figure out if this `?` is part of a ternary expression or part of optional chaining. The way I think about it: the symbol for optional chaining isn't `?`, it's `?.` If you're doing optional chaining, you'll always be using both characters.

GPT answer, Verb Count: 17 , **Adjective Count:** 18

GPT-4o Answer: The optional chaining operator `?.` does not support dynamic property access using bracket notation directly. However, you can achieve the same functionality by combining the optional chaining operator with the bracket notation. You need to ensure the object exists before attempting to access the property dynamically. Here's how you can do it: `const value = a?.[b]?.c;`
Explanation: - `a?.[b]` checks if `a` is not `null` or `undefined`, and if so, accesses the property dynamically using `b`. - `?.c` then safely accesses the property `c` of the result from `a?.[b]`, ensuring no error occurs if the intermediate value is `null` or `undefined`. This syntax is valid in TypeScript 3.7 and later.

Co-Pilot, Verb Count: 10 , **Adjective Count:** 6

Co-Pilot Answer: The issue you're encountering is because TypeScript does not support optional chaining with dynamic property names directly. You need to use the bracket notation to achieve this with a combination of optional chaining and the `&&` (AND) operator. Here's how you can do it: `const value = a?.[b] && a[b]?.c;` This way, you can dynamically access properties while handling cases where `a` or `a[b]` might be `null` or `undefined`.

Month: Dec-19

Question 1:

Tags: javascript, jsx, reactjs

Question: I am generating a `dl` in React:

```
<dl> { highlights.map(highlight => { const count = text.split(highlight).length - 1; return ( <> <dt key={`dt-${highlight.id}`}>{highlight}</dt> <dd key={`dd-${highlight.id}`}>{count}</dd> </> ); }) } </dl>
```

 This gives me the warning: Warning: Each child in a list should have a unique "key" prop. This will remove the warning, but doesn't generate the HTML I want:

```
<dl> { highlights.map(highlight => { const count = text.split(highlight).length - 1; return ( <div key={highlight.id}> <dt>{highlight}</dt> <dd>{count}</dd> </div> ); }) } </dl>
```

 And I cannot add a `key` prop to a fragment (`<> </>`). How can work around this? I am using React 16.12.0.

Answer Score: 536

Stack Overflow answer, Verb Count: 4 , Adjective Count: 3

Stack Overflow Answer: To add a key to a fragment you need to use full Fragment syntax:

```
<React.Fragment key={your key}> ... </React.Fragment>
```

 See docs here <https://reactjs.org/docs/fragments.html#keyed-fragments>

GPT answer, Verb Count: 14 , Adjective Count: 10

GPT-4o Answer: The issue arises because React fragments do not support the `key` prop when using the shorthand syntax (`<>`). To resolve this and keep the desired HTML structure, you can use the long-form `React.Fragment` syntax, which supports the `key` prop. Here's how you can fix the issue:

```
<dl> { highlights.map(highlight => { const count = text.split(highlight).length - 1; return ( <React.Fragment key={highlight.id}> <dt>{highlight}</dt> <dd>{count}</dd> </React.Fragment> ); }) } </dl>
```

 This approach allows you to maintain the correct `dl` structure while providing a unique `key` for each fragment, thus removing the warning.

Co-Pilot, Verb Count: 16 , Adjective Count: 15

Co-Pilot Answer: The warning you're seeing is due to the fact that React requires a unique `key` prop for each child in a list, and you can't add a `key` prop directly to a fragment. To work around this, you can use a wrapper component that will allow you to assign a unique `key` to each set of `<dt>` and `<dd>` elements. Here's an example of how you can do this:

```
const DtDdWrapper = ({ id, highlight, count }) => ( <> <dt key={`dt-${id}`}>{highlight}</dt> <dd key={`dd-${id}`}>{count}</dd> </> );
```

 Then use this wrapper component inside your `map` function:

```
<dl> { highlights.map(highlight => { const count = text.split(highlight).length - 1; return ( <DtDdWrapper key={highlight.id} id={highlight.id} highlight={highlight} count={count} /> ); }) } </dl>
```

 This will ensure that each set of `<dt>` and `<dd>` elements has a unique `key` while preserving the desired HTML structure.

Question 2:

Tags: python, version, conda, virtual-environment

Question: I created a conda environment with Python version 3.8, but it doesn't support matplotlib... So I am looking for something like this to change the Python version: `conda env my_env update to python=3.6`. Is this possible or do I need to recreate the environment? I have miniconda installed.

Answer Score: 302

Stack Overflow answer, Verb Count: 3 , Adjective Count: 2

Stack Overflow Answer: Activate the relevant environment, then install your target python version.

```
conda activate my_env conda install python=3.6
```

GPT answer, Verb Count: 16 , Adjective Count: 5

GPT-4o Answer: You can change the Python version in your existing conda environment without recreating it by using the following command: `conda install python=3.6` This will update the Python version in your environment to 3.6 while retaining the rest of your installed packages. Make sure to activate the environment first: `conda activate my_env` `conda install python=3.6` If there are compatibility issues with existing packages, you may need to recreate the environment with the desired Python version: `conda create -n my_env python=3.6`

Co-Pilot, Verb Count: 10 , Adjective Count: 4

Co-Pilot Answer: You can update the Python version of an existing conda environment using the following command: `conda install python=3.6` Activate your conda environment first: `conda activate my_env` Then run the install command: `conda install python=3.6` This will update the Python version in your current environment to 3.6 without needing to recreate it.

Question 3:

Tags: python, object-detection, low-memory, deep-learning, pytorch

Question: I think it's a pretty common message for PyTorch users with low GPU memory:

`RuntimeError: CUDA out of memory. Tried to allocate X MiB (GPU X; X GiB total capacity; X GiB already allocated; X MiB free; X cached)` I tried to process an image by loading each layer to GPU and then loading it back: `for m in self.children(): m.cuda() x = m(x) m.cpu() torch.cuda.empty_cache()` But it doesn't seem to be very effective. I'm wondering is there any tips and tricks to train large deep learning models while using little GPU memory.

Answer Score: 130

Stack Overflow answer, Verb Count: 30 , Adjective Count: 16

Stack Overflow Answer: Although `import torch torch.cuda.empty_cache()` provides a good alternative for clearing the occupied cuda memory and we can also manually clear the not in use variables by using, `import gc del variables gc.collect()` But still after using these commands, the error might appear again because pytorch doesn't actually clears the memory instead clears the reference to the memory occupied by the variables. So reducing the batch_size after restarting the kernel and finding the optimum batch_size is the best possible option (but sometimes not a very feasible one). Another way to get a deeper insight into the allocation of memory in gpu is to use: `torch.cuda.memory_summary(device=None, abbreviated=False)` wherein, both the arguments are optional. This gives a readable summary of memory allocation and allows you to figure the reason of CUDA running out of memory and restart the kernel to avoid the error from happening again (Just like I did in my case). Passing the data iteratively might help but changing the size of layers of your network or breaking them down would also prove effective (as sometimes the model also occupies a significant memory for example, while doing transfer learning).

GPT answer, Verb Count: 32 , Adjective Count: 25

GPT-4o Answer: To manage GPU memory effectively and avoid CUDA out-of-memory errors, here are some tips and tricks: 1. Use Gradient Accumulation Split your batch into smaller sub-batches and accumulate gradients across iterations before updating weights. This reduces the memory required for each forward and backward pass. `accumulation_steps = 4 # Number of sub-batches optimizer.zero_grad() for i in range(accumulation_steps): outputs = model(inputs[i]) loss = criterion(outputs, targets[i]) loss.backward() # Accumulate gradients if (i + 1) % accumulation_steps == 0: optimizer.step() optimizer.zero_grad()` 2. Use Mixed Precision Training Mixed precision reduces memory usage by storing some tensors in 16-bit floating point (FP16) instead of 32-bit (FP32). You can use NVIDIA's

`torch.cuda.amp` for this: `scaler = torch.cuda.amp.GradScaler()` for inputs, targets in dataloader: `with torch.cuda.amp.autocast(): outputs = model(inputs) loss = criterion(outputs, targets)` `scaler.scale(loss).backward()` `scaler.step(optimizer)` `scaler.update()` 3. Clear Unused Variables Explicitly delete intermediate variables and use `torch.cuda.empty_cache()` to free memory: `del intermediate_tensor torch.cuda.empty_cache()` 4. Model Sharding Move parts of the model to the GPU only when needed, like in your example. However, make sure to manage the tensor locations carefully: `for m in model.children(): m.cuda()` `x = m(x.cuda()) m.cpu()` `torch.cuda.empty_cache()` 5. Reduce Batch Size Lowering the batch size directly reduces the memory footprint. 6. Use Checkpointing Save intermediate activations during the forward pass and recompute them during the backward pass to save memory: `from torch.utils.checkpoint import checkpoint` `outputs = checkpoint(model.layer, inputs)` 7. Optimize Data Loading Use `pin_memory=True` in DataLoader and `non_blocking=True` during data transfer to GPU: `for inputs, targets in dataloader: inputs, targets = inputs.to(device, non_blocking=True), targets.to(device, non_blocking=True)` 8. Use Smaller Model Sizes If feasible, try using model pruning or distillation to reduce the model size.

Co-Pilot, Verb Count: 24 , Adjective Count: 22

Co-Pilot Answer: When dealing with CUDA out of memory errors in PyTorch, here are some tips and tricks you can use to manage GPU memory more effectively: 1. Gradient Accumulation: Accumulate gradients over multiple mini-batches to simulate a larger batch size without requiring more memory. `for i, data in enumerate(dataloader): inputs, labels = data outputs = model(inputs) loss = criterion(outputs, labels) loss = loss / accumulation_steps loss.backward()` `if (i+1) % accumulation_steps == 0: optimizer.step() optimizer.zero_grad()` 2. Mixed Precision Training: Use the `torch.cuda.amp` module to perform mixed precision training, which can reduce memory usage and speed up computations. `from torch.cuda.amp import autocast, GradScaler` `scaler = GradScaler()` for data in dataloader: `inputs, labels = data with autocast(): outputs = model(inputs) loss = criterion(outputs, labels)` `scaler.scale(loss).backward()` `scaler.step(optimizer)` `scaler.update()` `optimizer.zero_grad()` 3. Chunked Data Loading: Load and process data in smaller chunks to fit within GPU memory constraints. `for chunk in get_data_in_chunks(dataloader): inputs, labels = chunk outputs = model(inputs)` # Continue with training steps 4. Freeing Unused Memory: Use `torch.cuda.empty_cache()` to free up unused memory and avoid out of memory errors. `torch.cuda.empty_cache()` 5. Offloading Layers: Move individual layers to CPU after their forward and backward passes to save memory. `for m in self.children(): m.cuda()` `x = m(x) m.cpu()` `torch.cuda.empty_cache()` By implementing these strategies, you should be able to manage GPU memory more effectively while training large deep learning models.

Question 4:

Tags: python, macos, mercurial, homebrew, openssl

Question: When trying to use any `hg` Mercurial commands on the console, I keep getting this error. I installed Python using Homebrew and I am running Mac OS Catalina v. 10.15.1. Any reference would be appreciated. Here is the error I'm getting: `hg commit --amend ERROR:root:code for hash md5 was not found. Traceback (most recent call last): File "/usr/local/Cellar/python@2/2.7.15_3/Frameworks/Python.framework/Versions/2.7/lib/python2.7/hashlib.py", line 147, in <module> globals()[__func_name] = __get_hash(__func_name) File "/usr/local/Cellar/python@2/2.7.15_3/Frameworks/Python.framework/Versions/2.7/lib/python2.7/hashlib.py", line 97, in __get_builtin_constructor raise ValueError('unsupported hash type ' + name)`

```
ValueError: unsupported hash type md5 ERROR:root:code for hash sha1 was not
found. Traceback (most recent call last): File "/usr/local/Cellar/python@2/2
.7.15_3/Frameworks/Python.framework/Versions/2.7/lib/python2.7/hashlib.py",
line 147, in <module> globals()[__func_name] = __get_hash(__func_name) File
"/usr/local/Cellar/python@2/2.7.15_3/Frameworks/Python.framework/Versions/2.
7/lib/python2.7/hashlib.py", line 97, in __get_builtin_constructor raise
ValueError('unsupported hash type ' + name) ValueError: unsupported hash
type sha1 ERROR:root:code for hash sha224 was not found. Traceback (most
recent call last): File "/usr/local/Cellar/python@2/2.7.15_3/Frameworks/Pyth
on.framework/Versions/2.7/lib/python2.7/hashlib.py", line 147, in <module>
globals()[__func_name] = __get_hash(__func_name) File "/usr/local/Cellar/pyt
hon@2/2.7.15_3/Frameworks/Python.framework/Versions/2.7/lib/python2.7/hashli
b.py", line 97, in __get_builtin_constructor raise ValueError('unsupported
hash type ' + name) ValueError: unsupported hash type sha224 ERROR:root:code
for hash sha256 was not found. Traceback (most recent call last): File "/usr
/local/Cellar/python@2/2.7.15_3/Frameworks/Python.framework/Versions/2.7/lib
/python2.7/hashlib.py", line 147, in <module> globals()[__func_name] =
__get_hash(__func_name) File "/usr/local/Cellar/python@2/2.7.15_3/Frameworks
/Python.framework/Versions/2.7/lib/python2.7/hashlib.py", line 97, in
__get_builtin_constructor raise ValueError('unsupported hash type ' + name)
ValueError: unsupported hash type sha256 ERROR:root:code for hash sha384 was
not found. Traceback (most recent call last): File "/usr/local/Cellar/python
@2/2.7.15_3/Frameworks/Python.framework/Versions/2.7/lib/python2.7/hashlib.p
y", line 147, in <module> globals()[__func_name] = __get_hash(__func_name)
File "/usr/local/Cellar/python@2/2.7.15_3/Frameworks/Python.framework/Versio
ns/2.7/lib/python2.7/hashlib.py", line 97, in __get_builtin_constructor
raise ValueError('unsupported hash type ' + name) ValueError: unsupported
hash type sha384 ERROR:root:code for hash sha512 was not found. Traceback
(most recent call last): File "/usr/local/Cellar/python@2/2.7.15_3/Framework
s/Python.framework/Versions/2.7/lib/python2.7/hashlib.py", line 147, in
<module> globals()[__func_name] = __get_hash(__func_name) File "/usr/local/C
ellar/python@2/2.7.15_3/Frameworks/Python.framework/Versions/2.7/lib/python2
.7/hashlib.py", line 97, in __get_builtin_constructor raise
ValueError('unsupported hash type ' + name) ValueError: unsupported hash
type sha512 Traceback (most recent call last): File "/usr/local/bin/hg",
line 43, in <module> dispatch.run() File "/usr/local/Cellar/mercurial/4.9/li
b/python2.7/site-packages/hgdemandimport/demandimportpy2.py", line 150, in
__getattr__ self._load() File "/usr/local/Cellar/mercurial/4.9/lib/python2.7
/site-packages/hgdemandimport/demandimportpy2.py", line 94, in _load
_origimport, head, globals, locals, None, level) File "/usr/local/Cellar/mer
curial/4.9/lib/python2.7/site-packages/hgdemandimport/demandimportpy2.py",
line 43, in _hgextimport return importfunc(name, globals, *args, **kwargs)
File "/usr/local/Cellar/mercurial/4.9/lib/python2.7/site-packages/mercurial/
dispatch.py", line 625, in <module> class lazyaliasentry(object): File "/usr
/local/Cellar/mercurial/4.9/lib/python2.7/site-packages/mercurial/dispatch.p
y", line 636, in lazyaliasentry @util.propertycache File "/usr/local/Cellar/
mercurial/4.9/lib/python2.7/site-packages/hgdemandimport/demandimportpy2.py"
, line 150, in __getattr__ self._load() File "/usr/local/Cellar/mercurial/4.
9/lib/python2.7/site-packages/hgdemandimport/demandimportpy2.py", line 94,
in _load _origimport, head, globals, locals, None, level) File "/usr/local/C
ellar/mercurial/4.9/lib/python2.7/site-packages/hgdemandimport/demandimportp
y2.py", line 43, in _hgextimport return importfunc(name, globals, *args,
**kwargs) File "/usr/local/Cellar/mercurial/4.9/lib/python2.7/site-packages/
```

```
mercurial/util.py", line 180, in <module> 'md5': hashlib.md5, File "/usr/local/Cellar/mercurial/4.9/lib/python2.7/site-packages/hgdemandimport/demandimportpy2.py", line 151, in __getattr__ return getattr(self._module, attr)
AttributeError: 'module' object has no attribute 'md5' I also tried following the instruction on this issue but none of the solutions seem to work brew link openssl --force
Warning: Refusing to link macOS-provided software: openssl@1.1 If you need to have openssl@1.1 first in your PATH run: echo 'export PATH="/usr/local/opt/openssl@1.1/bin:$PATH"' >> ~/.zshrc For compilers to find openssl@1.1 you may need to set: export LDFLAGS="-L/usr/local/opt/openssl@1.1/lib" export CPPFLAGS="-I/usr/local/opt/openssl@1.1/include" For pkg-config to find openssl@1.1 you may need to set: export PKG_CONFIG_PATH="/usr/local/opt/openssl@1.1/lib/pkgconfig"
```

Answer Score: 713

Stack Overflow answer, Verb Count: 19 , **Adjective Count:** 13

Stack Overflow Answer: Running `brew reinstall python@2` didn't work for my existing Python 2.7 virtual environments. Inside them there were still `ERROR:root:code for hash sha1 was not found` errors. I encountered this problem after I ran `brew upgrade openssl`. And here's the fix: `$ ls /usr/local/Cellar/openssl ...which shows 1.0.2t` According to the existing version, run: `$ brew switch openssl 1.0.2t ...which shows Cleaning /usr/local/Cellar/openssl/1.0.2t Opt link created for /usr/local/Cellar/openssl/1.0.2t` After that, run the following command in a Python 2.7 virtualenv: `(my-venv) $ python -c "import hashlib;m=hashlib.md5();print(m.hexdigest())"` ...which shows `d41d8cd98f00b204e9800998ecf8427e` No more errors.

GPT answer, Verb Count: 43 , **Adjective Count:** 19

GPT-4o Answer: The error indicates that the required hash functions (e.g., MD5, SHA1) are not found because the Python installation cannot access the necessary OpenSSL libraries. Here's how you can resolve this issue: 1. Relink OpenSSL Ensure OpenSSL is properly linked to your Python installation. Run the following commands to fix the linkage: `brew install openssl brew link --force openssl` If the linking fails, you might need to export the paths manually: `export LDFLAGS="-L/usr/local/opt/openssl/lib" export CPPFLAGS="-I/usr/local/opt/openssl/include" export PKG_CONFIG_PATH="/usr/local/opt/openssl/lib/pkgconfig"` 2. Reinstall Python with OpenSSL Uninstall and reinstall Python using Homebrew to ensure it is built with OpenSSL support: `brew reinstall python@2 --with-openssl` 3. Rebuild Python Hashlib If the issue persists, rebuild Python's hashlib module to ensure it can find the OpenSSL functions: `brew reinstall python@2` 4. Verify OpenSSL Integration Run the following commands to verify that Python has access to OpenSSL: `python -c "import ssl; print(ssl.OPENSSL_VERSION)"` If OpenSSL is correctly installed and linked, it will display the OpenSSL version. 5. Use Python 3 Python 2 is deprecated, and support for it has ended. Consider upgrading your project to Python 3 to avoid compatibility issues: `brew install python python3 -m venv venv source venv/bin/activate` After performing these steps, retry the `hg` command. The issue should be resolved.

Co-Pilot, Verb Count: 20 , **Adjective Count:** 11

Co-Pilot Answer: To resolve the "code for hash md5 was not found" error in Mercurial on macOS Catalina, you can try the following steps: 1. Upgrade Mercurial: Ensure you are using the latest version of Mercurial, as newer versions have better compatibility with Python 3. 2. Install pycryptodome: Install the pycryptodome library, which provides support for various cryptographic algorithms including MD5.

`pip install pycryptodome` 3. Modify Mercurial Configuration: Change the hashing algorithm used by Mercurial to SHA-1 by adding the following line to your Mercurial configuration file (`~/.hgrc` or `%USERPROFILE%\hgrc`): `[ui] hash = sha1` These steps should help resolve the error you're encountering.

Question 5:

Tags: javascript, typescript, vue.js, vuejs3, vue-composition-api

Question: While Vue Composition API RFC Reference site has many advanced use scenarios with the `watch` module, there are no examples of how to watch component props. Neither is it mentioned in Vue Composition API RFC's main page or `vuejs/composition-api` in Github. I've created a Codesandbox to elaborate on this issue. `<template> <div id="app">
 <p>Prop watch demo with select input using v-model:</p> <PropWatchDemo :selected="testValue"/> </div> </template>` `<script> import { createComponent, onMounted, ref } from "@vue/composition-api"; import PropWatchDemo from "../components/PropWatchDemo.vue"; export default createComponent({ name: "App", components: { PropWatchDemo }, setup: (props, context) => { const testValue = ref("initial"); onMounted(() => { setTimeout(() => { console.log("Changing input prop value after 3s delay"); testValue.value = "changed"; // This value change does not trigger watchers? }, 3000); }); return { testValue }; } }); </script> <template> <select v-model="selected"> <option value="null">null value</option> <option value>Empty value</option> </select> </template> <script> import { createComponent, watch } from "@vue/composition-api"; export default createComponent({ name: "MyInput", props: { selected: { type: [String, Number], required: true } }, setup(props) { console.log("Setup props:", props); watch((first, second) => { console.log("Watch function called with args:", first, second); // First arg function registerCleanup, second is undefined }); // watch(props, (first, second) => { // console.log("Watch props function called with args:", first, second); // // Logs error: // // Failed watching path: "[object Object]" Watcher only accepts simple // // dot-delimited paths. For full control, use a function instead. // }) watch(props.selected, (first, second) => { console.log("Watch props.selected function called with args:", first, second); // Both props are undefined so it's just a bare callback func to be run }); return {}; } }); </script>` Although my question and code example were initially with JavaScript, I'm using TypeScript. Tony Tom's first answer although working, led to a type error. Which was solved by Michal Lev's answer. So I've tagged this question with `typescript` afterward. Here is my polished yet barebones version of the reactive wirings for this custom select component, on top of `<b-form-select>` from `bootstrap-vue` (otherwise agnostic-implementation but this underlying component does emit `@input` and `@change` events both, based on whether the change was made programmatically or by user interaction). `<template> <b-form-select v-model="selected" :options="{ }" @input="handleSelection('input', $event)" @change="handleSelection('change', $event)" /> </template> <script lang="ts"> import { createComponent, SetupContext, Ref, ref, watch, computed, } from '@vue/composition-api'; interface Props { value?: string | number | boolean; } export default createComponent({ name: 'CustomSelect', props: { value: { type: [String, Number, Boolean], required: false, // Accepts null and undefined as well }, }, setup(props: Props, context: SetupContext) { // Create a Ref from prop, as two-way binding is allowed only with sync -modifier, // with passing prop in parent and explicitly emitting update event on child: // Ref:`

```

https://v2.vuejs.org/v2/guide/components-custom-events.html#sync-Modifier //
Ref: https://medium.com/@jithilmt/vue-js-2-two-way-data-binding-in-parent-and-child-components-1cd271c501ba
const selected: Ref<Props['value']> =
  ref(props.value); const handleSelection = function emitUpdate(type: 'input'
  | 'change', value: Props['value']) { // For sync -modifier where 'value' is
  the prop name context.emit('update:value', value); // For @input and/or
  @change event propagation // @input emitted by the select component when
  value changed <programmatically> // @change AND @input both emitted on <user
  interaction> context.emit(type, value); }; // Watch prop value change and
  assign to value 'selected' Ref watch(() => props.value, (newValue:
  Props['value']) => { selected.value = newValue; }); return { selected,
  handleSelection, }; }; </script>

```

Answer Score: 249

Stack Overflow answer, Verb Count: 33 , **Adjective Count:** 47

Stack Overflow Answer: If you take a look at `watch` typings here it makes it clear the first argument of `watch` could be an `array`, `function` or `Ref<T>` `props` passed to the `setup` function are a reactive object (likely by `readonly(reactive())`, its properties are getters. So what you are doing is passing the value of the getter as the 1st argument of `watch`, string "initial" in this case. Because Vue 2 `$watch` API is used under the hood (and same function exists in Vue 3), you are effectively trying to watch non-existent property with name "initial" on your component instance. Your callback was only called once. The reason it was called at least once is because the new `watch` API is behaving like the current `$watch` with the `immediate` option (UPDATE 03/03/2021 - this was later changed and in release version of Vue 3, `watch` is lazy same way as it was in Vue 2) So by accident you are doing the same thing Tony Tom suggested but with the wrong value. In both cases, it is not valid code when you are using TypeScript. You can do this instead: `watch(() => props.selected, (first, second) => { console.log("Watch props.selected function called with args:", first, second); });` Here the 1st function is executed immediately by Vue to collect dependencies (to know what should trigger the callback) and 2nd function is the callback itself. Other way would be to convert props object using `toRefs` so its properties would be of type `Ref<T>` and you can pass them as the first argument of `watch`. However, most of the time watching props is not needed. Simply use `props.xxx` directly in your template (or `setup`) and let Vue do the rest.

GPT answer, Verb Count: 22 , **Adjective Count:** 17

GPT-4o Answer: To watch component props with the Vue Composition API, you can use the `watch` function and pass a getter function to access the specific prop. Below is the corrected implementation: `import { createComponent, watch, ref } from "@vue/composition-api"; export default createComponent({ name: "MyInput", props: { selected: { type: [String, Number], required: true } }, setup(props) { // Watching the 'selected' prop watch(() => props.selected, (newValue, oldValue) => { console.log("Selected changed from", oldValue, "to", newValue); }); return { }; } });` Explanation: 1. The `watch` function accepts a getter function as its first argument. In this case, `() => props.selected` is used to access the specific `selected` prop. 2. The second argument is a callback function that receives the new and old values of the watched prop, allowing you to react to changes. This approach ensures the proper watching of props in a TypeScript-compatible and reactive manner using the Vue Composition API.

Co-Pilot, Verb Count: 15 , **Adjective Count:** 10

Co-Pilot Answer: To watch component props with the Vue Composition API, you can use the `watch` function in the `setup` function. Here's how you can watch the prop `selected`: `<template> <select v-model="selected"> <option value="null">null value</option> <option value>Empty value</option> </select> </template> <script> import {`

```
createComponent, watch, ref } from "@vue/composition-api"; export default
createComponent({ name: "MyInput", props: { selected: { type: [String,
Number], required: true } }, setup(props) { const selected =
ref(props.selected); watch(() => props.selected, (newValue) => {
selected.value = newValue; console.log("Selected prop changed to:",
newValue); }); return { selected }; } }); </script> This will watch for changes in the
selected prop and update the selected ref accordingly.
```

Month: Jan-20

Question 1:

Tags: javascript, typescript, reactjs, react-functional-component

Question: I am learning TypeScript and some bits are confusing to me. One bit is below: `interface Props { name: string; } const PrintName: React.FC<Props> = (props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>) } const PrintName2 = (props: Props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>) }` For both functional components above, I see TypeScript generates the same JS code. The `PrintName2` component seems more streamlined to me as far as readability. I wonder what's the difference between the two definitions and if anyone is using second type of React component?

Answer Score: 204

Stack Overflow answer, Verb Count: 27 , Adjective Count: 37

Stack Overflow Answer: Thanks all for the answers. They are correct but I was looking for a more detailed version. I did some more research and found this on React+TypeScript Cheatsheets on GitHub. Function Components These can be written as normal functions that take a props argument and return a JSX element. `type AppProps = { message: string }; /* could also use interface */ const App = ({ message }: AppProps) => <div>{message}</div>;` What about `React.FC/React.FunctionComponent`? You can also write components with `React.FunctionComponent` (or the shorthand `React.FC`): `const App: React.FC<{ message: string }> = ({ message }) => (<div>{message}</div>);` Some differences from the "normal function" version: It provides typechecking and autocomplete for static properties like `displayName`, `propTypes`, and `defaultProps` - However, there are currently known issues using `defaultProps` with `React.FunctionComponent`. See this issue for details - scroll down to our `defaultProps` section for typing recommendations there. It provides an implicit definition of children (see below) - however there are some issues with the implicit children type (e.g. DefinitelyTyped#33006), and it might be considered a better style to be explicit about components that consume children, anyway. `const Title: React.FunctionComponent<{ title: string }> = ({ children, title }) => <div title={title}>{children}</div>;` In the future, it may automatically mark props as readonly, though that's a moot point if the props object is destructured in the parameter list. `React.FunctionComponent` is explicit about the return type, while the normal function version is implicit (or else needs additional annotation). In most cases, it makes very little difference which syntax is used, but the `React.FC` syntax is slightly more verbose without providing clear advantage, so precedence was given to the "normal function" syntax.

GPT answer, Verb Count: 32 , Adjective Count: 37

GPT-4o Answer: ``html The difference between the two lies in how the TypeScript type is defined and enforced for the functional components. Below is an explanation: `React.FC<Props>: const PrintName: React.FC = (props) => { return (<div> <p style={{ fontWeight:`


```

props.priority ? "bold" : "normal" }}> {props.name} </p> </div> ); }; 1.
React.FC automatically types the props for the functional component using the provided interface
(Props in this case). 2. It adds implicit support for children, even if not specified in Props. This can be
an advantage when you want to allow children by default. 3. It ensures the return type is a React
element, which can make the code more type-safe. Plain Type Annotation (props: Props): const
PrintName2 = (props: Props) => { return ( <div> <p style={{ fontWeight:
props.priority ? "bold" : "normal" }}> {props.name} </p> </div> ); }; 1. You
explicitly type the props parameter using the Props interface. 2. The component does not implicitly
include children. If you need children, you must explicitly add them to the Props interface. 3. The
return type is inferred by TypeScript, so it might not provide the same level of strict type enforcement as
React.FC. **Key Differences**: - **Implicit Children**: React.FC includes children automatically,
while the plain annotation does not. - **Return Type**: React.FC enforces the return type to be a
React element. - **Readability**: Plain annotation (props: Props) is more streamlined and often
preferred for simplicity unless you need the additional features of React.FC. **Which to Use?** - Use
React.FC if you need to include children implicitly or prefer stricter type safety for the return type. -
Use plain annotation for simplicity and readability, especially if you don't require children or the
additional features of React.FC. ``

```

Co-Pilot, Verb Count: 19 , Adjective Count: 29

Co-Pilot Answer: The difference between the two definitions lies mainly in how they leverage TypeScript types. Here's a detailed comparison: `interface Props { name: string; }` Using `React.FC`: `const PrintName: React.FC<Props> = (props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>); }` Using standard function syntax: `const PrintName2 = (props: Props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>); }` Key Differences: 1. `React.FC`: The `React.FC` type (or `React.FunctionComponent`) is a helper type from React that provides additional type information. It implicitly adds the `children` prop to the component, ensuring it can accept children elements. This can be useful if you want to pass children to the component. 2. Standard function syntax: This approach is more streamlined and does not implicitly add the `children` prop. It's more straightforward and preferred when you don't need the additional props added by `React.FC`. Both approaches are valid, and the choice depends on your preference and specific use cases. If you don't need the extra functionality provided by `React.FC`, using the standard function syntax can be more concise and readable.

Question 2:

Tags: python, visual-studio-code, python-poetry

Question: I need help. VSCode will NEVER find poetry virtualenv interpreter no matter what I try. Installed poetry Python package manager using a standard `$ curl` method as explained in the official documentation. Started a project by `$ poetry new finance-essentials_37-64`, installed poetry environment with `$ poetry install`. So now I can see that I indeed have a virtual environment by: `Jaepil@Jaepil-PC MINGW64 /e/VSCoDeProjects/finance_essentials_37-64 $ poetry env list >> finance-essentials-37-64-SCQrHB_N-py3.7 (Activated)` and this virtualenv is installed at: `C:\Users\Jaepil\AppData\Local\pypoetry\Cache\virtualenvs`, which has `finance-essentials-37-64-SCQrHB_N-py3.7` directory. However, VSCode is unable to find this virtualenv in its 'select interpreter' command. I only see a bunch of Anaconda and Pipenv environments but not the poetry environment's interpreter that I've just made. I also added `"python.venvPath": "~/cache/pypoetry/virtualenvs"`, to my `settings.json` as suggested in here, but to no avail. Still doesn't work. I also tried an absolute path, by adding `"python.venvPath": "C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\Cache\\virtualenvs"`, to the same

settings, but it also doesn't work. VSCode settings reference states that it has `python.poetryPath` as a default but it doesn't seem to work either. Should I change the default value "poetry" in this case? `python.poetryPath "poetry"` Specifies the location of the Poetry dependency manager executable, if installed. The default value "poetry" assumes the executable is in the current path. The Python extension uses this setting to install packages when Poetry is available and there's a `poetry.lock` file in the workspace folder. I'm on Windows 10 pro 64bit & Has Python 3.7.6 installed on the system. PS C:\Users\Jaepil> python Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32

Answer Score: 474

Stack Overflow answer, Verb Count: 19 , **Adjective Count:** 8

Stack Overflow Answer: You just need to type in your shell: `poetry config virtualenvs.in-project true` The virtualenv will be created inside the project path and vscode will recognize. Consider adding this to your `.bashrc` or `.zshrc`. If you already have created your project, you need to re-create the virtualenv to make it appear in the correct place: `poetry env list` # shows the name of the current environment `poetry env remove <current environment>` `poetry install` # will create a new environment using your updated configuration

GPT answer, Verb Count: 30 , **Adjective Count:** 29

GPT-4o Answer: The issue seems to be related to VSCode not detecting the Poetry virtual environment automatically. Below are the steps to resolve this: 1. Locate the Virtual Environment Path: `poetry env info --path` This command will give you the exact path to the virtual environment directory. Copy this path for the next steps. 2. Manually Add the Interpreter to VSCode: In VSCode, press `Ctrl + Shift + P` (or `Cmd + Shift + P` on macOS) and select `Python: Select Interpreter`. If your Poetry environment does not appear, proceed to the next step. 3. Add the Interpreter Path Manually: (a) Open the Command Palette (`Ctrl + Shift + P`) and select `Python: Select Interpreter`. (b) Choose `Enter interpreter path` and select `Find...` (c) Navigate to the copied path from step 1 and select the Python executable inside the `bin` (Linux/macOS) or `Scripts` (Windows) directory, e.g., `C:\Users\Jaepil\AppData\Local\pypoetry\Cache\virtualenvs\finance-essentials-37-64-SCQrHB_N-py3.7\Scripts\python.exe`. 4. Update VSCode Settings: To make the Poetry virtual environment path a default for VSCode, add this to your VSCode `settings.json`: { "python.venvPath": "C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\Cache\\virtualenvs", "python.poetryPath": "C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\poetry.exe" // or the path where poetry is installed } 5. Verify Poetry Path: If you're using a custom Poetry installation path, ensure the executable is available in the system PATH or provide the full path in "python.poetryPath" as shown above. 6. Restart VSCode: Restart VSCode after making these changes and try selecting the interpreter again. If the issue persists, ensure that the Python extension in VSCode is updated to the latest version, as older versions might not correctly support Poetry environments.

Co-Pilot, Verb Count: 24 , **Adjective Count:** 4

Co-Pilot Answer: It sounds like you've done a lot of troubleshooting already! Here's a step-by-step solution to help VSCode recognize your Poetry virtualenv: Open your VSCode `settings.json` file and add the following line: `"python.poetryPath": "C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\Cache\\virtualenvs"` Save the `settings.json` file and restart VSCode. Press `Ctrl+Shift+P` to open the command palette and type `Python: Select Interpreter`. You should now see your Poetry virtualenv listed under the available interpreters. Select it to set it as your active interpreter. If it's still not showing up, try running the

following command in your terminal to get the exact path of your Poetry virtualenv and manually add it: `poetry env info --path` Copy the path and add it to the `python.venvPath` setting in your `settings.json` file. Does this help resolve the issue?

Question 3:

Tags: javascript, arrays, function, typescript, optional-chaining

Question: I'm trying to use optional chaining with an array instead of an object but not sure how to do that: Here's what I'm trying to do `myArray.filter(x => x.testKey === myTestKey)?.[0]`. Also trying similar thing with a function: `let x = {a: () => {}, b: null} console.log(x?.b());` But it's giving a similar error - how can I use optional chaining with an array or a function?

Answer Score: 569

Stack Overflow answer, Verb Count: 14 , **Adjective Count:** 11

Stack Overflow Answer: You need to put a `.` after the `?` to use optional chaining: `myArray.filter(x => x.testKey === myTestKey)?.[0]` Playground link Using just the `?` alone makes the compiler think you're trying to use the conditional operator (and then it throws an error since it doesn't see a `:` later) Optional chaining isn't just a TypeScript thing - it is a finished proposal in plain JavaScript too. It can be used with bracket notation like above, but it can also be used with dot notation property access: `const obj = { prop2: { nested2: 'val2' } }; console.log(obj.prop1?.nested1, obj.prop2?.nested2);` And with function calls: `const obj = { fn2: () => console.log('fn2 running') }; obj.fn1?.(); obj.fn2?.();`

GPT answer, Verb Count: 48 , **Adjective Count:** 16

GPT-4o Answer: I need you to answer the following question. I will be copying and pasting your answer in excel so I can later compare it to other answers. When answering the question please type out your reply in plain text using html tags, header tags, bold tags or code tags. Do not use any other html tags such as lists or `.` You may not use any markdown code and this includes markdown that is rendered. I don't want to see any formatting. If you have any code examples can please inclose them inside tags. Do not try and structure your response like a HTML document by using things like I am only asking you to use html tags like and to add structure to the document. I only want you to return your answer to the question, do not type the question back to me in your response. Only answer the question using plain text and html tags do not further explain anything after in markdown. Simply include just your answer do not start off the answer by writing anything such as "Answer:" or "Here is your answer:" or anything along those lines. Follow these instructions very strictly, here is the question I need you to type out the answer for: I'm trying to use optional chaining with an array instead of an object but not sure how to do that: Here's what I'm trying to do `myArray.filter(x => x.testKey === myTestKey)?.[0]`. Also trying similar thing with a function: `let x = {a: () => {}, b: null} console.log(x?.b());` But it's giving a similar error - how can I use optional chaining with an array or a function?

Co-Pilot, Verb Count: 8 , **Adjective Count:** 9

Co-Pilot Answer: To use optional chaining with an array and a function, you can modify your approach as follows: With an array: `const myArray = [{ testKey: 1 }, { testKey: 2 }]; const myTestKey = 1; const result = myArray.filter(x => x.testKey === myTestKey)?.[0]; console.log(result);` With a function: `let x = { a: () => {}, b: null }; console.log(x?.b?.());` Using `?.` ensures that if the preceding value is `null` or

undefined, the expression short-circuits and returns undefined, preventing errors.

Question 4:

Tags: java, intellij-idea

Question: I'm using IntelliJ IDEA Ultimate 2019.3.1. Whenever I try to start any simple Java Maven project (may it be even a simple Hello World) I get the following error: `Error:java: error: release version 5 not supported` Running `java --version` by terminal I get the following output: `openjdk 11.0.5 2019-10-15 OpenJDK Runtime Environment (build 11.0.5+10-post-Ubuntu-0ubuntu1.1) OpenJDK 64-Bit Server VM (build 11.0.5+10-post-Ubuntu-0ubuntu1.1, mixed mode, sharing)` Running `javac --version` by terminal I get the following output: `javac 11.0.5` Going to the Settings of the Java Compiler (as suggested here) I see this: I tried editing the "Target bytecode version" to 1.8 but I get the following errors: `Error:(1, 26) java: package javafx.application does not exist` `Error:(2, 20) java: package javafx.stage does not exist` `Error:(4, 27) java: cannot find symbol symbol: class Application` `Error:(12, 23) java: cannot find symbol symbol: class Stage location: class Main` `Error:(7, 9) java: cannot find symbol symbol: method launch(java.lang.String[]) location: class Main` `Error:(11, 5) java: method does not override or implement a method from a supertype` Changing it to version 1.11 I get this error instead: `Error:java: Source option 5 is no longer supported. Use 6 or later.` What do you think is the problem? How may I solve it?

Answer Score: 483

Stack Overflow answer, Verb Count: 12 , Adjective Count: 5

Stack Overflow Answer: See <https://stackoverflow.com/a/12900859/104891>. First of all, set the language level/release versions in `pom.xml` like that: `<properties>`
`<maven.compiler.source>1.8</maven.compiler.source>`
`<maven.compiler.target>1.8</maven.compiler.target>` `</properties>` Maven sets the default to 1.5 otherwise. You will also need to include the `maven-compiler-plugin` if you haven't already: `<dependency>` `<groupId>org.apache.maven.plugins</groupId>`
`<artifactId>maven-compiler-plugin</artifactId>` `<version>3.8.1</version>`
`</dependency>` Also, try to change the Java version in each of these places: File -> Project structure -> Project -> Project SDK -> 11. File -> Project structure -> Project -> Project language level -> 11. File -> Project structure -> Project -> Modules -> -> Sources -> 11 In project -> ctrl + alt + s -> Build, Execution, Deployment -> Compiler -> Java Compiler -> Project bytecode version -> 11 In project -> ctrl + alt + s -> Build, Execution, Deployment -> Compiler -> Java Compiler -> Module -> 1.11.

GPT answer, Verb Count: 20 , Adjective Count: 10

GPT-4o Answer: The issue arises because your project or IDE is configured to use Java 5 as the source compatibility version, which is not supported by JDK 11. To resolve this issue, follow these steps: 1. Update Project's Java Version: `File > Project Structure > Project > Project SDK: Set to Java 11` `File > Project Structure > Project > Language Level: Set to Java 11 (e.g., "SDK default (11)")`. 2. Update Compiler Settings: `File > Settings > Build, Execution, Deployment > Compiler > Java Compiler > Target bytecode version: Set to 11`. 3. Update Maven Compiler Plugin: Ensure that your Maven project's `pom.xml` file has the correct compiler plugin configuration: `<build>` `<plugins>` `<plugin>`
`<groupId>org.apache.maven.plugins</groupId>`
`<artifactId>maven-compiler-plugin</artifactId>` `<version>3.8.1</version>`
`<configuration>` `<source>11</source>` `<target>11</target>` `</configuration>`
`</plugin>` `</plugins>` `</build>` 4. Verify JavaFX Dependencies (if needed): If your project uses JavaFX, you need to add the JavaFX libraries explicitly because they are no longer bundled with

JDK 11. Add the appropriate JavaFX SDK dependency to your Maven `pom.xml`: `<dependencies>`
`<dependency>` `<groupId>org.openjfx</groupId>`
`<artifactId>javafx-controls</artifactId>` `<version>11.0.2</version>`
`</dependency>` `</dependencies>` 5. Reload Project: Right-click on your Maven
project in IntelliJ > Reload Project. After these changes, your project should compile
successfully with JDK 11.

Co-Pilot, Verb Count: 20 , Adjective Count: 3

Co-Pilot Answer: It seems like the issue is related to the Java version compatibility with IntelliJ IDEA 2019.3.1. Here's how you can resolve it: Open IntelliJ IDEA and go to File > Project Structure. Under Project, set the Project SDK to 11 (since you have OpenJDK 11 installed). Go to Modules, select your module, and set the Module SDK to 11 as well. Under Project > Project Compiler Output, ensure the path is correct. Go to Build, Compiler > Java Compiler, and set the Target bytecode version to 11. Apply the changes and rebuild your project. This should resolve the version mismatch error. If you still encounter issues, consider updating IntelliJ IDEA to a newer version that supports Java 11 out of the box.

Question 5:

Tags: java, maven, jenkins, maven-central

Question: Recently Maven build jobs running in Jenkins are failing with the below exception saying that they couldn't pull dependencies from Maven Central and should use HTTPS. I'm not sure how to change the requests from HTTP to HTTPS. Could someone guide me on this matter? [ERROR] Unresolvable build extension: Plugin `org.apache.maven.wagon:wagon-ssh:2.1` or one of its dependencies could not be resolved: Failed to collect dependencies for `org.apache.maven.wagon:wagon-ssh:jar:2.1` (): Failed to read artifact descriptor for `org.apache.maven.wagon:wagon-ssh:jar:2.1`: Could not transfer artifact `org.apache.maven.wagon:wagon-ssh:pom:2.1` from/to central (`http://repo.maven.apache.org/maven2`): Failed to transfer file: `http://repo.maven.apache.org/maven2/org/apache/maven/wagon/wagon-ssh/2.1/wagon-ssh-2.1.pom`. Return code is: 501, ReasonPhrase:HTTPS Required. -> [Help 2] Waiting for Jenkins to finish collecting data[ERROR] Plugin `org.apache.maven.plugins:maven-clean-plugin:2.4.1` or one of its dependencies could not be resolved: Failed to read artifact descriptor for `org.apache.maven.plugins:maven-clean-plugin:jar:2.4.1`: Could not transfer artifact `org.apache.maven.plugins:maven-clean-plugin:pom:2.4.1` from/to central (`http://repo.maven.apache.org/maven2`): Failed to transfer file: `http://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.4.1/maven-clean-plugin-2.4.1.pom`. Return code is: 501 , ReasonPhrase:HTTPS Required. -> [Help 1]

Answer Score: 178

Stack Overflow answer, Verb Count: 16 , Adjective Count: 6

Stack Overflow Answer: The reason for the observed error is explained in Central 501 HTTPS Required Effective January 15, 2020, The Central Repository no longer supports insecure communication over plain HTTP and requires that all requests to the repository are encrypted over HTTPS. It looks like latest versions of Maven (tried with 3.6.0, 3.6.1) are already using the HTTPS URL by default. Here are the dates when the major repositories will switch: Your Java builds might break starting January 13th (if you haven't yet switched repo access to HTTPS) Update: Seems like from maven 3.2.3 maven central is accessed via HTTPS See <https://stackoverflow.com/a/25411658/5820670> Maven Change log (<http://maven.apache.org/docs/3.2.3/release-notes.html>)

GPT answer, Verb Count: 20 , Adjective Count: 16

GPT-4o Answer: The error occurs because Maven Central now requires HTTPS for artifact downloads, and your Jenkins configuration or `pom.xml` file is still pointing to the old HTTP URLs. To resolve this issue, you need to update your Maven settings to use HTTPS. 1. Update Maven Settings: Edit the `settings.xml` file, which is typically located in `~/.m2/settings.xml` (for Linux/macOS) or `C:\Users\YourUsername\.m2\settings.xml` (for Windows). Replace the repository URL with the HTTPS version: `<mirrors> <mirror> <id>central</id> <mirrorOf>central</mirrorOf> <url>https://repo.maven.apache.org/maven2</url> </mirror> </mirrors>` 2. Update `pom.xml` (if needed): If your `pom.xml` contains explicit references to the old HTTP repository, update them to HTTPS: `<repositories> <repository> <id>central</id> <url>https://repo.maven.apache.org/maven2</url> </repository> </repositories>` 3. Verify Jenkins Maven Configuration: In Jenkins, ensure that your Maven configuration is pointing to the correct `settings.xml` file: `Manage Jenkins > Global Tool Configuration > Maven > Settings file > Path to your updated settings.xml`. 4. Test the Build: Run your build job in Jenkins again. The HTTPS configuration should resolve the dependency issues.

Co-Pilot, Verb Count: 13 , Adjective Count: 2

Co-Pilot Answer: The error you are encountering indicates that Maven Central requires HTTPS for secure communication. To resolve this issue, you need to update your Maven configuration to use HTTPS instead of HTTP. Here's how you can do it: `<!-- Update your Maven settings.xml file, usually located in the .m2 directory --> <settings xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/settings-1.0.0.xsd"> <mirrors> <mirror> <id>central</id> <mirrorOf>central</mirrorOf> <url>https://repo.maven.apache.org/maven2</url> <releases> <enabled>true</enabled> </releases> <snapshots> <enabled>true</enabled> </snapshots> </mirror> </mirrors> </settings>` This configuration will ensure that Maven uses HTTPS when connecting to Maven Central, resolving the dependency issues.

Month: Feb-20

Question 1:

Tags: javascript, typescript, reactjs, react-functional-component

Question: I am learning TypeScript and some bits are confusing to me. One bit is below: `interface Props { name: string; } const PrintName: React.FC<Props> = (props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>) } const PrintName2 = (props: Props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>) }` For both functional components above, I see TypeScript generates the same JS code. The `PrintName2` component seems more streamlined to me as far as readability. I wonder what's the difference between the two definitions and if anyone is using second type of React component?

Answer Score: 204

Stack Overflow answer, Verb Count: 27 , Adjective Count: 37

Stack Overflow Answer: Thanks all for the answers. They are correct but I was looking for a more detailed version. I did some more research and found this on React+TypeScript Cheatsheets on

GitHub. Function Components These can be written as normal functions that take a props argument and return a JSX element. `type AppProps = { message: string }; /* could also use interface */ const App = ({ message }: AppProps) => <div>{message}</div>;` What about `React.FC/React.FunctionComponent`? You can also write components with `React.FunctionComponent` (or the shorthand `React.FC`): `const App: React.FC<{ message: string }> = ({ message }) => (<div>{message}</div>);` Some differences from the "normal function" version: It provides typechecking and autocomplete for static properties like `displayName`, `propTypes`, and `defaultProps` - However, there are currently known issues using `defaultProps` with `React.FunctionComponent`. See this issue for details - scroll down to our `defaultProps` section for typing recommendations there. It provides an implicit definition of children (see below) - however there are some issues with the implicit children type (e.g. DefinitelyTyped#33006), and it might be considered a better style to be explicit about components that consume children, anyway. `const Title: React.FunctionComponent<{ title: string }> = ({ children, title }) => <div title={title}>{children}</div>;` In the future, it may automatically mark props as readonly, though that's a moot point if the props object is destructured in the parameter list. `React.FunctionComponent` is explicit about the return type, while the normal function version is implicit (or else needs additional annotation). In most cases, it makes very little difference which syntax is used, but the `React.FC` syntax is slightly more verbose without providing clear advantage, so precedence was given to the "normal function" syntax.

GPT answer, Verb Count: 26 , Adjective Count: 29

GPT-4o Answer: The difference between the two definitions lies in how the props are typed and how TypeScript infers types. Here's a breakdown: 1. Using `React.FC` (`PrintName`): When using `React.FC<Props>`, TypeScript automatically infers certain things: - The component is a functional component. - The component's props are typed explicitly. - `children` is automatically included in the props (even if not specified in `Props`). - Provides additional type checks and autocomplete for static properties like `defaultProps` and `propTypes`. However, `React.FC` has limitations. It includes `children` by default, which can sometimes lead to unexpected behavior if your component does not expect `children`. 2. Without `React.FC` (`PrintName2`): When not using `React.FC`, you explicitly type the props parameter: `const PrintName2 = (props: Props) => { ... }` This approach does not include `children` unless explicitly defined in `Props`. It is more lightweight and avoids potential issues with unexpected `children`. Recommendation: The second approach (`PrintName2`) is often preferred for modern React projects because it avoids the inclusion of unnecessary `children` in the props and is simpler. Use `React.FC` if you specifically need the additional features it provides.

Co-Pilot, Verb Count: 19 , Adjective Count: 29

Co-Pilot Answer: The difference between the two definitions lies mainly in how they leverage TypeScript types. Here's a detailed comparison: `interface Props { name: string; }` Using `React.FC`: `const PrintName: React.FC<Props> = (props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>); }` Using standard function syntax: `const PrintName2 = (props: Props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>); }` Key Differences: 1. `React.FC`: The `React.FC` type (or `React.FunctionComponent`) is a helper type from React that provides additional type information. It implicitly adds the `children` prop to the component, ensuring it can accept children elements. This can be useful if you want to pass children to the component. 2. Standard function syntax: This approach is more streamlined and does not implicitly add the `children` prop. It's more straightforward and preferred when you don't need the additional props added by `React.FC`. Both approaches are valid, and the choice depends on your preference and specific use cases. If you don't need the extra functionality provided by `React.FC`, using the standard function syntax can be more concise and readable.

Question 2:

Tags: python, visual-studio-code, python-poetry

Question: I need help. VSCode will NEVER find poetry virtualenv interpreter no matter what I try. Installed poetry Python package manager using a standard `$ curl` method as explained in the official documentation. Started a project by `$ poetry new finance-essentials_37-64`, installed poetry environment with `$ poetry install`. So now I can see that I indeed have a virtual environment by: `Jaepil@Jaepil-PC MINGW64 /e/VSCoDeProjects/finance_essentials_37-64 $ poetry env list >> finance-essentials-37-64-SCQrHB_N-py3.7 (Activated)` and this virtualenv is installed at: `C:\Users\Jaepil\AppData\Local\pypoetry\Cache\virtualenvs`, which has `finance-essentials-37-64-SCQrHB_N-py3.7` directory. However, VSCode is unable to find this virtualenv in its 'select interpreter' command. I only see a bunch of Anaconda and Pipenv environments but not the poetry environment's interpreter that I've just made. I also added `"python.venvPath": "~/ .cache/pypoetry/virtualenvs"`, to my `settings.json` as suggested in here, but to no avail. Still doesn't work. I also tried an absolute path, by adding `"python.venvPath": "C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\Cache\\virtualenvs"`, to the same settings, but it also doesn't work. VSCode settings reference states that it has `python.poetryPath` as a default but it doesn't seem to work either. Should I change the default value `"poetry"` in this case? `python.poetryPath "poetry"` Specifies the location of the Poetry dependency manager executable, if installed. The default value `"poetry"` assumes the executable is in the current path. The Python extension uses this setting to install packages when Poetry is available and there's a `poetry.lock` file in the workspace folder. I'm on Windows 10 pro 64bit & Has Python 3.7.6 installed on the system. `PS C:\Users\Jaepil> python Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32`

Answer Score: 474

Stack Overflow answer, Verb Count: 19 , **Adjective Count:** 8

Stack Overflow Answer: You just need to type in your shell: `poetry config virtualenvs.in-project true` The virtualenv will be created inside the project path and vscode will recognize. Consider adding this to your `.bashrc` or `.zshrc`. If you already have created your project, you need to re-create the virtualenv to make it appear in the correct place: `poetry env list` # shows the name of the current environment `poetry env remove <current environment>` `poetry install` # will create a new environment using your updated configuration

GPT answer, Verb Count: 33 , **Adjective Count:** 26

GPT-4o Answer: To resolve the issue where VSCode does not detect the Poetry virtual environment, follow these steps: 1. Verify Poetry Path Ensure that the Poetry executable is accessible from your system PATH. You can verify this by running: `poetry --version` If this command works, the Poetry path is correctly configured. If not, add the Poetry path to your system PATH environment variable. 2. Set the Poetry Virtual Environment Path in VSCode Add the virtual environment path to your VSCode `settings.json`. Use the full absolute path for `python.venvPath`: `{ "python.venvPath": "C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\Cache\\virtualenvs" }` Restart VSCode after making this change. 3. Select the Interpreter Manually If VSCode still does not detect the interpreter, locate the exact Python executable inside the virtual environment directory: `C:\\Users\\Jaepil\\AppData\\Local\\pypoetry\\Cache\\virtualenvs\\finance-essentials-37-64-SCQrHB_N-py3.7\\Scripts\\python.exe` Use the VSCode "Select Interpreter" command (`Ctrl+Shift+P` > "Python: Select Interpreter") and choose the above path manually. 4. Use the `python.poetryPath` Setting In case Poetry is not being detected, explicitly set the Poetry executable in `settings.json`: `{ "python.poetryPath":`

```
"C:\\Users\\Jaepil\\AppData\\Local\\Programs\\Python\\Poetry\\poetry.exe" }
```

Replace the path with the actual location of the Poetry executable on your system. 5. Reinstall VSCode Python Extension If none of the above steps work, try reinstalling the Python extension in VSCode. Sometimes, extension updates resolve environment detection issues. 6. Activate the Virtual Environment in VSCode Terminal Activate the Poetry environment in the integrated terminal before running your code: `poetry shell` This ensures the correct environment is being used for running scripts or commands.

Co-Pilot, Verb Count: 19 , Adjective Count: 29

Co-Pilot Answer: The difference between the two definitions lies mainly in how they leverage TypeScript types. Here's a detailed comparison: `interface Props { name: string; }` Using `React.FC`: `const PrintName: React.FC<Props> = (props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>); }` Using standard function syntax: `const PrintName2 = (props: Props) => { return (<div> <p style={{ fontWeight: props.priority ? "bold" : "normal" }}> {props.name} </p> </div>); }` Key Differences: 1. `React.FC`: The `React.FC` type (or `React.FunctionComponent`) is a helper type from React that provides additional type information. It implicitly adds the `children` prop to the component, ensuring it can accept children elements. This can be useful if you want to pass children to the component. 2. Standard function syntax: This approach is more streamlined and does not implicitly add the `children` prop. It's more straightforward and preferred when you don't need the additional props added by `React.FC`. Both approaches are valid, and the choice depends on your preference and specific use cases. If you don't need the extra functionality provided by `React.FC`, using the standard function syntax can be more concise and readable.

Question 3:

Tags: javascript, arrays, function, typescript, optional-chaining

Question: I'm trying to use optional chaining with an array instead of an object but not sure how to do that: Here's what I'm trying to do `myArray.filter(x => x.testKey === myTestKey)?.[0]`. Also trying similar thing with a function: `let x = {a: () => {}, b: null} console.log(x?.b());` But it's giving a similar error - how can I use optional chaining with an array or a function?

Answer Score: 569

Stack Overflow answer, Verb Count: 14 , Adjective Count: 11

Stack Overflow Answer: You need to put a `.` after the `?` to use optional chaining: `myArray.filter(x => x.testKey === myTestKey)?.[0]` Playground link Using just the `?` alone makes the compiler think you're trying to use the conditional operator (and then it throws an error since it doesn't see a `:` later) Optional chaining isn't just a TypeScript thing - it is a finished proposal in plain JavaScript too. It can be used with bracket notation like above, but it can also be used with dot notation property access: `const obj = { prop2: { nested2: 'val2' } }; console.log(obj.prop1?.nested1, obj.prop2?.nested2);` And with function calls: `const obj = { fn2: () => console.log('fn2 running') }; obj.fn1?.(); obj.fn2?.();`

GPT answer, Verb Count: 26 , Adjective Count: 10

GPT-4o Answer: Optional chaining can be used with arrays and functions, but it requires proper syntax. Here's how you can fix your issues: 1. Using Optional Chaining with an Array You cannot directly use `?.` after an array operation like `filter`. Instead, you need to apply it to the element accessed after filtering. Here's the correct syntax: `const firstMatch = myArray.filter(x => x.testKey === myTestKey)?.[0];` This ensures that `?.` is applied to the result of the filter operation, safely accessing the first element only if the array exists and has elements. 2. Using Optional

Chaining with a Function To safely call a function that might not exist, you can use the following syntax: `let x = { a: () => {}, b: null }; console.log(x?.b?.());` This ensures that `b` is checked for existence before attempting to call it as a function. These changes should resolve the errors you encountered.

Co-Pilot, Verb Count: 8 , Adjective Count: 9

Co-Pilot Answer: To use optional chaining with an array and a function, you can modify your approach as follows: With an array: `const myArray = [{ testKey: 1 }, { testKey: 2 }]; const myTestKey = 1; const result = myArray.filter(x => x.testKey === myTestKey)?.[0]; console.log(result);` With a function: `let x = { a: () => {}, b: null }; console.log(x?.b?.());` Using `?.` ensures that if the preceding value is `null` or `undefined`, the expression short-circuits and returns `undefined`, preventing errors.

Question 4:

Tags: java, intellij-idea

Question: I'm using IntelliJ IDEA Ultimate 2019.3.1. Whenever I try to start any simple Java Maven project (may it be even a simple Hello World) I get the following error: `Error:java: error: release version 5 not supported` Running `java --version` by terminal I get the following output: `openjdk 11.0.5 2019-10-15 OpenJDK Runtime Environment (build 11.0.5+10-post-Ubuntu-0ubuntu1.1) OpenJDK 64-Bit Server VM (build 11.0.5+10-post-Ubuntu-0ubuntu1.1, mixed mode, sharing)` Running `javac --version` by terminal I get the following output: `javac 11.0.5` Going to the Settings of the Java Compiler (as suggested here) I see this: I tried editing the "Target bytecode version" to 1.8 but I get the following errors: `Error:(1, 26) java: package javafx.application does not exist` `Error:(2, 20) java: package javafx.stage does not exist` `Error:(4, 27) java: cannot find symbol symbol: class Application` `Error:(12, 23) java: cannot find symbol symbol: class Stage location: class Main` `Error:(7, 9) java: cannot find symbol symbol: method launch(java.lang.String[]) location: class Main` `Error:(11, 5) java: method does not override or implement a method from a supertype` Changing it to version 1.11 I get this error instead: `Error:java: Source option 5 is no longer supported. Use 6 or later.` What do you think is the problem? How may I solve it?

Answer Score: 483

Stack Overflow answer, Verb Count: 12 , Adjective Count: 5

Stack Overflow Answer: See <https://stackoverflow.com/a/12900859/104891>. First of all, set the `language level/release versions` in `pom.xml` like that: `<properties>`
`<maven.compiler.source>1.8</maven.compiler.source>`
`<maven.compiler.target>1.8</maven.compiler.target>` `</properties>` Maven sets the default to 1.5 otherwise. You will also need to include the `maven-compiler-plugin` if you haven't already: `<dependency>` `<groupId>org.apache.maven.plugins</groupId>`
`<artifactId>maven-compiler-plugin</artifactId>` `<version>3.8.1</version>`
`</dependency>` Also, try to change the Java version in each of these places: File -> Project structure -> Project -> Project SDK -> 11. File -> Project structure -> Project -> Project language level -> 11. File -> Project structure -> Project -> Modules -> -> Sources -> 11 In project -> ctrl + alt + s -> Build, Execution, Deployment -> Compiler -> Java Compiler -> Project bytecode version -> 11 In project -> ctrl + alt + s -> Build, Execution, Deployment -> Compiler -> Java Compiler -> Module -> 1.11.

GPT answer, Verb Count: 26 , Adjective Count: 13

GPT-4o Answer: The issue is that your project is configured to use an outdated source version (Java 5), which is not supported by your JDK (Java 11). Here's how to resolve it: 1. Update Maven Compiler

Plugin Configuration Ensure that your `pom.xml` specifies the correct source and target versions compatible with Java 11: `<build> <plugins> <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-compiler-plugin</artifactId> <version>3.8.1</version> <configuration> <source>11</source> <target>11</target> </configuration> </plugin> </plugins> </build>` This ensures Maven compiles your code with Java 11. 2. Update IntelliJ Compiler Settings In IntelliJ IDEA: - Go to File → Project Structure → Project. - Set the Project SDK to JDK 11. - Set the Project language level to 11 - `Local variable syntax for lambda parameters`. - Under Modules, ensure the language level and SDK are also set to Java 11. 3. Check Your Build Tool If using Maven, ensure IntelliJ is configured to use the correct JDK for Maven: - Go to File → Settings → Build, Execution, Deployment → Build Tools → Maven. - Set the JVM under Runner to your Java 11 installation. 4. Update Dependencies If your project uses JavaFX or other libraries, ensure they are compatible with Java 11. JavaFX is no longer bundled with Java 11 and needs to be added as a separate dependency. Example for Maven: `<dependencies> <dependency> <groupId>org.openjfx</groupId> <artifactId>javafx-controls</artifactId> <version>11.0.2</version> </dependency> </dependencies>` Following these steps should resolve the compilation and version errors.

Co-Pilot, Verb Count: 21 , Adjective Count: 4

Co-Pilot Answer: The issue you're encountering is due to a version mismatch between the Java Development Kit (JDK) you're using and the target bytecode version set in IntelliJ IDEA. Here's how you can resolve it: `Open IntelliJ IDEA and go to File > Project Structure. Under Project, set the Project SDK to 11 (since you have OpenJDK 11 installed). Go to Modules, select your module, and set the Module SDK to 11 as well. Under Project > Project Compiler Output, ensure the path is correct. Go to Build, Compiler > Java Compiler, and set the Target bytecode version to 11. Apply the changes and rebuild your project.` This should resolve the version mismatch error. If you still encounter issues, consider updating IntelliJ IDEA to a newer version that supports Java 11 out of the box.

Question 5:

Tags: java, maven, jenkins, maven-central

Question: Recently Maven build jobs running in Jenkins are failing with the below exception saying that they couldn't pull dependencies from Maven Central and should use HTTPS. I'm not sure how to change the requests from HTTP to HTTPS. Could someone guide me on this matter? [ERROR] Unresolvable build extension: Plugin `org.apache.maven.wagon:wagon-ssh:2.1` or one of its dependencies could not be resolved: Failed to collect dependencies for `org.apache.maven.wagon:wagon-ssh:jar:2.1 ()`: Failed to read artifact descriptor for `org.apache.maven.wagon:wagon-ssh:jar:2.1`: Could not transfer artifact `org.apache.maven.wagon:wagon-ssh:pom:2.1` from/to central (`http://repo.maven.apache.org/maven2`): Failed to transfer file: `http://repo.maven.apache.org/maven2/org/apache/maven/wagon/wagon-ssh/2.1/wagon-ssh-2.1.pom`. Return code is: 501, ReasonPhrase:HTTPS Required. -> [Help 2] Waiting for Jenkins to finish collecting data[ERROR] Plugin `org.apache.maven.plugins:maven-clean-plugin:2.4.1` or one of its dependencies could not be resolved: Failed to read artifact descriptor for `org.apache.maven.plugins:maven-clean-plugin:jar:2.4.1`: Could not transfer artifact `org.apache.maven.plugins:maven-clean-plugin:pom:2.4.1` from/to central (`http://repo.maven.apache.org/maven2`): Failed to transfer file: `http://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.4.1/maven-clean-plugin-2.4.1.pom`. Return code is:

501 , ReasonPhrase:HTTPS Required. -> [Help 1]

Answer Score: 178

Stack Overflow answer, Verb Count: 16 , **Adjective Count:** 6

Stack Overflow Answer: The reason for the observed error is explained in Central 501 HTTPS Required Effective January 15, 2020, The Central Repository no longer supports insecure communication over plain HTTP and requires that all requests to the repository are encrypted over HTTPS. It looks like latest versions of Maven (tried with 3.6.0, 3.6.1) are already using the HTTPS URL by default. Here are the dates when the major repositories will switch: Your Java builds might break starting January 13th (if you haven't yet switched repo access to HTTPS) Update: Seems like from maven 3.2.3 maven central is accessed via HTTPS See <https://stackoverflow.com/a/25411658/5820670> Maven Change log (<http://maven.apache.org/docs/3.2.3/release-notes.html>)

GPT answer, Verb Count: 33 , **Adjective Count:** 17

GPT-4o Answer: This issue arises because Maven Central now enforces the use of HTTPS instead of HTTP. You need to update your Maven configuration to use HTTPS. Follow these steps: 1. Update Maven Settings Modify the `settings.xml` file in your Maven `conf` directory (e.g., `/path/to/apache-maven/conf/settings.xml`) or in your user home directory (e.g., `~/.m2/settings.xml`). Replace any references to `http://repo.maven.apache.org/maven2` with `https://repo.maven.apache.org/maven2`. Example: `<mirrors> <mirror> <id>central</id> <mirrorOf>central</mirrorOf> <url>https://repo.maven.apache.org/maven2</url> </mirror> </mirrors>` 2. Update Maven Version Ensure that you are using a recent Maven version (preferably Maven 3.6.3 or later) as older versions may default to HTTP. To check your Maven version, run: `mvn -v` To update Maven, download the latest version from Apache Maven. 3. Update Project Configuration In your `pom.xml`, replace any repository URLs using HTTP with HTTPS: `<repositories> <repository> <id>central</id> <url>https://repo.maven.apache.org/maven2</url> </repository> </repositories>` 4. Verify Jenkins Configuration If Jenkins uses its own Maven installation, ensure that the Jenkins Maven version is updated and the `settings.xml` used by Jenkins is updated to use HTTPS. To configure Maven in Jenkins: - Go to Manage Jenkins → Global Tool Configuration. - Update or add the Maven installation and point it to the updated Maven version. These steps should resolve the issue and allow your Maven jobs in Jenkins to successfully pull dependencies from Maven Central.

Co-Pilot, Verb Count: 13 , **Adjective Count:** 7

Co-Pilot Answer: To resolve the issue, you need to update your Maven configuration to use HTTPS instead of HTTP for Maven Central. Here's how you can do it: `<!-- Add or update the repository URL in your project's pom.xml --> <repositories> <repository> <id>central</id> <url>https://repo.maven.apache.org/maven2</url> </repository> </repositories>` Alternatively, you can update the `settings.xml` file: `<!-- Add or update the repository URL in your settings.xml file --> <profiles> <profile> <repositories> <repository> <id>central</id> <url>https://repo.maven.apache.org/maven2</url> </repository> </repositories> </profile> </profiles>` Make sure to replace any existing HTTP URLs with HTTPS URLs in your Maven configuration files.

Month: Mar-20

Question 1:

Tags: python, python-poetry

Question: According to poetry's docs, the proper way to setup a new project is with `poetry new poetry-demo`, however this creates a project based on the now deprecated python2.7 by creating the following toml file: `[tool.poetry] name = "poetry-demo" version = "0.1.0" description = "" authors = ["Harsha Goli <harshagoli@gmail.com>"] [tool.poetry.dependencies] python = "^2.7" [tool.poetry.dev-dependencies] pytest = "^4.6" [build-system] requires = ["poetry>=0.12"] build-backend = "poetry.masonry.api"` How can I update this to 3.7? Simply changing `python = "^2.7"` to `python = "^3.7"` results in the following error when `poetry install` is run: `[SolverProblemError] The current project's Python requirement (2.7.17) is not compatible with some of the required packages Python requirement: - zipp requires Python >=3.6 Because no versions of pytest match >=4.6,<4.6.9 || >4.6.9,<5.0 and pytest (4.6.9) depends on importlib-metadata (>=0.12), pytest (>=4.6,<5.0) requires importlib-metadata (>=0.12). And because no versions of importlib-metadata match >=0.12,<1.5.0 || >1.5.0 and importlib-metadata (1.5.0) depends on zipp (>=0.5), pytest (>=4.6,<5.0) requires zipp (>=0.5). Because zipp (3.1.0) requires Python >=3.6 and no versions of zipp match >=0.5,<3.1.0 || >3.1.0, zipp is forbidden. Thus, pytest is forbidden. So, because poetry-demo depends on pytest (^4.6), version solving failed.`

Answer Score: 13

Stack Overflow answer, Verb Count: 17 , **Adjective Count:** 7

Stack Overflow Answer: Interestingly, poetry is silently failing due to a missing package the tool itself relies on and continues to install a broken venv. Here's how you fix it. `sudo apt install python3-venv poetry env remove python3 poetry install` I had to remove pytest, and then reinstall with `poetry add pytest`. EDIT: I ran into this issue again when upgrading a project from python3.7 to python3.8 - for this instead of installing `python3-venv`, you'd want to install `python3.8-venv` instead

GPT answer, Verb Count: 23 , **Adjective Count:** 21

GPT-4o Answer: To update the `pyproject.toml` file to use Python 3.7 and resolve the dependency conflicts, follow these steps: 1. Update the Python Version Change the Python requirement in the `[tool.poetry.dependencies]` section to: `python = "^3.7"` 2. Update the `pytest` Dependency The issue is caused by `pytest` version conflicts. Update the `pytest` dependency to a version compatible with Python 3.7. For example: `pytest = "^6.0"` 3. Update the `pyproject.toml` File Your `pyproject.toml` should look like this: `[tool.poetry] name = "poetry-demo" version = "0.1.0" description = "" authors = ["Harsha Goli "] [tool.poetry.dependencies] python = "^3.7" [tool.poetry.dev-dependencies] pytest = "^6.0" [build-system] requires = ["poetry>=0.12"] build-backend = "poetry.masonry.api"` 4. Reinstall Dependencies Run the following commands to update your dependencies: `poetry lock --no-update poetry install` 5. Verify the Environment Ensure that the virtual environment uses Python 3.7: `poetry env info` If the Python version is incorrect, recreate the environment: `poetry env remove python poetry env use python3.7 poetry install` This should resolve the issue and configure the project to use Python 3.7 successfully.

Co-Pilot, Verb Count: 16 , **Adjective Count:** 7

Co-Pilot Answer: To update your Poetry project to use Python 3.7, you need to change the Python version in the `pyproject.toml` file and ensure compatibility with your dependencies. Here's how you can do it: `[tool.poetry] name = "poetry-demo" version = "0.1.0" description = "" authors = ["Harsha Goli <harshagoli@gmail.com>"] [tool.poetry.dependencies] python = "^3.7" [tool.poetry.dev-dependencies] pytest = "^6.0" [build-system] requires = ["poetry>=0.12"] build-backend =`

"`poetry.masonry.api`" In this updated configuration, I've changed the Python version to `^3.7` and updated the `pytest` version to `^6.0` to ensure compatibility with Python 3.7. After making these changes, run `poetry update` to update your dependencies.

Question 2:

Tags: python, cpython, list, python-3.x, python-internals

Question: Apparently `list(a)` doesn't overallocate, `[x for x in a]` overallocates at some points, and `[*a]` overallocates all the time? Here are sizes `n` from 0 to 12 and the resulting sizes in bytes for the three methods: 0 56 56 56 1 64 88 88 2 72 88 96 3 80 88 104 4 88 88 112 5 96 120 120 6 104 120 128 7 112 120 136 8 120 120 152 9 128 184 184 10 136 184 192 11 144 184 200 12 152 184 208 Computed like this, reproducible at repl.it, using Python 3.8: `from sys import getsizeof for n in range(13): a = [None] * n print(n, getsizeof(list(a)), getsizeof([x for x in a]), getsizeof([*a]))` So: How does this work? How does `[*a]` overallocate? Actually, what mechanism does it use to create the result list from the given input? Does it use an iterator over `a` and use something like `list.append`? Where is the source code? (Colab with data and code that produced the images.) Zooming in to smaller `n`:
Zooming out to larger `n`:

Answer Score: 87

Stack Overflow answer, Verb Count: 40 , **Adjective Count:** 50

Stack Overflow Answer: `[*a]` is internally doing the C equivalent of: Make a new, empty `list` Call `newlist.extend(a)` Returns `list`. So if you expand your test to: `from sys import getsizeof for n in range(13): a = [None] * n l = [] l.extend(a) print(n, getsizeof(list(a)), getsizeof([x for x in a]), getsizeof([*a]), getsizeof(l))` Try it online! you'll see the results for `getsizeof([*a])` and `l = []`; `l.extend(a); getsizeof(l)` are the same. This is usually the right thing to do; when `extending` you're usually expecting to add more later, and similarly for generalized unpacking, it's assumed that multiple things will be added one after the other. `[*a]` is not the normal case; Python assumes there are multiple items or iterables being added to the `list` (`[*a, b, c, *d]`), so overallocation saves work in the common case. By contrast, a `list` constructed from a single, presized iterable (with `list()`) may not grow or shrink during use, and overallocating is premature until proven otherwise; Python recently fixed a bug that made the constructor overallocate even for inputs with known size. As for `list` comprehensions, they're effectively equivalent to repeated `appends`, so you're seeing the final result of the normal overallocation growth pattern when adding an element at a time. To be clear, none of this is a language guarantee. It's just how CPython implements it. The Python language spec is generally unconcerned with specific growth patterns in `list` (aside from guaranteeing amortized `O(1)` `appends` and `pops` from the end). As noted in the comments, the specific implementation changes again in 3.9; while it won't affect `[*a]`, it could affect other cases where what used to be "build a temporary `tuple` of individual items and then `extend` with the `tuple`" now becomes multiple applications of `LIST_APPEND`, which can change when the overallocation occurs and what numbers go into the calculation.

GPT answer, Verb Count: 43 , **Adjective Count:** 27

GPT-4o Answer: The different behaviors in allocation are due to the internal mechanisms of how lists are created and expanded in Python. Here's a detailed explanation of how each method works and why they behave differently: 1. Using `list(a)`: When you create a list from another iterable using `list(a)`, Python allocates memory exactly for the elements in the iterable without any extra space. This is because Python knows the size of the iterable beforehand and allocates the exact amount of memory needed. 2. Using list comprehension `[x for x in a]`: List comprehension may overallocate initially as an optimization strategy. Python anticipates that the list may grow, and pre-allocates extra space to avoid frequent memory reallocations, which can be costly in terms of performance. However,

this overallocation is only done at certain points and not consistently. 3. Using unpacking `[a]`: When using the unpacking syntax `[a]`, Python creates a new list and iterates over the elements of `a`, appending each element to the new list. The list grows dynamically as elements are added, which involves overallocation of memory to accommodate future additions and minimize the number of reallocations. The mechanism for creating a list from an iterable using unpacking involves calling the internal list constructor and appending elements one by one, similar to how `list.append` works. This results in an overallocated list, as the list grows dynamically to handle the elements being added. Here's a reference to the source code where this behavior is implemented: Python's C implementation of list objects, specifically in the `listobject.c` file.

Co-Pilot, Verb Count: 43 , Adjective Count: 27

Co-Pilot Answer: The different behaviors in allocation are due to the internal mechanisms of how lists are created and expanded in Python. Here's a detailed explanation of how each method works and why they behave differently: 1. Using `list(a)`: When you create a list from another iterable using `list(a)`, Python allocates memory exactly for the elements in the iterable without any extra space. This is because Python knows the size of the iterable beforehand and allocates the exact amount of memory needed. 2. Using list comprehension `[x for x in a]`: List comprehension may overallocate initially as an optimization strategy. Python anticipates that the list may grow, and pre-allocates extra space to avoid frequent memory reallocations, which can be costly in terms of performance. However, this overallocation is only done at certain points and not consistently. 3. Using unpacking `[a]`: When using the unpacking syntax `[a]`, Python creates a new list and iterates over the elements of `a`, appending each element to the new list. The list grows dynamically as elements are added, which involves overallocation of memory to accommodate future additions and minimize the number of reallocations. The mechanism for creating a list from an iterable using unpacking involves calling the internal list constructor and appending elements one by one, similar to how `list.append` works. This results in an overallocated list, as the list grows dynamically to handle the elements being added. Here's a reference to the source code where this behavior is implemented: Python's C implementation of list objects, specifically in the `listobject.c` file.

Question 3:

Tags: python, pylint

Question: I am learning to program with python and I am having issues with importing from a module in a package. I am using visual studio code with Python 3.8.2 64 bit. My Project Directory `.vscode`, `ecommerce`, `__init__.py`, `database.py`, `products.py`, `payments`, `__init__.py`, `authorizenet.py`, `paypal.py`, `__init__.py`, `main.py` in the `ecommerce/products.py` file I have: `#products.py from .database import Database p = Database(3,2)` So that I can import the `Database` class from the `ecommerce/database.py` file. But I get error `ImportError : Attempted relative import with no known parent package`

Answer Score: 35

Stack Overflow answer, Verb Count: 41 , Adjective Count: 63

Stack Overflow Answer: It seems, from Python docs and experimenting, that relative imports (involving `.`, `..` etc) only work if the importing module has a `__name__` other than `__main__`, and further, the `__name__` of the importing module is `pkg.module_name`, i.e., it has to be imported from above in the directory hierarchy (to have a parent pkg as part of it's `__name__`.) OR the importing module is being specified via module syntax that includes a parent pkg as `python -m pkg.module`, in which case it's `__name__` is still `__main__`, so it is being run as a script, yet relative imports will work. Here `__package__` is set and used to find the parent package while `__name__` is `__main__`; more here. [After all that, it appears that `__package__` and `sys.path` are key to determining if/how

relative imports work. `__name__` indicates script or module(i.e., `__main__` or `module_name`). `__package__` indicates where in the package the relative imports occur with respect to, and the top of `__package__` needs to be in `sys.path`.] So, continuing with @AmitTendulkar 's example, if you run this as `> python main.py` or `> python -m main` or `> python -m ecommerce.products` from the project root directory, or enter interactive python from that root directory and `import main`, or `import ecommerce.products` the relative imports in `products.py` will work. But if you `> python products.py` or `> python -m products` from within `ecommerce` directory, or enter interactive python from that `ecommerce` directory and `import products` they will fail. It is helpful to add `print("In module products __package__, __name__ =", __package__, __name__)` etc. in each file to debug. UPDATE: How imports work depend on `sys.path` and `__package__`, not on `__name__`. Issued from `/home/jj`, `> python sub/mod.py` has a `sys.path`, `__package__` of `/home/jj/sub`, `None` -absolute imports of modules in `sys.path` work, relative imports fail. `> python -m sub.mod` has `sys.path`, `__package__` of `/home/jj`, `sub` -absolute imports of modules in `sys.path` work, relative imports work relative to `sys.path + __package__`. It is more helpful to add `import sys print("In module products sys.path[0], __package__ =", sys.path[0], __package__)` etc. in each file to debug.

GPT answer, Verb Count: 34 , Adjective Count: 24

GPT-4o Answer: The error occurs because you are attempting a relative import without running the script in the correct package context. To resolve this issue, follow these steps: 1. Correct the Python Execution Context Run the script from the parent directory of your package so that Python recognizes `ecommerce` as a package. Use the following command: `python -m ecommerce.products` This tells Python to treat `ecommerce` as a package and properly resolve the relative import. 2. Use Absolute Imports If you do not want to rely on running the script with the `-m` flag, you can replace the relative import in `ecommerce/products.py` with an absolute import: `from ecommerce.database import Database` This approach avoids the need for the script to be executed in a specific way. 3. Configure VSCode Settings In VSCode, you may encounter similar issues if the interpreter is not set up correctly. Ensure that the Python path is configured to the root of your project. Add the following to your `.vscode/settings.json`: `{ "python.pythonPath": "", "python.analysis.extraPaths": ["./*"] }` 4. Use an Entry Point Script If you want to run `ecommerce/products.py` directly, create an entry point script `main.py` in the root directory with the following content: `from ecommerce.products import some_function some_function()` Then run `main.py` instead of `products.py`. Conclusion Choose between absolute imports or running the script as a module with `-m`. Configuring VSCode settings properly also helps avoid path-related errors.

Co-Pilot, Verb Count: 11 , Adjective Count: 6

Co-Pilot Answer: This issue arises because the Python interpreter is not recognizing the package structure correctly when you run `products.py` directly. To fix this, you should run the `main.py` script, which acts as the entry point for your package, and then import `products.py`. Here's how you can do it: `main.py from ecommerce.products import p` When you run `main.py`, Python will recognize the package structure, and the relative imports will work correctly.

Question 4:

Tags: javascript, reactjs, react-strictmode

Question: I am trying to use a function as a prop inside a component and this component is a child of another component. But the function is not working.? Can I know why. This is the warning i am receiving in the console. Warning: findDOMNode is deprecated in StrictMode. findDOMNode was passed an instance of Transition which is inside StrictMode. Instead, add a ref directly to the element you want to reference This is my code `class Todo extends Component { state = { show: false, editTodo: { id: "", title: "", isCompleted: false } } handleClose =`

```

() => { this.setState({ show: false }) } handleShow = () => {
this.setState({ show: true }) } getStyle () { return { background:
'#f4f4f4', padding: '10px', borderBottom: '1px #ccc dotted', textDecoration:
this.props.todo.isCompleted ? 'line-through' : 'none' } } //this method
checks for changes in the edit field handleChange = (event) => {
this.setState({ title: event.target.value })
console.log(this.state.editTodo.title); } render () { //destructuring const
{ id, title } = this.props.todo; return ( <div style={this.getStyle()}> <p>
<input type='checkbox' style={{ margin: "0px 20px" }}
onChange={this.props.markComplete.bind(this, id)} /> {' '} {title} <Button
style={{ float: "right", margin: "0px 10px" }} variant="warning" size={"sm"}
onClick={this.handleShow}>Edit</Button>{' '} <Button style={{ float: "right"
}} variant="danger" size={"sm"} onClick={this.props.DelItem.bind(this,
id)}>Delete</Button> </p> <Modal show={this.state.show}
onHide={this.handleClose}> <Modal.Header closeButton> <Modal.Title>Edit your
Task!</Modal.Title> </Modal.Header> <Modal.Body > <FormGroup > <Form.Control
type="text" value={this.state.editTodo.title} onChange={this.handleChange}
/> </FormGroup> </Modal.Body> <Modal.Footer> <Button variant="secondary"
onClick={this.handleClose}> Close </Button> <Button variant="primary"
onClick={this.handleClose}> Save Changes </Button> </Modal.Footer> </Modal>
</div> ) } }

```

Answer Score: 28

Stack Overflow answer, Verb Count: 3 , Adjective Count: 5

Stack Overflow Answer: The `setState` call looks like it's being written to the wrong place. Make sure it's on the `editTodo` object: `handleChange = (event) => { this.setState(state => ({ editTodo: { ...state.editTodo, title: event.target.value, }, })); }`

GPT answer, Verb Count: 22 , Adjective Count: 13

GPT-4o Answer: The warning you are seeing is because the `findDOMNode` method is deprecated in React's StrictMode. Instead, you should add a ref directly to the element you want to reference. You can use the `ref` attribute to directly access DOM elements. Here is an updated version of your code:

```

class Todo extends Component { state = { show: false, editTodo: { id: "",
title: "", isCompleted: false } }; handleClose = () => { this.setState({
show: false }); }; handleShow = () => { this.setState({ show: true }); };
getStyle() { return { background: '#f4f4f4', padding: '10px', borderBottom:
'1px #ccc dotted', textDecoration: this.props.todo.isCompleted ?
'line-through' : 'none' }; } handleChange = (event) => { this.setState({
editTodo: { ...this.state.editTodo, title: event.target.value } }); };
render() { const { id, title } = this.props.todo; return ( <div
style={this.getStyle()}> <p> <input type='checkbox' style={{ margin: '0px
20px' }} onChange={this.props.markComplete.bind(this, id)} /> {' '} {title}
<Button style={{ float: 'right', margin: '0px 10px' }} variant='warning'
size='sm' onClick={this.handleShow}>Edit</Button> <Button style={{ float:
'right' }} variant='danger' size='sm' onClick={this.props.DelItem.bind(this,
id)}>Delete</Button> </p> <Modal show={this.state.show}
onHide={this.handleClose}> <Modal.Header closeButton> <Modal.Title>Edit your
Task!</Modal.Title> </Modal.Header> <Modal.Body> <FormGroup> <Form.Control
type='text' value={this.state.editTodo.title} onChange={this.handleChange}
/> </FormGroup> </Modal.Body> <Modal.Footer> <Button variant='secondary'
onClick={this.handleClose}> Close </Button> <Button variant='primary'
onClick={this.handleClose}> Save Changes </Button> </Modal.Footer> </Modal>

```

</div>); } } By updating your code to use the `ref` attribute, you can avoid the deprecated `findDOMNode` method and resolve the warning.

Co-Pilot, Verb Count: 33 , Adjective Count: 31

Co-Pilot Answer: ``html The warning occurs because `findDOMNode` is being used by the `Modal` component from `react-bootstrap`, which is deprecated in `StrictMode`. To resolve this, you should update your code to use refs instead of relying on `findDOMNode`. Here are the steps to fix this issue: 1. Update React-Bootstrap Version Ensure you are using the latest version of `react-bootstrap`. Newer versions have better support for `StrictMode`. Update the package by running: `npm install react-bootstrap` 2. Replace Deprecated Usage The `Modal` component and its related functionality might internally use `findDOMNode`. To avoid this, consider replacing the `Modal` with a custom implementation or ensure proper usage of refs as required. 3. Fix the `value` Issue in Your Code In your code, the `value` of the input inside the `Modal` references `this.state.editTodo.title`, but this field is not updated correctly due to the lack of mutation of `editTodo` in `handleChange`. Update your `handleChange` method as follows: `handleChange = (event) => { this.setState((prevState) => ({ editTodo: { ...prevState.editTodo, title: event.target.value, }, })); };` 4. Correct `Modal` Implementation If the `Modal` is causing the issue, you may wrap it with a parent component and forward refs properly. An example: `import React, { forwardRef } from 'react'; import { Modal } from 'react-bootstrap'; const ForwardRefModal = forwardRef((props, ref) => ({props.children}));` Then, replace your `Modal` usage with `ForwardRefModal`. 5. Test Your Changes After implementing the above changes, ensure your app works without warnings in `StrictMode`. ``

Question 5:

Tags: python, fastapi, uvicorn

Question: I tried to run FastAPI using uvicorn webserver but it throws an error. I run this command, `uvicorn api:app --reload --host 0.0.0.0` but there is an error in the terminal. `Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit) Started reloader process [23445] Error loading ASGI app. Could not import module "api". Stopping reloader process [23445]`

Answer Score: 302

Stack Overflow answer, Verb Count: 33 , Adjective Count: 22

Stack Overflow Answer: TL;DR Add the directory name in front of your filename `uvicorn src.main:app` or `cd` into that directory `cd src uvicorn main:app` Long Answer It happens because you are not in the same folder with your FastAPI app instance more specifically: Let's say i have an app-tree like this; `my_fastapi_app/ ,îú,îÄ,îÄ app.yaml ,îú,îÄ,îÄ docker-compose.yml ,îú,îÄ,îÄ src ,îÇ¬¬¬ ,îî,îÄ,îÄ main.py ,îî,îÄ,îÄ tests ,îú,îÄ,îÄ test_xx.py ,îî,îÄ,îÄ test_yy.py $ pwd # Present Working Directory /home/yagiz/Desktop/my_fastapi_app` I'm not inside the same folder with my app instance, so if I try to run my app with uvicorn I'll get an error like yours `$ uvicorn main:app --reload INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit) INFO: Started reloader process [40645] using statreload ERROR: Error loading ASGI app. Could not import module "main".` The answer is so simple, add the folder name in front of your filename `uvicorn src.main:app --reload` or you can change your working directory `cd src` Now i'm inside of the folder with my app instance `src ,îî,îÄ,îÄ main.py` Run your uvicorn again `$ uvicorn main:app --reload INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit) INFO: Started reloader process [40726] using statreload INFO: Started server process [40728] INFO: Waiting for application startup. INFO: Application startup complete.`

GPT answer, Verb Count: 30 , Adjective Count: 27

GPT-4o Answer: The error indicates that `uvicorn` is unable to locate the `api` module or the `app` object within it. Here are the possible reasons and solutions: 1. Verify Module and Application Name Ensure that the file containing your FastAPI app is named `api.py` and the FastAPI instance is defined as `app` within that file: `# api.py from fastapi import FastAPI app = FastAPI() @app.get("/") def read_root(): return {"Hello": "World"}` 2. Check the Current Working Directory Run the `uvicorn` command from the directory where `api.py` is located. If `api.py` is inside a folder, navigate to that folder first: `cd /path/to/folder uvicorn api:app --reload --host 0.0.0.0` 3. Check for Typos Ensure there are no typos in the command. It should exactly match the module and variable names: `uvicorn api:app --reload --host 0.0.0.0` 4. Python Path Issue If `api.py` is in a subdirectory (e.g., `myproject/api.py`), use the module import path: `uvicorn myproject.api:app --reload --host 0.0.0.0` 5. Missing Dependencies Ensure that FastAPI and Uvicorn are installed in your environment. Install them using: `pip install fastapi uvicorn` 6. Debugging Imports Run the following Python command to verify that `api` and `app` can be imported without issues: `python -c "from api import app; print(app)"` If this fails, double-check the module and app definitions.

Co-Pilot, Verb Count: 19 , Adjective Count: 16

Co-Pilot Answer: The error you're encountering suggests that Uvicorn cannot find the module named "api". This can occur if the module is not in the correct directory or there is a naming issue. Here are some steps to resolve the issue: 1. Ensure that the `api.py` file is in the same directory from which you are running the `uvicorn` command. The structure should look like this: `/your_project_directory/ api.py` 2. Verify that the `app` instance is defined in the `api.py` file: `from fastapi import FastAPI app = FastAPI() @app.get("/") def read_root(): return {"Hello": "World"}` 3. Run the command from the same directory as `api.py` or provide the full path to the module. If `api.py` is inside a folder, modify the command accordingly: `uvicorn folder_name.api:app --reload --host 0.0.0.0` Following these steps should resolve the module import error.

Month: Apr-20

Question 1:

Tags: java, android, kotlin, android-handler

Question: How do I fix the deprecation warning in this code? Alternatively, are there any other options for doing this? `Handler().postDelayed({ context?.let { //code } }, 3000)`

Answer Score: 715

Stack Overflow answer, Verb Count: 6 , Adjective Count: 8

Stack Overflow Answer: Only the parameterless constructor is deprecated, it is now preferred that you specify the `Looper` in the constructor via the `Looper.getMainLooper()` method. Use it for Java `new Handler(Looper.getMainLooper()).postDelayed(new Runnable() { @Override public void run() { // Your Code } }, 3000);` Use it for Kotlin `Handler(Looper.getMainLooper()).postDelayed({ // Your Code }, 3000)` Source : developer.android.com

GPT answer, Verb Count: 13 , Adjective Count: 19

GPT-4o Answer: The use of `Handler().postDelayed` is deprecated in newer versions of Android. You can use `Handler(Looper.getMainLooper())` or `Handler(Looper.myLooper())`, or better yet, use `Handler.postDelayed` with the `Handler(Looper.getMainLooper())` instance for main thread operations. Alternatively, you can use `Handler` from Kotlin's coroutines with

`Dispatchers.Main`. Here's how you can fix it: Option 1: Using `Handler` with `Looper`
`Handler(Looper.getMainLooper()).postDelayed({ context?.let { //code } }, 3000)` Option 2: Using Kotlin Coroutines Switch to coroutines for better structure and modern practices: `import kotlinx.coroutines.* GlobalScope.launch(Dispatchers.Main) { delay(3000) context?.let { //code } }` Option 2 is the preferred method for modern Android development as it avoids using the deprecated `Handler` API and improves readability and maintainability.

Co-Pilot, Verb Count: 4 , Adjective Count: 3

Co-Pilot Answer: To fix the deprecation warning for `Handler().postDelayed`, you can use `Handler(Looper.getMainLooper()).postDelayed`. Here is the updated code:
`Handler(Looper.getMainLooper()).postDelayed({ context?.let { //code } }, 3000)`

Question 2:

Tags: python, python-3.x, pathlib

Question: I am currently accessing the parent directory of my file using `Pathlib` as follows:
`Path(__file__).parent` When I print it, and this gives me the following output: `print('Parent:', Path(__file__).parent)` #output `/home/user/EC/main-folder` The `main-folder` has a `.env` file which I want to access and for that I want to join the parent path with the `.env`. Right now, I did: `dotenv_path = os.path.join(Path(__file__).parent, ".env")` which works. But I would like to know, if there is a `Pathlib` alternate to `os.path.join()`? Something like:
`dotenv_path = pathlib_alterate_for_join(Path(__file__).parent, ".env")`

Answer Score: 78

Stack Overflow answer, Verb Count: 1 , Adjective Count: 2

Stack Overflow Answer: Use `pathlib.Path.joinpath`:
`(Path(__file__).parent).joinpath('.env')`

GPT answer, Verb Count: 6 , Adjective Count: 9

GPT-4o Answer: Yes, there is a `Pathlib` alternative to `os.path.join()`. You can use the `/` operator provided by `pathlib` for path concatenation. Here's how you can do it: `from pathlib import Path dotenv_path = Path(__file__).parent / ".env" print(dotenv_path)` The `/` operator is overloaded in `pathlib` to concatenate paths in a clean and Pythonic way.

Co-Pilot, Verb Count: 6 , Adjective Count: 5

Co-Pilot Answer: You can use the `Pathlib` module to join paths directly with the `/` operator. Here's how you can do it: `from pathlib import Path dotenv_path = Path(file).parent / ".env"` This method is more idiomatic and leverages the capabilities of the `Pathlib` module to handle file paths.

Question 3:

Tags: javascript, regex

Question: I stumbled across a surprising (to me) fact. `console.log("asdf".replace(/./g, "x"))`; Why two replacements? It seems any non-empty string without newlines will produce exactly two replacements for this pattern. Using a replacement function, I can see that the first replacement is for the entire string, and the second is for an empty string.

Answer Score: 110

Stack Overflow answer, Verb Count: 23 , Adjective Count: 30

Stack Overflow Answer: As per the ECMA-262 standard, String.prototype.replace calls RegExp.prototype[@@replace], which says: 11. Repeat, while done is false a. Let result be ? RegExpExec(rx, S). b. If result is null, set done to true. c. Else result is not null, i. Append result to the end of results. ii. If global is false, set done to true. iii. Else, 1. Let matchStr be ? ToString(? Get(result, "0")). 2. If matchStr is the empty String, then a. Let thisIndex be ? ToLength(? Get(rx, "lastIndex")). b. Let nextIndex be AdvanceStringIndex(S, thisIndex, fullUnicode). c. Perform ? Set(rx, "lastIndex", nextIndex, true). where rx is /.*/g and S is 'asdf'. See 11.c.iii.2.b: b. Let nextIndex be AdvanceStringIndex(S, thisIndex, fullUnicode). Therefore in 'asdf'.replace(/.*/g, 'x') it is actually: result (undefined), results = [], lastIndex = 0 result = 'asdf', results = ['asdf'], lastIndex = 4 result = '', results = ['asdf', ''], lastIndex = 4, AdvanceStringIndex, set lastIndex to 5 result = null, results = ['asdf', ''], return Therefore there are 2 matches.

GPT answer, Verb Count: 13 , Adjective Count: 14

GPT-4o Answer: This behavior occurs because the pattern /.*/g matches two parts in the string: "asdf".replace(/.*/g, (match, offset) => { console.log(`Matched: '\${match}' at offset \${offset}`); return "x"; }); The output is: Matched: 'asdf' at offset 0 Matched: '' at offset 4 Explanation: 1. The regex /.*/ with the g (global) flag matches zero or more of any character (except newlines). 2. The first match is the entire string "asdf". 3. After consuming the string, the regex also matches an empty string at the end because .* can match zero characters. This is an edge case in regex matching. As a result, two replacements occur: one for the entire string and one for the empty string at the end.

Co-Pilot, Verb Count: 15 , Adjective Count: 13

Co-Pilot Answer: The reason you see two replacements is due to the behavior of the regular expression /.*/ in JavaScript, which matches zero or more of any character except newlines. Here's what happens step-by-step: The pattern /.*/ matches the entire string "asdf" first and replaces it with "x". After matching the entire string, /.*/ then matches an empty string at the end of the input (because it can match zero characters) and replaces it with another "x". To see this more clearly, you can use a replacement function that logs each replacement: console.log("asdf".replace(/.*/g, (match) => { console.log("Matched:", match); return "x"; })); This will output: Matched: asdf Matched: This illustrates the two matches: one for the entire string and one for the empty string at the end.

Question 4:

Tags: html, javascript, debugging, google-chrome-extension

Question: I'm trying to display an image selected from the local machine and I need the location of that image for a JavaScript function. But I'm unable to get the location. To get the image location, I tried using console.log, but nothing returns.

```
console.log(document.getElementById("uploadPreview")); Here's the HTML code:
<!DOCTYPE html> <html> <head> <title></title> </head> <body> <div
align="center" style="padding-top: 50px"> <img align="center"
id="uploadPreview" style="width: 100px; height: 100px;" /> </div> <div
align="center" style="padding-left: 30px"> <input id="uploadImage"
type="file" name="myPhoto" onchange="PreviewImage();" /> </div> <script
type="text/javascript"> function PreviewImage() { var oFReader = new
FileReader();
oFReader.readAsDataURL(document.getElementById("uploadImage").files[0]);
```

```
oFReader.onload = function (oFREvent) {
document.getElementById("uploadPreview").src = oFREvent.target.result;
console.log(document.getElementById("uploadPreview").src); }; } </script>
</body> </html>
```

Console Output: Here's the warning: DevTools failed to load SourceMap: Could not load content for chrome-extension://alplpnakfeabeieibipdmaenpmbgknjce/include.preload.js.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME

Answer Score: 479

Stack Overflow answer, Verb Count: 14 , **Adjective Count:** 4

Stack Overflow Answer: That's because Chrome added support for source maps. Go to the developer tools (F12 in the browser), then select the three dots in the upper right corner, and go to Settings. Then, look for Sources, and disable the options: "Enable JavaScript source maps" "Enable CSS source maps" If you do that, that would get rid of the warnings. It has nothing to do with your code. Check the developer tools in other pages and you will see the same warning.

GPT answer, Verb Count: 27 , **Adjective Count:** 16

GPT-4o Answer: The issue you are encountering is because of browser security policies. When you use the `FileReader` API, it reads the file content as a data URL (a base64-encoded string representing the file), not its local file path. Browsers restrict direct access to local file paths for security reasons. Your function is correctly using the `FileReader` API to read the file and display it in the `img` element. The `src` attribute of the image will contain the base64 data, not the file's local path. If you need to display the image, the provided code is working as intended. Here's the corrected code:

```
function PreviewImage() { var oFReader = new FileReader();
oFReader.readAsDataURL(document.getElementById("uploadImage").files[0]);
oFReader.onload = function (oFREvent) {
document.getElementById("uploadPreview").src = oFREvent.target.result;
console.log(document.getElementById("uploadPreview").src); // Logs the
base64 data URL }; } If you still want to retrieve the local file path, you cannot do so due to
browser restrictions. However, the base64 data URL can be used for most purposes, including
displaying the image or uploading it to a server.
```

Co-Pilot, Verb Count: 11 , **Adjective Count:** 6

Co-Pilot Answer: To display the image selected from the local machine and obtain its location, you need to modify your JavaScript function to correctly handle the `FileReader`. Here's how you can do it:

```
<!DOCTYPE html> <html> <head> <title></title> </head> <body> <div
align="center" style="padding-top: 50px"> <img align="center"
id="uploadPreview" style="width: 100px; height: 100px;" /> </div> <div
align="center" style="padding-left: 30px"> <input id="uploadImage"
type="file" name="myPhoto" onchange="PreviewImage();" /> </div> <script
type="text/javascript"> function PreviewImage() { var oFReader = new
FileReader();
oFReader.readAsDataURL(document.getElementById("uploadImage").files[0]);
oFReader.onload = function (oFREvent) {
document.getElementById("uploadPreview").src = oFREvent.target.result;
console.log(document.getElementById("uploadPreview").src); }; } </script>
</body> </html>
```

Question 5:

Tags: javascript, logical-or, nullish-coalescing

Question: Related to Is there a "null coalescing" operator in JavaScript? - JavaScript now has a `??` operator which I see in use more frequently. Previously most JavaScript code used `||`. `let userAge`

`= null // These values will be the same. let age1 = userAge || 21 let age2 = userAge ?? 21` In what circumstances will `??` and `||` behave differently?

Answer Score: 850

Stack Overflow answer, Verb Count: 46 , **Adjective Count:** 47

Stack Overflow Answer: The OR operator `||` uses the right value if left is falsy, while the nullish coalescing operator `??` uses the right value if left is `null` or `undefined`. These operators are often used to provide a default value if the first one is missing. But the OR operator `||` can be problematic if your left value might contain `"` or `0` or `false` (because these are falsy values): `console.log(12 || "not found") // 12 console.log(0 || "not found") // "not found" console.log("jane" || "not found") // "jane" console.log("" || "not found") // "not found" console.log(true || "not found") // true console.log(false || "not found") // "not found" console.log(undefined || "not found") // "not found" console.log(null || "not found") // "not found"` In many cases, you might only want the right value if left is `null` or `undefined`. That's what the nullish coalescing operator `??` is for: `console.log(12 ?? "not found") // 12 console.log(0 ?? "not found") // 0 console.log("jane" ?? "not found") // "jane" console.log("" ?? "not found") // "" console.log(true ?? "not found") // true console.log(false ?? "not found") // false console.log(undefined ?? "not found") // "not found" console.log(null ?? "not found") // "not found"` While the `??` operator isn't available in current LTS versions of Node (v10 and v12), you can use it with some versions of TypeScript or Node: The `??` operator was added to TypeScript 3.7 back in November 2019. And more recently, the `??` operator was included in ES2020, which is supported by Node 14 (released in April 2020). When the nullish coalescing operator `??` is supported, I typically use it instead of the OR operator `||` (unless there's a good reason not to).

GPT answer, Verb Count: 28 , **Adjective Count:** 50

GPT-4o Answer: The `??` (nullish coalescing operator) and `||` (logical OR operator) behave differently in how they treat "falsy" values. Here's the key difference: `||` considers all falsy values as needing replacement. Falsy values include `undefined`, `null`, `false`, `0`, `NaN`, and an empty string `"`. `??` considers only `null` and `undefined` as needing replacement. Other falsy values, like `0`, `false`, `NaN`, or `"`, are retained. Here's an example: `let value1 = 0; let result1 = value1 || 42; // 42 (because 0 is falsy with ||) let result2 = value1 ?? 42; // 0 (because ?? only replaces null/undefined) let value2 = null; let result3 = value2 || 42; // 42 (null is falsy) let result4 = value2 ?? 42; // 42 (null is nullish) let value3 = undefined; let result5 = value3 || 42; // 42 (undefined is falsy) let result6 = value3 ?? 42; // 42 (undefined is nullish) let value4 = ""; let result7 = value4 || "default"; // "default" (empty string is falsy) let result8 = value4 ?? "default"; // "" (empty string is not nullish)` Summary: Use `??` when you want to replace only `null` or `undefined` and retain other falsy values like `0`, `false`, or `"`. Use `||` when you want to replace all falsy values.

Co-Pilot, Verb Count: 8 , **Adjective Count:** 41

Co-Pilot Answer: The `??` (nullish coalescing) operator and the `||` (logical OR) operator behave differently in how they handle certain falsy values. Here's a detailed explanation: The `||` operator returns the right-hand side value if the left-hand side value is any falsy value (like `null`, `undefined`, `0`, `NaN`, `"`, or `false`). The `??` operator returns the right-hand side value only if the left-hand side value is `null` or `undefined`. Here's a comparison with examples: `let userAge = 0; Using ||: let age1 = userAge || 21; // age1 will be 21 Using ??: let age2 = userAge ?? 21; // age2 will be 0` In this example, `userAge` is `0`, which is a falsy value. The `||` operator treats `0` as falsy and returns `21`, while the `??` operator only considers `null` or `undefined` as nullish values and thus returns `0`.

Month: May-20

Question 1:

Tags: python, ubuntu, ubuntu-20.04

Question: I recently upgraded my OS to Ubuntu 20.04 LTS. Now when I try to import a library like Numpy in Python, I get the following error: `ImportError: libffi.so.6: cannot open shared object file: No such file or directory` I tried installing the `libffi` package, but apt can't locate it: `sudo apt-get install libffi` Reading package lists... Done Building dependency tree Reading state information... Done E: Unable to locate package libffi

Answer Score: 113

Stack Overflow answer, Verb Count: 22 , Adjective Count: 10

Stack Overflow Answer: It seems like I fixed it. I could be wrong, but here is what I think happened: Ubuntu 20.04 upgraded libffi6 to libffi7 Python is still looking for libffi6 What I did to fix it : Locate `libffi.so.7` in your system `$ find /usr/lib -name "libffi.so*"` Create a simlink named `libffi.so.6` that points to `libffi.so.7`: `sudo ln -s /usr/path/to/libffi.so.7 /usr/lib/path/to/libffi.so.6` UPDATE: As noted by many users, this fix could have unintended consequences. The better way to do it is to reinstall python as @amichaud explained. This should be used as a last resort IF you're not using pyenv/virtualenv/etc in which case removing python will cause a lot of dependencies to be removed as well.

GPT answer, Verb Count: 18 , Adjective Count: 16

GPT-4o Answer: The error occurs because Ubuntu 20.04 LTS uses a newer version of the `libffi` library, but some Python packages or applications might still depend on `libffi.so.6`, which is not included by default. To resolve this issue, you can manually install the older version of the library alongside the newer one. Follow these steps: `sudo apt-get update sudo apt-get install libffi-dev` If this does not resolve the issue (as `libffi.so.6` might still be missing), download and install the specific version: `wget http://archive.ubuntu.com/ubuntu/pool/main/libf/libffi/libffi6_3.2.1-8_amd64.deb sudo dpkg -i libffi6_3.2.1-8_amd64.deb` This will add `libffi.so.6` to your system while retaining the compatibility of other libraries. After installation, try importing your library in Python again.

Co-Pilot, Verb Count: 15 , Adjective Count: 6

Co-Pilot Answer: To resolve the `ImportError: libffi.so.6: cannot open shared object file: No such file or directory` on Ubuntu 20.04 LTS, you can try the following steps: 1. Reinstall the `libffi6` package: `sudo apt-get install --reinstall libffi6` 2. Create a symbolic link to `libffi.so.7`: `sudo ln -s /usr/lib/x86_64-linux-gnu/libffi.so.7 /usr/lib/x86_64-linux-gnu/libffi.so.6` 3. Install the `libffi-dev` package: `sudo apt-get install libffi-dev` After trying these steps, try importing Numpy again to see if the issue is resolved.

Question 2:

Tags: html, javascript, node.js, typescript, vue.js

Question: I wanted to set title to my webpage created with vue cli 3 and thus looked into `public/index.html`. There, I found `<title><%= htmlWebpackPlugin.options.title %></title>`. How do I set and modify `htmlWebpackPlugin.options.title` in vue cli 3 project?

Answer Score: 201

Stack Overflow answer, Verb Count: 80 , Adjective Count: 70

Stack Overflow Answer: Looking at the popularity of the question, I decided to add an elaborate answer with references to make it more authentic and complete. I have also created an article on this topic and covered this topic in this and this courses. Though the question is looking for setting `htmlWebpackPlugin.options.title`, the ultimate effect is changing the title of the web-page. 1. Most convenient and trivial solution The simplest way to do this is to modify the `public/index.html` and hard-code the title.

```
<!DOCTYPE html> <html lang=""> <head> <meta charset="utf-8"> <meta http-equiv="X-UA-Compatible" content="IE=edge"> <meta name="viewport" content="width=device-width,initial-scale=1.0"> <link rel="icon" href="<%= BASE_URL %>favicon.ico"> <title> <%= htmlWebpackPlugin.options.title %> </title> </head> <body> <noscript> <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable it to continue.</strong> </noscript> <div id="app"></div> <!-- built files will be auto injected --> </body> </html>
```

 This is the default `public/index.html` that is generated by vue cli. And in this, you just need to change `<title> <%= htmlWebpackPlugin.options.title %> </title>` to `<title>Title of your choice</title>` 2. Change the name field in `package.json` Another simple solution is to change the `"name": "your-project-name"`. However, there are many restrictions on the name you can use in `package.json`. You can read more about this here. Basically, `package.json` must contain a name and that must be lowercase and one word, and may contain hyphens and underscores. 3. Using pages field in `vue.config.js` `vue.config.js` is an optional file that you can add to provide additional configurations to Vue CLI and this file, if present, will be automatically loaded by Vue CLI. You need to create `vue.config.js` in the root folder - the folder containing you `package.json` file. According to Vue documentation, you can use pages field to define entrypoint for multi-page app. However, you can also use this to define title for single page app as well. Create `vue.config.js` in the root directory and add pages field to your exports as follows:

```
module.exports = { pages: { index: { // entry for the page entry: 'src/main.js', title: 'My Title', }, } }
```

 Note that if you are already running development server, this change will be reflected only when you stop and restart the development server. In other words, these changes will not be hot reloaded. 4. Chaining Webpack You can chain Webpack in `vue.config.js` as shown below

```
module.exports = { chainWebpack: config => { config.plugin('html').tap(args => { args[0].title = "My Vue App"; return args; }) } }
```

 Note that similar to solution 3, this change will be reflected only when you stop and restart the development server, in case you are already running development server. In other words, these changes will not be hot reloaded. 5. Modify title in lifecycle hooks using JavaScript The next solution in the list is to use JavaScript to modify the title. You can do this either in `mounted` lifecycle hook of your root component or if you want different title for different routes, you can do the same for components loaded by each route.

```
<script> export default { data() { return { // }; }, mounted() { document.title = 'new title' } } </script>
```

 6. Use Vue Meta Finally you can use Vue Meta to manage all metadata for your Vue app including title. First you need to add Vue Meta to your project and then use `metaInfo` field as shown below to configure metadata for your page or route.

```
{ metaInfo: { meta: [ { charset: 'utf-8' }, { name: 'viewport', content: 'width=device-width, initial-scale=1' }, { title: 'My title' } ] } }
```

 Conclusion The first 4 solutions are static ways of changing your title or in other words you can't change your title at runtime using these ways. Also all of these are not hot reloaded. The last 2 options use JavaScript and can manipulate the title at runtime.

GPT answer, Verb Count: 16 , Adjective Count: 4

GPT-4o Answer: To set and modify `htmlWebpackPlugin.options.title` in a Vue CLI 3 project, follow these steps: 1. Open the `vue.config.js` file in the root of your project (create it if it doesn't exist).

```
module.exports = { chainWebpack: config => { config.plugin('html').tap(args => { args[0].title = 'Your Custom Title';
```

`return args; }); } };` 2. Replace 'Your Custom Title' with the desired title for your webpage. 3. Save the file and restart the development server if it's running. This will set the title dynamically in `public/index.html` based on the value provided in `vue.config.js`.

Co-Pilot, Verb Count: 7 , Adjective Count: 5

Co-Pilot Answer: To set and modify `htmlWebpackPlugin.options.title` in a Vue CLI 3 project, you need to configure the `vue.config.js` file. Here's how you can do it: `module.exports = { configureWebpack: { plugins: [new HtmlWebpackPlugin({ title: 'Your Page Title' })] } };` By adding this configuration to `vue.config.js`, you can set the title for your webpage.

Question 3:

Tags: javascript, google-chrome, cjk

Question: If you double-click English text in Chrome, the whitespace-delimited word you clicked on is highlighted. This is not surprising. However, the other day I was clicking while reading some text in Japanese and noticed that some words were highlighted at word boundaries, even though Japanese doesn't have spaces. Here's some example text: „À©„Àì„ÀßÁîü„Çà„Äü„Äã„À®„Çì„À®Ë¶ãÄΩì„Äâ„Ä§„Äã„Ä~„ÄÇ%Ωì„Äß„ÇÇËñÑÊöö„ÄÑ„Äð„ÇÄ„Äð„ÇÄ„Äó„ÄüÊâÄ„Äß„Éã„É£„É°„Éã„É£„É°Ê≥£„ÄÑ„Ä¶„Äñ„Äü% [ã„Ä†„Äë„ÄØË®ðÊÜð„Äó„Ä¶„Äñ„Çã„ÄÇ')) // [" „À©„Àì" , " „Àß" , " „Áîü„Çà" , " „Äü„Äã" , " „À®„Çì„À®" , " „Ë¶ãÄΩì" , " „Äâ" , " „Ä§" , " „Äã" , " „Ä~" , " „ÄÇ" , " „%Ωì„Äß„ÇÇ" , " „ËñÑÊöö„ÄÑ" , " „Äð„ÇÄ„Äð„ÇÄ" , " „Äó„Äü" , " „ÊâÄ" , " „Äß" , " „Éã„É£„É°„Éã„É£„É°" , " „Ê≥£" , " „ÄÑ" , " „Ä¶" , " „Äñ„Äü% [ã" , " „Ä†„Äë" , " „ÄØ" , " „Ë®ðÊÜð" , " „Äó" , " „Ä¶" , " „Äñ„Çã" , " „ÄÇ"] I also made a jsfiddle that shows this. The quality is not amazing but I'm surprised this is supported at all.

Answer Score: 170

Stack Overflow answer, Verb Count: 10 , Adjective Count: 11

Stack Overflow Answer: So it turns out v8 has a non-standard multi-language word segmenter and it handles Japanese. `function tokenizeJA(text) { var it = Intl.v8BreakIterator(['ja-JP'], {type:'word'}) it.adoptText(text) var words = [] var cur = 0, prev = 0 while (cur < text.length) { prev = cur cur = it.next() words.push(text.substring(prev, cur)) } return words } console.log (tokenizeJA(' „À©„Àì„ÀßÁîü„Çà„Äü„Äã„À®„Çì„À®Ë¶ãÄΩì„Äâ„Ä§„Äã„Ä~„ÄÇ%Ωì„Äß„ÇÇËñÑÊöö„ÄÑ„Äð„ÇÄ„Äð„ÇÄ„Äó„ÄüÊâÄ„Äß„Éã„É£„É°„Éã„É£„É°Ê≥£„ÄÑ„Ä¶„Äñ„Äü% [ã„Ä†„Äë„ÄØË®ðÊÜð„Äó„Ä¶„Äñ„Çã„ÄÇ')) // [" „À©„Àì" , " „Àß" , " „Áîü„Çà" , " „Äü„Äã" , " „À®„Çì„À®" , " „Ë¶ãÄΩì" , " „Äâ" , " „Ä§" , " „Äã" , " „Ä~" , " „ÄÇ" , " „%Ωì„Äß„ÇÇ" , " „ËñÑÊöö„ÄÑ" , " „Äð„ÇÄ„Äð„ÇÄ" , " „Äó„Äü" , " „ÊâÄ" , " „Äß" , " „Éã„É£„É°„Éã„É£„É°" , " „Ê≥£" , " „ÄÑ" , " „Ä¶" , " „Äñ„Äü% [ã" , " „Ä†„Äë" , " „ÄØ" , " „Ë®ðÊÜð" , " „Äó" , " „Ä¶" , " „Äñ„Çã" , " „ÄÇ"]` I also made a jsfiddle that shows this. The quality is not amazing but I'm surprised this is supported at all.

GPT answer, Verb Count: 34 , Adjective Count: 16

GPT-4o Answer: Chrome determines how to highlight Japanese text using Unicode Text Segmentation rules, which are part of the Unicode Standard. Specifically, Chrome relies on the Word Boundary rules defined in Unicode Standard Annex #29 (UAX #29), titled "Unicode Text Segmentation". These rules specify how to determine word boundaries in languages without explicit spaces, like Japanese. Key points about how Chrome likely implements this: 1. **Grapheme Cluster:** Chrome first identifies grapheme clusters, which are user-perceived characters. For example, combining characters (like diacritics) are grouped with their base character. 2. **Word Boundary Rules:** Unicode defines rules for determining word boundaries based on character classes. For Japanese, this

Co-Pilot, Verb Count: 17 , Adjective Count: 7

Question 4:

Question: I'm trying to import `myArr` from `hello.js` into `index.js`. However I get an error of `Error [ERR_REQUIRE_ESM]: Must use import to load ES Module File hello.js export let myArr = ['hello', 'hi', 'hey']; File index.js import { myArr } from './hello.js'; console.log(myArr);` Where am I going wrong?

Stack Overflow answer, Verb Count: 8 , Adjective Count: 4

GPT answer, Verb Count: 15 , Adjective Count: 8

Co-Pilot, Verb Count: 19 , Adjective Count: 12

Co-Pilot Answer: The error you're encountering indicates that your environment expects the module format to be ES Module (ESM), but it is not set up to handle this. Here's how you can resolve it: 1. Ensure that your Node.js environment is configured to use ES Modules. You can do this by adding a `"type": "module"` field in your `package.json` file: `{ "type": "module" }` 2. Alternatively, if you don't want to change the entire project to use ES Modules, you can rename your files with the `.mjs` extension. This signals to Node.js that these are ES Module files. Rename `hello.js` to `hello.mjs` and `index.js` to `index.mjs` and then run: `import { myArr } from './hello.mjs'; console.log(myArr);` These steps will configure your project to correctly handle ES Modules and resolve the import error.

Question 5:

Tags: java, dart, android-studio, flutter, flutter-doctor

Question: Exception in thread "main" java.lang.NoClassDefFoundError: javax.xml.bind.annotation.XmlSchema at com.android.repository.api.SchemaModule\$SchemaModuleVersion.<init>(SchemaModule.java:156) at com.android.repository.api.SchemaModule.<init>(SchemaModule.java:75) at com.android.sdklib.repository.AndroidSdkHandler.<clinit>(AndroidSdkHandler.java:81) at com.android.sdklib.tool.sdkmanager.SdkManagerCli.main(SdkManagerCli.java:73) at com.android.sdklib.tool.sdkmanager.SdkManagerCli.main(SdkManagerCli.java:48) Caused by: java.lang.ClassNotFoundException: javax.xml.bind.annotation.XmlSchema at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:581) at java.base/jdk.internal.loader.ClassLoaders\$AppClassLoader.loadClass(ClassLoaders.java:178) at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521) ... 5 more I get this error above when running flutter doctor --android-licenses. Does anyone know why this is happening and how I can fix it? I'm already running it on the flutter console.

Answer Score: 724

Stack Overflow answer, Verb Count: 10 , **Adjective Count:** 6

Stack Overflow Answer: Installing Android SDK Command-line tools from Android SDK Manager did the job for me. Open Tools > SDK Manager From the left choose, Appearance & Behavior > System Settings > Android SDK Select SDK Tools from the top menu Check Android SDK Command-line tools and click 'apply'. Alternative to step #1: WINDOWS: File > Settings (Ctrl+Alt+S) / MAC : Android Studio > Preferences This fix works for Windows, Linux and Mac OS X. If you are facing a problem in spite of the above steps, you can try upgrading your system to JDK 8. Official documentation of upgrading to JDK 8 is available here.

GPT answer, Verb Count: 31 , **Adjective Count:** 14

GPT-4o Answer: The error occurs because the `javax.xml.bind.annotation.XmlSchema` class, which is part of the Java XML Binding (JAXB) module, has been removed from Java starting from version 9. This issue happens when tools or libraries depend on JAXB and the application is running with Java 9 or later. To fix this, you need to add the JAXB module explicitly to your Java environment. Here are the steps: 1. ****Install the JAXB API**** If it's not already installed, add the JAXB API by running the following command: `sdkmanager --install "extras;android;m2repository"` 2. ****Add JAXB dependencies**** Add the following dependencies to the command you use to run the tool: `--add-modules java.xml.bind --add-exports java.xml.bind/com.sun.xml.internal.bind=ALL-UNNAMED` 3. ****Use an earlier version of Java**** Alternatively, you can use Java 8 (which includes JAXB by default) to avoid these compatibility issues. Make sure your `JAVA_HOME` environment variable points to a Java 8 installation. 4. ****Set Java version explicitly for Flutter**** If you decide to use Java 8, set the Java version explicitly in your Flutter environment by updating your `flutter doctor --android-licenses` setup: `export JAVA_HOME=/path/to/java8 flutter doctor --android-licenses` After performing one of the above solutions, retry running `flutter doctor --android-licenses` and it should work without the error.

Co-Pilot, Verb Count: 20 , **Adjective Count:** 5

Co-Pilot Answer: The error you're encountering is due to the removal of the JAXB API from the JDK starting with Java 9. The `javax.xml.bind` package is no longer included by default in Java 11 and later

versions. To fix this, you need to add the JAXB API as a dependency in your project. Here's how you can do it: 1. If you're using Maven, add the following dependency to your pom.xml: `<dependency>
<groupId>javax.xml.bind</groupId> <artifactId>jaxb-api</artifactId>
<version>2.3.1</version> </dependency>` 2. If you're using Gradle, add the following dependency to your build.gradle: `implementation 'javax.xml.bind:jaxb-api:2.3.1'` 3. If you're using another build tool, make sure to add the JAXB API dependency accordingly. After adding the dependency, try running flutter doctor --android-licenses again. This should resolve the NoClassDefFoundError.

Month: Jun-20

Question 1:

Tags: python, linux, alpine-linux

Question: How do I install python3 and python3-pip on an alpine based image (without using a python image)? `$ apk add --update python3.8 python3-pip ERROR: unsatisfiable constraints: python3-pip (missing): required by: world[python3-pip] python3.8 (missing): required by: world[python3.8]`

Answer Score: 256

Stack Overflow answer, Verb Count: 6 , Adjective Count: 1

Stack Overflow Answer: This is what I use in a Dockerfile for an alpine image: `# Install python/pip ENV PYTHONUNBUFFERED=1 RUN apk add --update --no-cache python3 && ln -sf python3 /usr/bin/python RUN python3 -m ensurepip RUN pip3 install --no-cache --upgrade pip setuptools`

GPT answer, Verb Count: 11 , Adjective Count: 11

GPT-4o Answer: To install `python3` and `python3-pip` on an Alpine-based image, follow these steps: `apk update && apk add --no-cache python3 py3-pip` Explanation: - The `python3` and `py3-pip` packages are available in the main repository of Alpine Linux. - Use `--no-cache` to avoid caching the index, which reduces image size. - Make sure the Alpine base image is up-to-date by running `apk update` before installing packages.

Co-Pilot, Verb Count: 7 , Adjective Count: 1

Co-Pilot Answer: To install Python 3 and Python 3-pip on an Alpine-based image, you can use the following command: `apk add --update python3 py3-pip` This command will install Python 3 and pip for Python 3 on your Alpine-based image.

Question 2:

Tags: javascript, reactjs, react-hooks, react-hook-form

Question: I am creating a page for user to update personal data with React-Hook-Form. Once paged is loaded, I use `useEffect` to fetch the user's current personal data and set them into default value of the form. I put the fetched value into `defaultValue` of `<Controller />`. However, it is just not showing in the text box. Here is my code: `import React, {useState, useEffect, useCallback} from 'react'; import { useForm, Controller } from 'react-hook-form' import { URL } from '../constants'; const UpdateUserData = props => { const [userData, setUserData] = useState(null); const { handleSubmit, control } = useForm({mode: 'onBlur'}); const fetchUserData = useCallback(async account => { const userData = await fetch(`${URL}/user/${account}`) .then(res=> res.json());`

```

console.log(userData); setUserData(userData); }, []); useEffect(() => {
const account = localStorage.getItem('account'); fetchUserData(account); },
[fetchUserData]) const onSubmit = async (data) => { // TODO } return ( <div>
<form onSubmit={handleSubmit(onSubmit)}> <div> <label>User Name:</label>
<Controller as={<input type='text' />} control={control}
defaultValue={userData ? userData.name : ''} name='name' /> </div> <div>
<label>Phone:</label> <Controller as={<input type='text' />}
control={control} defaultValue={userData ? userData.phone : ''} name='phone'
/> </div> <button>Submit</button> </form> </div> ); } export default
UpdateUserData; The called API is working well and the value is actually set to userData state. {
name: "John", phone: "02-98541566" ... } I also tried to setUserData with mock data in
useEffect(), and it doesn't work either. Is there any problem in my above code?

```

Answer Score: 97

Stack Overflow answer, Verb Count: 8 , **Adjective Count:** 4

Stack Overflow Answer: You can use `setValue` (<https://react-hook-form.com/api/useform/setvalue>). Import it from `useForm`: `const { handleSubmit, control, setValue } = useForm({ mode: 'onBlur' });` Then call it with the user data after it's received: `useEffect(() => { if (userData) { setValue([{ name: userData.name }, { phone: userData.phone }]); } }, [userData]);` You can remove the default values from the form. EDIT: See alternative answers below if this does not work.

GPT answer, Verb Count: 26 , **Adjective Count:** 23

GPT-4o Answer: The issue arises because `defaultValue` in the `<Controller />` is only set once when the component mounts, and it does not update dynamically when `userData` changes. To solve this, you need to use the `useEffect` hook to update the form values dynamically after fetching the user data using `setValue` from `react-hook-form`. Here's the corrected code: `import React, { useState, useEffect, useCallback } from 'react'; import { useForm, Controller } from 'react-hook-form'; import { URL } from '../constants'; const UpdateUserData = props => { const [userData, setUserData] = useState(null); const { handleSubmit, control, setValue } = useForm({ mode: 'onBlur' }); const fetchUserData = useCallback(async account => { const userData = await fetch(`${URL}/user/${account}`).then(res => res.json()); console.log(userData); setUserData(userData); }, []); useEffect(() => { const account = localStorage.getItem('account'); fetchUserData(account); }, [fetchUserData]); useEffect(() => { if (userData) { setValue('name', userData.name); setValue('phone', userData.phone); } }, [userData, setValue]); const onSubmit = async (data) => { console.log('Form Data:', data); // TODO: handle form submission }; return (User Name: } control={control} name='name' defaultValue="" /> Phone: } control={control} name='phone' defaultValue="" /> Submit); } export default UpdateUserData;` Key changes: - Added a second `useEffect` to update form values dynamically using `setValue` when `userData` is fetched. - Kept `defaultValue` as an empty string since the values are set dynamically after fetching data.

Co-Pilot, Verb Count: 15 , **Adjective Count:** 16

Co-Pilot Answer: The issue with the code is that the `defaultValue` prop in `Controller` only sets the initial value when the component mounts and does not update when `userData` state changes. To fix this, you need to use the `setValue` function from `useForm` to set the value dynamically when `userData` is fetched. Here is the updated code: `import React, { useState, useEffect, useCallback } from 'react'; import { useForm, Controller } from 'react-hook-form'; import { URL } from '../constants'; const UpdateUserData`


```
= props => { const [userData, setUserData] = useState(null); const {
handleSubmit, control, setValue } = useForm({ mode: 'onBlur' }); const
fetchUserData = useCallback(async account => { const userData = await
fetch(`${URL}/user/${account}`).then(res => res.json());
console.log(userData); setUserData(userData); }, []); useEffect(() => {
const account = localStorage.getItem('account'); fetchUserData(account); },
[fetchUserData]); useEffect(() => { if (userData) { setValue('name',
userData.name); setValue('phone', userData.phone); } }, [userData,
setValue]); const onSubmit = async (data) => { // TODO }; return ( <div>
<form onSubmit={handleSubmit(onSubmit)}> <div> <label>User Name:</label>
<Controller as={<input type='text' />} control={control} name='name'
defaultValue="" /> </div> <div> <label>Phone:</label> <Controller as={<input
type='text' />} control={control} name='phone' defaultValue="" /> </div>
<button>Submit</button> </form> </div> ); }; export default UpdateUserData;
```

With this change, the form fields will update with the user's data once it has been fetched.

Question 3:

Tags: javascript, import, export, node.js

Question: I am getting this error `SyntaxError: Cannot use import statement outside a module` when trying to import from another javascript file. This is the first time I'm trying something like this. The main file is `main.js` and the module file is `mod.js`. `main.js: import * as myModule from "mod"; myModule.func(); mod.js: export function func(){ console.log("Hello World"); }` How can I fix this? Thanks

Answer Score: 195

Stack Overflow answer, Verb Count: 7 , Adjective Count: 4

Stack Overflow Answer: In order to use the import syntax (ESModules), you need to add the following to your `package.json` at the top level: `{ // ... "type": "module", }` If you are using a version of Node earlier than 13, you additionally need to use the `--experimental-modules` flag when you run the program: `node --experimental-modules program.js`

GPT answer, Verb Count: 22 , Adjective Count: 9

GPT-4o Answer: The error occurs because the JavaScript runtime does not recognize your files as ES modules by default. To fix this, you need to do the following: 1. Update your `package.json` to specify the module type: `{ "type": "module" }` 2. Ensure your import statement includes the file extension in `main.js: import * as myModule from "./mod.js"; myModule.func();` 3. If you're running this in Node.js, make sure you use a version that supports ES modules (Node.js 12+ with the module flag or 14+ natively supports ES modules). 4. If you are running this in the browser, ensure you use the `type="module"` attribute in your HTML file: [ES Module Example](#) By specifying the module type in `package.json` or using the `type="module"` attribute in your HTML, the JavaScript runtime knows to treat your files as ES modules.

Co-Pilot, Verb Count: 13 , Adjective Count: 4

Co-Pilot Answer: The error occurs because the JavaScript environment does not recognize your files as ES modules. To fix this, you can do the following: 1. Ensure you are using the latest version of Node.js. 2. Add a `package.json` file in your project directory with the following content: `{ "type": "module" }` This will tell Node.js to treat your files as ES modules. After doing this, your import statement should work correctly, and the error should be resolved.

Question 4:

Tags: javascript, reactjs, redux

Question: I am getting this warning in react: `index.js:1 Warning: Cannot update a component (`ConnectFunction`) while rendering a different component (`Register`).` To locate the bad `setState()` call inside ``Register`` I went to the locations indicated in the stack trace and removed all `setStates` but the warning still persists. Is it possible this could occur from redux dispatch? my code: `register.js`

```
class Register extends Component {
  render() {
    if( this.props.registerStatus === SUCCESS) { // Reset register status to allow return to register page
      this.props.dispatch(resetRegisterStatus())
      # THIS IS THE LINE THAT CAUSES THE ERROR ACCORDING TO THE STACK TRACE
      return <Redirect push to = {HOME}/> }
    return ( <div style = {{paddingTop: "180px", background: 'radial-gradient(circle, rgba(106,103,103,1) 0%, rgba(36,36,36,1) 100%)', height: "100vh"}}>
      <RegistrationForm/> </div> );
  }
}

function mapStateToProps( state ) {
  return {
    registerStatus: state.userReducer.registerStatus
  }
}

export default connect ( mapStateToProps ) ( Register );
```

function which triggers the warning in my `registerForm` component called by `register.js`

```
handleSubmit = async () => {
  if( this.isValidForm() ) {
    const details = {
      "username": this.state.username,
      "password": this.state.password,
      "email": this.state.email,
      "clearance": this.state.clearance
    }
    await this.props.dispatch( register(details) )
    if( this.props.registerStatus !== SUCCESS && this.mounted ) {
      this.setState( {errorMsg: this.props.registerError} )
      this.handleShowError()
    }
    else {
      if( this.mounted ) {
        this.setState( {errorMsg: "Error - registration credentials are invalid!"} )
        this.handleShowError()
      }
    }
  }
}
```

Stacktrace:

Answer Score: 24

Stack Overflow answer, Verb Count: 16 , **Adjective Count:** 4

Stack Overflow Answer: I fixed this issue by removing the dispatch from the register components render method to the `componentWillUnmount` method. This is because I wanted this logic to occur right before redirecting to the login page. In general it's best practice to put all your logic outside the render method so my code was just poorly written before. Hope this helps anyone else in future :) My refactored register component:

```
class Register extends Component {
  componentWillUnmount() { // Reset register status to allow return to register page
    if ( this.props.registerStatus !== "" )
      this.props.dispatch(resetRegisterStatus())
  }
  render() {
    if( this.props.registerStatus === SUCCESS ) {
      return <Redirect push to = {LOGIN}/>
    }
    return ( <div style = {{paddingTop: "180px", background: 'radial-gradient(circle, rgba(106,103,103,1) 0%, rgba(36,36,36,1) 100%)', height: "100vh"}}>
      <RegistrationForm/> </div> );
  }
}
```

GPT answer, Verb Count: 18 , **Adjective Count:** 8

GPT-4o Answer: The warning occurs because you are calling `this.props.dispatch(resetRegisterStatus())` during the rendering phase of the `Register` component. React does not allow updates (like state changes or dispatches) to happen during the render process. This is likely the source of the warning. You can fix this issue by moving the dispatch call to a lifecycle method like `componentDidUpdate` instead of directly calling it inside `render`.

Here's how you can fix it:

```
class Register extends Component {
  componentDidUpdate(prevProps) {
    if ( prevProps.registerStatus !== this.props.registerStatus && this.props.registerStatus === SUCCESS ) { // Reset register status to allow return to register page
      this.props.dispatch(resetRegisterStatus());
    }
  }
  render() {
    if (this.props.registerStatus === SUCCESS) {
      return ;
    }
    return ( );
  }
}

function mapStateToProps(state) {
  return { registerStatus:

```

```
state.userReducer.registerStatus, }; } export default
connect(mapStateToProps)(Register); This approach ensures that
dispatch(resetRegisterStatus()) is only called after the render phase, in the
componentDidUpdate lifecycle method, avoiding the React warning.
```

Co-Pilot, Verb Count: 16 , Adjective Count: 5

Co-Pilot Answer: The warning you are seeing is due to trying to dispatch an action while another component is rendering. To avoid this, you can use a lifecycle method like `componentDidUpdate` instead of dispatching during the render phase. Here's an updated version of the `Register` component:

```
import React, { Component } from 'react'; import { connect } from
'react-redux'; import { Redirect } from 'react-router-dom'; import {
resetRegisterStatus } from '../actions'; import RegistrationForm from
'./RegistrationForm'; import { SUCCESS, HOME } from '../constants'; class
Register extends Component { componentDidUpdate(prevProps) { if
(this.props.registerStatus === SUCCESS && prevProps.registerStatus !==
SUCCESS) { this.props.dispatch(resetRegisterStatus()); } } render() { if
(this.props.registerStatus === SUCCESS) { return <Redirect push to={HOME}
/>; } return ( <div style={{ paddingTop: '180px', background:
'radial-gradient(circle, rgba(106,103,103,1) 0%, rgba(36,36,36,1) 100%)',
height: '100vh' }}> <RegistrationForm /> </div> ); } } function
mapStateToProps(state) { return { registerStatus:
state.userReducer.registerStatus }; } export default
connect(mapStateToProps)(Register); By moving the dispatch logic to
componentDidUpdate, you avoid updating the state during the render phase, thus resolving the
warning.
```

Question 5:

Tags: javascript, node.js, npm, node-modules, typescript

Question: This seems to be a trivial problem, but it is not very obvious what settings/configurations need to be used to solve this issue. Here are the Hello World program directory structure and the source code: Directory Structure: | -- HelloWorldProgram | -- HelloWorld.ts | -- index.ts | -- package.json | -- tsconfig.json index.ts:

```
import {HelloWorld} from "../HelloWorld"; let world = new HelloWorld();
```

 HelloWorld.ts:

```
export class HelloWorld { constructor(){ console.log("Hello World!"); } }
```

 package.json:

```
{ "type": "module", "scripts": { "start": "tsc && node index.js" } }
```

 Now, execution of the command `tsc && node index.js` results in the following error:

```
internal/modules/run_main.js:54
internalBinding('errors').triggerUncaughtException( ^ Error
[ERR_MODULE_NOT_FOUND]: Cannot find module 'HelloWorld' imported from
HelloWorld\index.js Did you mean to import ../HelloWorld.js? at
finalizeResolution (internal/modules/esm/resolve.js:284:11) at moduleResolve
(internal/modules/esm/resolve.js:662:10) at Loader.defaultResolve [as
_resolve] (internal/modules/esm/resolve.js:752:11) at Loader.resolve
(internal/modules/esm/loader.js:97:40) at Loader.getModuleJob
(internal/modules/esm/loader.js:242:28) at ModuleWrap.<anonymous>
(internal/modules/esm/module_job.js:50:40) at link
(internal/modules/esm/module_job.js:49:36) { code: 'ERR_MODULE_NOT_FOUND' }
```

It is obvious that the problem seems to have been originated from the fact that in `index.ts` Typescript file there is no `.js` extension in the import statement (`import {HelloWorld} from "../HelloWorld";`). Typescript didn't throw any error during compilation. However, during runtime Node (v14.4.0) wants the `.js` extension. Hope the context is clear. Now, how to change the compiler

output setting (tsconfig.json or any flags) so that local relative path imports such as `import {HelloWorld} from ./Helloworld;` will get replaced by `import {HelloWorld} from ./Helloworld.js;` during Typescript to Javascript compilation in the `index.js` file? Note: It is possible to directly use the `.js` extension while importing inside typescript file. However, it doesn't help much while working with hundreds of old typescript modules, because then we have to go back and manually add `.js` extension. Rather than that for us better solution is to batch rename and remove all the `.js` extension from all the generated `.js` filenames at last.

Answer Score: 109

Stack Overflow answer, Verb Count: 22 , **Adjective Count:** 18

Stack Overflow Answer: The possible work-arounds we have come across are as follows: Use `.js` extension in the import: For new files, it is possible to simply add `".js"` extension in the import statement in TypeScript file while editing. Example: `import {HelloWorld} from "./Helloworld.js";` Extensionless filename If working with old projects, rather than going through each and every file and updating the import statements, we found it easier to simply batch rename and remove the `".js"` extension from the generated JavaScript via a simple automated script. Please note however that this might require a minor change in the server side code to serve these extension-less `".js"` files with the proper MIME type to the clients. Use regex to batch replace import statements Another option is to use regular expression to batch find and replace in all files the import statements to add the `.js` extension. An example: <https://stackoverflow.com/a/73075563/3330840> or similar other answers.

GPT answer, Verb Count: 27 , **Adjective Count:** 23

GPT-4o Answer: To resolve this issue, you need to configure the TypeScript compiler to append the `.js` file extension to your module imports during the compilation process. Here's how you can achieve this: Update your `tsconfig.json` file to include the following:

```
{ "compilerOptions": { "module": "esnext", "target": "esnext", "moduleResolution": "node", "outDir": "./dist", "esModuleInterop": true, "resolveJsonModule": true, "allowSyntheticDefaultImports": true, "strict": true, "noImplicitAny": true, "skipLibCheck": true, "forceConsistentCasingInFileNames": true, "noEmitOnError": false } }
```

 The key settings here are: 1. `"module": "esnext"`: Ensures TypeScript uses native ECMAScript modules. 2. `"moduleResolution": "node"`: Resolves imports like Node.js does. Next, add the `--outFileExtension` to explicitly control the behavior during compilation. Add a custom script in your `package.json`:

```
"scripts": { "build": "tsc && node dist/index.js", "postbuild": "find ./dist -name '*.js' | xargs sed -i 's/\\.ts/\\.js/g' }
```

 This script ensures that the generated files replace `.ts` with `.js`. Now, run the build process with: `npm run build` This will resolve the issue by adding the `.js` extension automatically in compiled output files, without requiring changes to all your existing TypeScript files.

Co-Pilot, Verb Count: 26 , **Adjective Count:** 16

Co-Pilot Answer: To resolve the issue and make sure the TypeScript compiler adds the `.js` extension during the compilation, you can use the `tsconfig.json` settings along with a custom script to handle the renaming of import paths in the generated JavaScript files. Here's how you can do it: 1. First, update your `tsconfig.json` to include the `esModuleInterop` and `allowJs` options:

```
{ "compilerOptions": { "target": "ES6", "module": "ES6", "moduleResolution": "node", "outDir": "./dist", "esModuleInterop": true, "allowJs": true }, "include": ["./*/.ts"], "exclude": ["node_modules"] }
```

 2. Create a custom script to add the `.js` extension to the import paths in the generated JavaScript files:

```
const fs = require('fs'); const path = require('path'); const directory = './dist'; function addJsExtension(dir) { fs.readdirSync(dir).forEach(file => { const
```

```
filePath = path.join(dir, file); const stat = fs.lstatSync(filePath); if
(stat.isDirectory()) { addJsExtension(filePath); } else if
(filePath.endsWith('.js')) { let content = fs.readFileSync(filePath,
'utf8'); content = content.replace(/(from\s+['"].\./.*)(['"])/g, '$1.js$2');
fs.writeFileSync(filePath, content, 'utf8'); } }); }
addJsExtension(directory); 3. Update your package.json to include a script that runs the
TypeScript compiler and then the custom script: { "type": "module", "scripts": {
"start": "tsc && node addJsExtension.js && node dist/index.js" } } By using
these steps, the TypeScript compiler will generate the JavaScript files and the custom script will update
the import paths to include the .js extension.
```

Month: Jul-20

Question 1:

Tags: python, packaging, pip, pyproject.toml

Question: Background I was about to try Python package downloaded from GitHub, and realized that it did not have a `setup.py`, so I could not install it with `pip install -e <folder>`. Instead, the package had a `pyproject.toml` file which seems to have very similar entries as the `setup.py` usually has. What I found Googling lead me into PEP-518 and it gives some critique to `setup.py` in Rationale section. However, it does not clearly tell that usage of `setup.py` should be avoided, or that `pyproject.toml` would as such completely replace `setup.py`. Questions Is the `pyproject.toml` something that is used to replace `setup.py`? Or should a package come with both, a `pyproject.toml` and a `setup.py`? How would one install a project with `pyproject.toml` in an editable state?

Answer Score: 147

Stack Overflow answer, Verb Count: 68 , **Adjective Count:** 61

Stack Overflow Answer: What is it for? Currently there are multiple packaging tools being popular in Python community and while `setuptools` still seems to be prevalent it's not a de-facto standard anymore. This situation creates a number of hassles for both end-users and developers: For `setuptools`-based packages installation from source / build of a distribution can fail if one doesn't have `setuptools` installed; `pip` doesn't support the installation of packages based on other packaging tools from source, so these tools had to generate a `setup.py` file to produce a compatible package. To build a distribution package one has to install the packaging tool first and then use tool-specific commands; If package author decides to change the packaging tool, workflows must be changed as-well to use different tool-specific commands. `pyproject.toml` is a new configuration file introduced by PEP-517 and PEP-518 to solve these problems: ... think of the (rough) steps required to produce a built artifact for a project: The source checkout of the project. Installation of the build system. Execute the build system. This PEP [518] covers step #2. PEP-517 covers step #3 ... Any tool can also extend this file with its own section (table) to accept tool-specific options, but it's up to them and not required. PEP-621 suggests using `pyproject.toml` to specify package core-metadata in static, tool-agnostic way. Which backends currently support this is shown in the following table: `enscons` `flit_core` `hatchling` `pdm-pep517` `poetry-core` `setuptools` 0.26.0+ 3.2+ 0.3+ 0.3.0+ Issue-#3332 61.0.0+ Does it replace `setup.py`? For `setuptools`-based packages `pyproject.toml` is not strictly meant to replace `setup.py`, but rather to ensure its correct execution if it's still needed. For other packaging tools, Æi yes, it is: Where the `build-backend` key exists, this takes precedence and the source tree follows the format and conventions of the specified backend (as such no `setup.py` is needed unless the backend requires it). Projects may still wish to include a `setup.py` for compatibility with tools that do not use this spec. How to install a package in editable mode? Originally "editable install" was a `setuptools`-specific feature and as such it was not

supported by PEP-517. Later, on PEP-660 extended this concept to packages using `pyproject.toml`. There are two possible conditions for installing a package in editable mode using `pip`: Modern: Both the frontend (`pip`) and a backend must support PEP-660. `pip` supports it since version-21.3; Legacy: Packaging tool must provide a `setup.py` file which supports the `develop` command. Since version-21.1 `pip` can also install packages using only `setup.cfg` file in editable mode. The following table describes the support of editable installs by various backends: `enscons` `flit_core` `hatchling` `pdm-pep517` `poetry-core` `setuptools` 0.28.0+ 3.4+ 0.3+ 0.8.0+ 1.0.8+ 64.0.0+

GPT answer, Verb Count: 37 , Adjective Count: 49

GPT-4o Answer: The `pyproject.toml` file is part of PEP 518 and is designed to replace or complement `setup.py` by providing a standardized way to specify build system requirements for Python projects. Here are the answers to your questions: Is `pyproject.toml` used to replace `setup.py`? Yes, `pyproject.toml` is designed to replace `setup.py` in many cases. It provides a more modern, standardized, and flexible approach to defining project metadata and build dependencies. Tools like `setuptools`, `poetry`, and `flit` use `pyproject.toml` to define project configurations. However, not all projects have transitioned to using `pyproject.toml` alone. Some still include a `setup.py` for backward compatibility or additional functionality that is not yet fully supported in `pyproject.toml`. Should a package come with both `pyproject.toml` and `setup.py`? A package can include both, but it is not strictly necessary. If the build system (like `setuptools` or `poetry`) fully supports `pyproject.toml`, the `setup.py` file may not be needed. However, including `setup.py` can be useful for compatibility with older tools or workflows. How to install a project with `pyproject.toml` in an editable state? You can install a project with `pyproject.toml` in an editable state using a tool that supports this functionality, like `pip`. Use the following command: `pip install -e .` For this to work, the project must use a build backend (e.g., `setuptools`, `poetry`, `flit`) that supports editable installations. The `pyproject.toml` file should specify the build backend and its dependencies in the `[build-system]` section. Example: `[build-system] requires = ["setuptools>=42", "wheel"] build-backend = "setuptools.build_meta"` If the backend does not support editable installs, you will need to install the project in a non-editable state or use a different tool to work with it.

Co-Pilot, Verb Count: 23 , Adjective Count: 21

Co-Pilot Answer: The `pyproject.toml` file, introduced by PEP 518, is designed to specify build system requirements for Python projects. It can be used to replace `setup.py` for projects using newer build systems like Poetry or Flit. However, for projects still using `setuptools`, `setup.py` may still be necessary. To install a project with `pyproject.toml` in an editable state, you can use the following command: `pip install -e .` This requires the `pyproject.toml` file to include a build-backend, such as `setuptools`, which ensures `pip` can handle the editable install. Here's an example `pyproject.toml` setup: `[build-system] requires = ["setuptools", "wheel"] build-backend = "setuptools.build_meta"` In summary, `pyproject.toml` can replace `setup.py` for specifying build requirements, but `setup.py` might still be used for compatibility with `setuptools`. To install a project in an editable state, use `pip install -e .` with the appropriate build system specified in `pyproject.toml`.

Question 2:

Tags: python, pip, python-poetry

Question: I am trying out Poetry in an existing project. It used `pyenv` and `virtual env` originally so I have a `requirements.txt` file with the project's dependencies. I want to import the `requirements.txt` file using Poetry, so that I can load the dependencies for the first time. I've looked through poetry's documentation, but I haven't found a way to do this. Is there a way to do it? I know that I can add all packages manually, but I was hoping for a more automated process, because there are a lot of

packages.

Answer Score: 254

Stack Overflow answer, Verb Count: 7 , **Adjective Count:** 4

Stack Overflow Answer: `poetry` doesn't support this directly. But if you have a handmade list of required packages (at best without any version numbers), that only contain the main dependencies and not the dependencies of a dependency you could do this: `$ cat requirements.txt | xargs poetry add`

GPT answer, Verb Count: 29 , **Adjective Count:** 21

GPT-4o Answer: Poetry does not natively support importing dependencies directly from a `requirements.txt` file, but you can achieve this using the `poetry add` command in combination with the contents of your `requirements.txt`. Here are two approaches to handle this: 1. Using a Shell Command to Automate Adding Dependencies You can use a shell command to read and add all dependencies from the `requirements.txt` file in one go: `cat requirements.txt | xargs poetry add` This command reads each line from `requirements.txt` and passes it to `poetry add`, which adds the dependencies to the `pyproject.toml` file. 2. Using a Python Script for Complex Requirements If you need more control (e.g., handling version constraints), you can use a Python script to parse the `requirements.txt` file and add dependencies via Poetry: `import subprocess with open('requirements.txt', 'r') as f: for line in f: dependency = line.strip() if dependency: # Skip empty lines subprocess.run(['poetry', 'add', dependency])` This script reads each dependency from `requirements.txt` and runs `poetry add` for each line. Notes: - Ensure that your `requirements.txt` file contains valid dependency specifications (e.g., `package==version`). - After importing, you can verify the dependencies in the `pyproject.toml` file and run `poetry install` to create the virtual environment and install the packages.

Co-Pilot, Verb Count: 13 , **Adjective Count:** 6

Co-Pilot Answer: To import the `requirements.txt` file into Poetry, you can use the following command: `poetry add -D --lock requirements.txt` This command will read the dependencies from the `requirements.txt` file and add them to your Poetry project, ensuring that the exact versions specified are used. After running this command, you can install the dependencies with: `poetry install` This will create a virtual environment for your project and install the dependencies.

Question 3:

Tags: python, pip, path, requirements.txt

Question: I am working on developing a python package. I use `pip freeze > requirements.txt` to add the required package into the requirement.txt file. However, I realized that some of the packages, instead of the package version, have some path in front of them. `numpy==1.19.0 packaging==20.4 pandas @ file:///opt/concourse/worker/volumes/live/38d1301c-8fa9-4d2f-662e-34dddf33b183/volume/pandas_1592841668171/work pandocfilters==1.4.2` Whereas, inside the environment, I get: `>>> pandas.__version__ '1.0.5'` Do you have any idea how to address this problem?

Answer Score: 361

Stack Overflow answer, Verb Count: 14 , **Adjective Count:** 11

Stack Overflow Answer: It looks like this is an open issue with `pip freeze` in version 20.1, the current workaround is to use: `pip list --format=freeze > requirements.txt` In a nutshell, this is caused by changing the behavior of `pip freeze` to include direct references for distributions installed from direct URL references. You can read more about the issue on GitHub: `pip freeze` does

not show version for in-place installs Output of "pip freeze" and "pip list --format=freeze" differ for packages installed via Direct URLs Better freeze of distributions installed from direct URL references

GPT answer, Verb Count: 27 , Adjective Count: 16

GPT-4o Answer: The issue occurs because some packages, like `pandas`, were installed from a local file or custom path instead of from PyPI. When you run `pip freeze`, it records the installation source rather than just the version for such packages. To address this, you can clean up the `requirements.txt` file by replacing the custom paths with the package name and version. Solution To fix this issue, follow these steps: 1. Use the Correct Version Manually edit the `requirements.txt` file to replace the problematic entries. For example, update: `pandas @ file:///opt/concourse/worker/volumes/live/38d1301c-8fa9-4d2f-662e-34dddf33b183/volume/pandas_1592841668171/work` to: `pandas==1.0.5` You can find the version using: `python -c "import pandas; print(pandas.__version__)"` 2. Reinstall Packages from PyPI To prevent this from happening in the future, reinstall any package with a custom path directly from PyPI: `pip install --force-reinstall pandas` This ensures that `pip freeze` will only record the version, not the source path. 3. Automate the Cleanup (Optional) If you have many such entries, you can use a script to automate the cleanup: `import re with open('requirements.txt', 'r') as file: lines = file.readlines() with open('requirements_cleaned.txt', 'w') as file: for line in lines: match = re.match(r'(.*) @ file.*', line) if match: package = match.group(1) version = __import__(package).__version__ file.write(f'{package}=={version}\n') else: file.write(line)` This script will generate a cleaned `requirements_cleaned.txt` file.

Co-Pilot, Verb Count: 16 , Adjective Count: 12

Co-Pilot Answer: The issue you're encountering is because some of the packages were installed from a local path or a specific build environment, resulting in the path being included in the `requirements.txt` file. To address this and ensure that only the package versions are listed, you can manually edit the `requirements.txt` file or use a virtual environment with only the necessary packages installed from the official repositories. Here's a way to automate the cleanup: `pip freeze | sed 's/.*@.*//g' > clean_requirements.txt` This command will remove any local paths from the output of `pip freeze` and save the clean list of packages and their versions to `clean_requirements.txt`.

Question 4:

Tags: html, javascript, reactjs, yarnpkg

Question: I'm trying to run very simple code, but I'm getting an error, I didn't use the create react app! It looks like my babel.config.js file is being ignored! This is the structure of my small project: My html file

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0">
<title>ReactJS</title> </head> <body> <div id="app"></div> <script src =
'bundle.js' ></script> </body> </html> My index.js file: import React from
'react'; import { render } from 'react-dom'; render(<h1>Hello World!!</h1>,
document.getElementById('app')); My package json: { "name": "front", "version":
"1.0.0", "main": "index.js", "license": "MIT", "scripts": { "dev":
"webpack-dev-server --mode development", "build": "webpack-dev-server --mode
production" }, "dependencies": { "@babel/cli": "^7.10.5", "@babel/core":
"^7.10.5", "@babel/plugin-proposal-class-properties": "^7.10.4",
"@babel/preset-env": "^7.10.4", "@babel/preset-react": "^7.10.4", "react":
"^16.13.1", "react-dom": "^16.13.1", "webpack": "^4.43.0", "webpack-cli":
"^3.3.12", "webpack-dev-server": "^3.11.0" }, "devDependencies": {
"@babel/plugin-transform-runtime": "^7.9.0", "babel-loader": "^8.1.0",
```

```

"webpack-dev-server": "^3.10.3" } } My webpack.config.js const path =
require('path'); module.exports = { entry: path.resolve(__dirname, 'src',
'index.js'), output: { path: path.resolve(__dirname, 'public'), filename:
'bundle.js' }, devServer: { contentBase: path.resolve(__dirname, 'public'),
}, module: { rules: [{ test: /\.js$/, exclude: /node_modules/, use: {
loader: 'babel-loader', } }] }, }; And this is my babel.config.js module.exports = {
"presets": ["@babel/preset-env", "@babel/preset-react"] }; Error when yarn
webpack-dev-server --mode development ERROR in ./src/index.js Module build
failed (from ./node_modules/babel-loader/lib/index.js): SyntaxError:
/root/treina/front/src/index.js: Support for the experimental syntax 'jsx'
isn't currently enabled (4:8): 2 | import { render } from 'react-dom'; 3 | >
4 | render(<h1>Hello World!!</h1>, document.getElementById('app')); | ^ Add
@babel/preset-react (https://git.io/JfeDR) to the 'presets' section of your
Babel config to enable transformation. If you want to leave it as-is, add
@babel/plugin-syntax-jsx (https://git.io/vb4yA) to the 'plugins' section to
enable parsing. at Parser._raise
(/root/treina/front/node_modules/@babel/parser/lib/index.js:757:17) at
Parser.raiseWithData
(/root/treina/front/node_modules/@babel/parser/lib/index.js:750:17) at
Parser.expectOnePlugin
(/root/treina/front/node_modules/@babel/parser/lib/index.js:8849:18) at
Parser.parseExprAtom
(/root/treina/front/node_modules/@babel/parser/lib/index.js:10170:22) at
Parser.parseExprSubscripts
(/root/treina/front/node_modules/@babel/parser/lib/index.js:9688:23) at
Parser.parseMaybeUnary
(/root/treina/front/node_modules/@babel/parser/lib/index.js:9668:21) at
Parser.parseExprOps
(/root/treina/front/node_modules/@babel/parser/lib/index.js:9538:23) at
Parser.parseMaybeConditional
(/root/treina/front/node_modules/@babel/parser/lib/index.js:9511:23) at
Parser.parseMaybeAssign
(/root/treina/front/node_modules/@babel/parser/lib/index.js:9466:21) at
Parser.parseExprListItem
(/root/treina/front/node_modules/@babel/parser/lib/index.js:10846:18) , Ñπ
ÔΩçwdmÔΩ£: Failed to compile. I'm using yarn and the WSL terminal

```

Answer Score: 191

Stack Overflow answer, Verb Count: 2 , **Adjective Count:** 2

Stack Overflow Answer: Just create a `.babelrc` file in the root of your project and add: {

```
"presets": ["@babel/preset-env", "@babel/preset-react"] }
```

GPT answer, Verb Count: 31 , **Adjective Count:** 27

GPT-4o Answer: The issue arises because Babel isn't correctly recognizing the JSX syntax in your code. This is due to either Babel not finding your `babel.config.js` file or an incorrect configuration. Here are steps to resolve the issue: Solution 1. Ensure `babel.config.js` is in the Project Root Verify that `babel.config.js` is located in the root directory of your project. The file should be at the same level as `package.json`. 2. Fix the Babel Configuration Ensure your `babel.config.js` contains the correct presets: `module.exports = { presets: ["@babel/preset-env", "@babel/preset-react"] };` 3. Update `webpack.config.js` Ensure the `babel-loader` is configured properly in your `webpack.config.js`. Verify this rule: `module: { rules: [{ test: /\.js$/, exclude: /node_modules/, use: { loader: 'babel-loader', }, },], },`

4. Install Missing Dependencies Ensure all required dependencies are installed, especially `@babel/preset-react`. Run: `yarn add @babel/preset-react --dev` 5. Restart Webpack Dev Server Stop the running development server and restart it: `yarn webpack-dev-server --mode development` 6. Use `.babelrc` as a Fallback If the issue persists, try replacing `babel.config.js` with a `.babelrc` file. Create a `.babelrc` in the root directory with the following content: `{ "presets": ["@babel/preset-env", "@babel/preset-react"] }` 7. Confirm WSL Permissions Ensure that your WSL environment has proper permissions to read all project files, including `babel.config.js`. Expected Outcome After following these steps, the JSX syntax should compile successfully, and the error will be resolved.

Co-Pilot, Verb Count: 18 , **Adjective Count:** 9

Co-Pilot Answer: To resolve the issue with Babel not transforming your JSX, ensure that your Babel configuration is being correctly picked up by Webpack. Here are the steps: 1. Verify that the `babel.config.js` file is correctly named and located at the project root. The contents should be: `module.exports = { "presets": ["@babel/preset-env", "@babel/preset-react"] };` 2. Ensure that your `webpack.config.js` file correctly references the Babel loader. Here is the updated `webpack.config.js`: `const path = require('path'); module.exports = { entry: path.resolve(__dirname, 'src', 'index.js'), output: { path: path.resolve(__dirname, 'public'), filename: 'bundle.js' }, devServer: { contentBase: path.resolve(__dirname, 'public'), }, module: { rules: [{ test: /\.js$/, exclude: /node_modules/, use: { loader: 'babel-loader', options: { presets: ['@babel/preset-env', '@babel/preset-react'] } } }] }, };` 3. Ensure that you have installed all the required Babel dependencies. You can add them via Yarn: `yarn add @babel/core @babel/preset-env @babel/preset-react babel-loader` With these changes, Babel should correctly transform your JSX code, and you should be able to run your project without encountering the syntax error.

Question 5:

Tags: java, java-platform-module-system, java-9, java-11

Question: We're currently migrating from Java 8 to Java 11. However, upgrading our services was less painful, than we anticipated. We basically only had to change the version number in our `build.gradle` file and the services were happily up and running. We upgraded libraries as well as (micro) services that use those libs. No problems until now. Is there any need to actually switch to modules? This would generate needless costs IMHO. Any suggestion or further reading material is appreciated. To clarify, are there any consequences if Java 9+ code is used without introducing modules? E.g. can it become incompatible with other code?

Answer Score: 183

Stack Overflow answer, Verb Count: 40 , **Adjective Count:** 19

Stack Overflow Answer: No. There is no need to switch to modules. There has never been a need to switch to modules. Java 9 and later releases support traditional JAR files on the traditional class path, via the concept of the unnamed module, and will likely do so until the heat death of the universe. Whether to start using modules is entirely up to you. If you maintain a large legacy project that isn't changing very much, then it's probably not worth the effort. If you work on a large project that's grown difficult to maintain over the years then the clarity and discipline that modularization brings could be beneficial, but it could also be a lot of work, so think carefully before you begin. If you're starting a new project then I highly recommend starting with modules if you can. Many popular libraries have, by now, been upgraded to be modules, so there's a good chance that all of the dependencies that you need are already available in modular form. If you maintain a library then I strongly recommend that you upgrade it to be a module if you haven't done so already, and if all of your library's dependencies have been converted. All this isn't to say that you won't encounter a few stumbling blocks when

moving past Java-18. Those that you do encounter will, however, likely have nothing to do with modules per se. The most common migration problems that we've heard about since we released Java-9 in 2017 have to do with changes to the syntax of the version string and to the removal or encapsulation of internal APIs (e.g., `sun.misc.Base64Decoder`) for which public, supported replacements have been available for years.

GPT answer, Verb Count: 37 , Adjective Count: 46

GPT-4o Answer: Java 9 introduced the module system (Project Jigsaw), but using it is not mandatory. If your code works without modules and you're not facing any compatibility issues, there's no immediate need to adopt the module system. Here are key considerations: 1. Compatibility Code that doesn't use modules will continue to work in the classpath, as the module system is backward compatible with non-modular applications. You can mix modular and non-modular code, but non-modular code will be treated as part of the "unnamed module." 2. Benefits of Modules Adopting the module system provides benefits like stronger encapsulation, better dependency management, and smaller runtime images through tools like `jlink`. However, these are not mandatory for most applications and are typically useful for large projects with many dependencies. 3. Potential Risks of Not Using Modules While there's no immediate risk, some potential concerns include: - `Reduced security and maintainability: Without modules, it's harder to enforce encapsulation of internal APIs.` - `Limited access to future features: Some new Java features might integrate better with the module system.` - `Compatibility issues in larger, complex ecosystems: If third-party libraries start requiring modules, migration could become more challenging later.` 4. When to Adopt Modules Adopting modules may be worth considering if: - `You're building a new project.` - `Your application is large and has complex dependencies.` - `You want to create runtime images for smaller distributions.` 5. Recommendation If your migration to Java 11 is smooth and you don't require the features provided by modules, there's no urgency to switch. However, keeping an eye on library updates and gradual adoption in future projects could help avoid future compatibility issues.

Co-Pilot, Verb Count: 27 , Adjective Count: 19

Co-Pilot Answer: There's no strict requirement to switch to modules when upgrading from Java 8 to Java 11. Java modules introduced in Java 9 offer several benefits like strong encapsulation, reliable configuration, and improved scalability of large applications. However, for many projects, especially smaller ones or those not requiring modularization benefits, sticking with the traditional classpath approach is perfectly fine. If you don't switch to modules, your code will continue to run on the classpath as before. Here are some considerations: `Maintainability: Modules provide better dependency management and encapsulation. This can be beneficial in large, complex projects.` `Compatibility: Most libraries and frameworks support both classpath and module systems. Some newer libraries might take advantage of modules, but most should still work with classpath-based projects.` `Migration Path: You can migrate to modules incrementally. Start by modularizing one part of your application at a time.` In summary, there's no immediate need to switch to modules if your existing approach works. However, it's worth considering for future-proofing and taking advantage of the benefits modules offer.

Month: Aug-20

Question 1:

Tags: javascript, vue.js, vue-component, vuejs3, vue-composition-api

Question: How to create Event Bus in Vue 3? In Vue 2, it was: `export const bus = new Vue(); bus.$on(...) bus.$emit(...)` In Vue 3, `Vue` is not a constructor anymore, and

`Vue.createApp({});` returns an object that has no `$on` and `$emit` methods.

Answer Score: 222

Stack Overflow answer, Verb Count: 34 , **Adjective Count:** 26

Stack Overflow Answer: As suggested in official docs you could use mitt library to dispatch events between components, let suppose that we have a sidebar and `header` which contains a button that close/open the sidebar and we need that button to toggle some property inside the sidebar component : in main.js import that library and create an instance of that emitter and define as a global property:

Installation : `npm install --save mitt` Usage : `import { createApp } from 'vue'`
`import App from './App.vue'` `import mitt from 'mitt';` `const emitter = mitt();`
`const app = createApp(App);` `app.config.globalProperties.emitter = emitter;`
`app.mount('#app');` in header emit the `toggle-sidebar` event with some payload :
`<template> <header> <button @click="toggleSidebar"/>toggle</button>`
`</header> </template> <script > export default { data() { return {`
`sidebarOpen: true }; }, methods: { toggleSidebar() { this.sidebarOpen =`
`!this.sidebarOpen; this.emitter.emit("toggle-sidebar", this.sidebarOpen); }`
`} }; </script>` In sidebar receive the event with the payload: `<template> <aside`
`class="sidebar" :class="{ 'sidebar--toggled': !isOpen}> ... </aside>`
`</template> <script> export default { name: "sidebar", data() { return {`
`isOpen: true }; }, mounted() { this.emitter.on("toggle-sidebar", isOpen => {`
`this.isOpen = isOpen; }); } }; </script>` For those using composition api they could use
emitter as follows : Create a file `src/composables/useEmitter.js` `import { getCurrentInstance`
`} from 'vue'` `export default function useEmitter() { const internalInstance =`
`getCurrentInstance(); const emitter =`
`internalInstance.appContext.config.globalProperties.emitter; return emitter;`
`}` And from there on you can use `useEmitter` just like you would with `useRouter`: `import`
`useEmitter from '@composables/useEmitter'` `export default { setup() { const`
`emitter = useEmitter() ... } ... }` Using the composition API You could also take benefit
from the new composition API and define a composable event bus : `eventBus.js` `import { ref }`
`from "vue";` `const bus = ref(new Map());` `export default function`
`useEventsBus(){ function emit(event, ...args) { bus.value.set(event, args);`
`} return { emit, bus } }` in component A do: `import useEventsBus from`
`'./eventBus'; ... //in script setup or inside the setup hook const`
`{emit}=useEventsBus() ... emit('sidebarCollapsed',val)` in component B : `const {`
`bus } = useEventsBus()` `watch(()=>bus.value.get('sidebarCollapsed'), (val) =>`
`{ // destruct the parameters const [sidebarCollapsedBus] = val ?? []`
`sidebarCollapsed.value = sidebarCollapsedBus })`

GPT answer, Verb Count: 20 , **Adjective Count:** 15

GPT-4o Answer: In Vue 3, you can create an Event Bus using the `mitt` library, which is a lightweight event emitter. Here's how you can do it: Step 1: Install `mitt` `npm install mitt` Step 2: Create the Event Bus Create a new file, e.g., `eventBus.js`, and initialize the Event Bus: `import mitt from 'mitt';` `export const eventBus = mitt();` Step 3: Use the Event Bus To emit an event: `import { eventBus } from './eventBus';` `eventBus.emit('eventName', payload);` To listen for an event: `import { eventBus } from './eventBus';`
`eventBus.on('eventName', (payload) => { console.log('Event received:',`
`payload); });` To remove an event listener: `eventBus.off('eventName',`
`handlerFunction);` Alternative: Provide/Inject API If you want to stick with Vue's built-in tools, you can use the `provide/inject` API to share the event bus instance across components: In your main application file: `import { createApp } from 'vue';` `import App from './App.vue';`
`import mitt from 'mitt';` `const app = createApp(App);` `app.provide('eventBus',`
`mitt());` `app.mount('#app');` In a component: `// To emit an event const eventBus =`


```
inject('eventBus'); eventBus.emit('eventName', payload); // To listen for an
event eventBus.on('eventName', (payload) => { console.log('Event received:',
payload); });
```

Co-Pilot, Verb Count: 12 , Adjective Count: 7

Co-Pilot Answer: In Vue 3, you can create an Event Bus using the mitt library, which provides a tiny event emitter. Here's how you can set it up: `import mitt from 'mitt'; const emitter = mitt(); export default emitter;` Now you can use the emitter to emit and listen for events in your components: For emitting an event: `import emitter from './path/to/emitter'; emitter.emit('event-name', eventData);` For listening to an event: `import emitter from './path/to/emitter'; emitter.on('event-name', (eventData) => { console.log(eventData); });` This approach allows you to create a centralized event bus in Vue 3, similar to how it was done in Vue 2.

Question 2:

Tags: javascript, environment-variables, node.js, nestjs, dotenv

Question: I am working on my first NestJS application, which was working fine with hardcoded database connecting string in `app.module.ts`. But then as per our requirements, I had to pick the database config values from environment files. For that, I followed the configuration documentation on the nestjs documentation website - <https://docs.nestjs.com/techniques/configuration> But the issue is that I need to use the `.env` variables inside the same file for database connection, which is failing. Here is my original code that was working fine: `@Module({ imports: [MongooseModule.forRoot(`mongodb+srv://myusername:mypassword@myhost.net?retryWrites=true&w=majority&db=dbname`, { useUrlParser: true, dbName: 'dbname' })], ProductModule, CategoryModule,], controllers: [AppController, HealthCheckController,], providers: [AppService, CustomLogger], })` Now, I wanted to pick those DB values from `.env` files which are like `local.env`, `dev.env` etc. depending on the environment. Now, my this code is not working: `@Module({ imports: [ConfigModule.forRoot({ envFilePath: `${process.env.NODE_ENV}.env` }), MongooseModule.forRoot(`mongodb+srv://${process.env.DB_USER}:${process.env.DB_PASS}@myhost.net?retryWrites=true&w=majority&db=dbname`, { useUrlParser: true, dbName: 'dbname' })], ProductModule, CategoryModule,], controllers: [AppController, HealthCheckController,], providers: [AppService, CustomLogger], })`

Answer Score: 89

Stack Overflow answer, Verb Count: 44 , Adjective Count: 42

Stack Overflow Answer: 1. Keeping using ConfigModule You need to set `NODE_ENV` in npm scripts so that it can be used to load an env file based on the env. `"scripts": { "start:local": "NODE_ENV=local npm run start" "start:dev": "NODE_ENV=dev npm run start" }` Now you can just use the ConfigModule: `@Module({ imports: [ConfigModule.forRoot({ envFilePath: `${process.env.NODE_ENV}.env` }), MongooseModule.forRoot(`mongodb+srv://${process.env.DB_USER}:${process.env.DB_PASS}@myhost.net?retryWrites=true&w=majority&db=dbname`, { useUrlParser: true, dbName: 'dbname' })] ... })` 2. Using dotenv npm install dotenv Add some scripts to your package.json to set what env you are in. `"scripts": { ... "start:local": "NODE_ENV=local npm run start" "start:dev": "NODE_ENV=dev npm run start" }` Import dotenv in main.ts file. Make sure you do it at the top of the file. `require('dotenv').config({ path: `../${process.env.NODE_ENV}.env` });` 3. Using env-cmd You can use env-cmd npm package. `npm install env-cmd` And add some commands for different envs in package.json, for example: `"scripts": { ... "start:local": "env-cmd -f local.env npm run start" "start:dev": "env-cmd -f dev.env npm run start" }` ... Now you can use the

env variables, for example: `MongooseModule.forRoot(`mongodb+srv://${process.env.DB_USER}:${process.env.DB_PASS}@myhost.net?retryWrites=true&w=majority&db=dbname`, { useUrlParser: true, dbName: 'dbname' })`
`process.env.MONGO_CONNECTION_STRING` Update: To overcome the env set command problem in different platforms, you can install `cross-env` package. `npm install -D cross-env` And update the scripts: `"scripts": { "start:local": "cross-env NODE_ENV=local npm run start" "start:dev": "cross-env NODE_ENV=dev npm run start" }`

GPT answer, Verb Count: 26 , Adjective Count: 23

GPT-4o Answer: To use environment variables in your NestJS application for dynamic database connection strings, you need to ensure that the `ConfigService` is properly injected and used. The issue with your current code is that you are trying to access environment variables directly via `ConfigModule.get`, which is not how the `ConfigModule` is designed to work. Here is the corrected implementation: Step 1: Update `app.module.ts` `import { Module } from '@nestjs/common'; import { MongooseModule } from '@nestjs/mongoose'; import { ConfigModule, ConfigService } from '@nestjs/config'; import { ProductModule } from '../product/product.module'; import { CategoryModule } from '../category/category.module'; import { ApplicationController } from '../app.controller'; import { HealthCheckController } from '../health-check.controller'; import { AppService } from '../app.service'; import { CustomLogger } from '../custom-logger.service'; @Module({ imports: [ConfigModule.forRoot({ envFilePath: `${process.env.NODE_ENV}.env` }), MongooseModule.forRootAsync({ imports: [ConfigModule], useFactory: async (configService: ConfigService) => ({ uri: `mongodb+srv://${configService.get('DB_USER')}:${configService.get('DB_PASS')}@myhost.net`, useUrlParser: true, dbName: configService.get('DB_NAME'), }), inject: [ConfigService], }), ProductModule, CategoryModule,], controllers: [AppController, HealthCheckController], providers: [AppService, CustomLogger], }) export class AppModule {}` Step 2: Ensure Environment Files Are Correctly Set Up Create your environment files like `local.env` or `dev.env` and include the necessary variables: `DB_USER=myusername DB_PASS=mypassword DB_NAME=dbname` Explanation 1. `MongooseModule.forRootAsync` allows asynchronous configuration of the database connection, making it suitable for retrieving dynamic environment variables. 2. The `ConfigService` is injected into the factory function to access environment variables. 3. The `envFilePath` in `ConfigModule.forRoot` ensures the correct environment file is loaded based on `process.env.NODE_ENV`. With these changes, your application will use the environment variables for the database connection dynamically and correctly.

Co-Pilot, Verb Count: 13 , Adjective Count: 4

Co-Pilot Answer: The issue arises because `ConfigModule.get` is not directly accessible as a method. Instead, you need to use dependency injection to access the configuration service. Here's how to update your code: `import { ConfigModule, ConfigService } from '@nestjs/config'; import { MongooseModule } from '@nestjs/mongoose'; import { Module } from '@nestjs/common'; import { ProductModule } from '../product/product.module'; import { CategoryModule } from '../category/category.module'; import { ApplicationController } from '../app.controller'; import { HealthCheckController } from '../health-check.controller'; import { AppService } from '../app.service'; import { CustomLogger } from '../custom-logger.service'; @Module({ imports: [ConfigModule.forRoot({ envFilePath: `${process.env.NODE_ENV}.env` }), MongooseModule.forRootAsync({ imports: [ConfigModule], useFactory: async (configService: ConfigService) => ({ uri: `mongodb+srv://${configService.get('DB_USER')}:${configService.get('DB_PASS')}@myhost.net`, useUrlParser: true, dbName: configService.get('DB_NAME'), }), inject: [ConfigService], }), ProductModule, CategoryModule,], controllers: [AppController, HealthCheckController], providers: [AppService, CustomLogger], }) export class AppModule {}`

```
'DB_USER'}}:${configService.get('DB_PASS')}@myhost.net?retryWrites=true&w=ma
jority&db=dbname, useUrlParser: true, dbName: 'dbname', })), inject:
[ConfigService], })), ProductModule, CategoryModule, ], controllers: [
AppController, HealthCheckController, ], providers: [AppService,
CustomLogger], }) export class AppModule {}
```

With this approach, the configuration values will be correctly injected and used for setting up the database connection.

Question 3:

Tags: java, intellij-idea, spring-boot

Question: I have cloned a spring boot project from GitLab having multiple branches in IntelliJ. I checkout to the branch in which I want to work. But I am getting "Java file outside of source root" on all the files. Please suggest what I am doing wrong. I just installed IntelliJ and it's the first project that I am importing.

Answer Score: 152

Stack Overflow answer, Verb Count: 12 , **Adjective Count:** 4

Stack Overflow Answer: If you do an 'import from git', IntelliJ doesn't import the project structure from maven (or gradle) automatically. One way to do this afterwards is to right-click on the pom.xml file in the root directory, and select 'Add as maven project'. Then the folders will be marked correctly, and dependent libraries will be imported. There should be no need to mark individual source folders manually. I prefer to do a git clone outside of IntelliJ, and afterwards open the project in IntelliJ from the local filesystem, then IntelliJ imports the maven project structure automatically.

GPT answer, Verb Count: 29 , **Adjective Count:** 9

GPT-4o Answer: The "Java file outside of source root" error occurs because IntelliJ IDEA does not recognize the directories in your project as source roots. To fix this issue, follow these steps: Step 1: Verify Project Structure Go to File > Project Structure or press **Ctrl+Alt+Shift+S**. In the "Project Structure" window: - Navigate to Modules on the left panel. - Check if your source directories (e.g., [src/main/java](#) and [src/test/java](#)) are marked as "Source" and "Test Source" roots respectively. - If not, click the + Add Content Root button, locate your project folder, and mark the appropriate directories as source roots by right-clicking on them and selecting Mark Directory as > Sources Root or Test Sources Root. Step 2: Sync Maven or Gradle Project If the project is a Maven or Gradle project: - Open the "Maven" or "Gradle" tool window in IntelliJ. - Click the Reload All Projects button to ensure all dependencies and configurations are correctly imported. Step 3: Check Module Settings In the "Modules" tab of the "Project Structure" window: - Ensure the module SDK is set correctly under the "Dependencies" tab. - Set it to a valid JDK (e.g., Java 11 or Java 17) if it's missing or incorrect. Step 4: Refresh Project After making the above changes: - Right-click on your project folder in the "Project" tool window and select Reload from Disk. - Alternatively, click File > Invalidate Caches / Restart > Invalidate and Restart. These steps should resolve the "Java file outside of source root" error and allow you to work on your Spring Boot project in IntelliJ.

Co-Pilot, Verb Count: 13 , **Adjective Count:** 3

Co-Pilot Answer: The "Java file outside of source root" warning in IntelliJ indicates that your project structure is not properly set up. To resolve this, you need to mark the correct directories as "source roots". Here are the steps to fix this: [Right-click on the project directory in the Project view. Select "Mark Directory as". Choose "Sources Root" for the main source directory \(usually src/main/java\). If you have test sources, mark the test directory \(usually src/test/java\) as "Test Sources Root".](#) By marking the directories correctly, IntelliJ will recognize them as part of the project source root, and the warning should disappear.

Question 4:

Tags: python, virtualenv

Question: I was creating a new virtual environment on Ubuntu 20.04: `$ virtualenv my_env` But it gave an error: `ModuleNotFoundError: No module named 'virtualenv.seed.embed.via_app_data'` Other info: `$ virtualenv --version`
`virtualenv 20.0.17 from`
`/usr/lib/python3/dist-packages/virtualenv/___init___py`

Answer Score: 65

Stack Overflow answer, Verb Count: 3 , Adjective Count: 2

Stack Overflow Answer: Try to create the virtual environment using directly venv module `python3 -m venv my_env`

GPT answer, Verb Count: 25 , Adjective Count: 27

GPT-4o Answer: The error occurs because the version of `virtualenv` installed on your system is outdated and incompatible with some of its dependencies. To resolve the issue, update `virtualenv` to the latest version using the following steps: Step 1: Upgrade `virtualenv` Run the following command to upgrade `virtualenv`: `pip install --upgrade virtualenv` If you're using Python 3, ensure you are upgrading the correct `virtualenv` version: `pip3 install --upgrade virtualenv` Step 2: Verify the Upgrade Check the installed version of `virtualenv`: `virtualenv --version` Ensure it shows the latest version (e.g., 20.24.2 or higher). Step 3: Create the Virtual Environment Retry creating the virtual environment: `virtualenv my_env` Alternative Solution If you are unable to upgrade `virtualenv` using `pip`, consider reinstalling it: `pip uninstall virtualenv pip install virtualenv` These steps should resolve the issue and allow you to create a new virtual environment without encountering the error.

Co-Pilot, Verb Count: 11 , Adjective Count: 5

Co-Pilot Answer: It seems like the issue is related to the version of `virtualenv` you're using. Try updating `virtualenv` to the latest version: `$ pip install --upgrade virtualenv` If the problem persists, you can try creating the virtual environment using Python's built-in `venv` module: `$ python3 -m venv my_env` This should create the virtual environment without any errors.

Question 5:

Tags: python, conda, python-3.9

Question: I'm trying to install python 3.9 in a conda enviroment. I tried creating a new conda env using the following command, `conda create --name myenv python=3.9` But I got an error saying package not found because python 3.9 is not yet released So, I manually created a folder in `envs` folder and tried to list all envs. But I couldn't get the manually created new environment. So, how do I install python 3.9 in a conda env with all functionalities like `pip` working?

Answer Score: 119

Stack Overflow answer, Verb Count: 24 , Adjective Count: 13

Stack Overflow Answer: To create python 3.11 conda environment use the following command `conda create -n py311 python=3.11 py311 - environment name` Update 3 To create python 3.10 conda environment use the following command `conda create -n py310 python=3.10 py310 - environment name` Update 2 You can now directly create python 3.9 environment using the following command `conda create -n py39 python=3.9 py39 - environment name` Update 1 Python 3.9 is now available in conda-forge. To download the tar file - https://anaconda.org/conda-forge/python/3.9.0/download/linux-64/python-3.9.0-h852b56e_0_cpython.t

ar.bz2 Anaconda Page - <https://anaconda.org/conda-forge/python> As pointed out in the comments, python 3.9 is not yet there on any channels. So, it cannot be install yet via conda. Instead, you can download the python 3.9 executable and install it. Once the installation is done, a new executable will be created for python 3.9 and pip 3.9 will be created. Python: `python3.7 python3.7-config python3.7m python3.7m-config python3.9 python3.9-config pip pip3 pip3.7 pip3.8 pip3.9 pipreqs` In order to install ipython for python 3.9, `pip3.9 install ipython`

GPT answer, Verb Count: 26 , Adjective Count: 23

GPT-4o Answer: To install Python 3.9 in a Conda environment, follow these steps: Step 1: Ensure Conda is Updated Update Conda to the latest version: `conda update -n base -c defaults conda` Step 2: Create a Conda Environment with Python 3.9 Run the following command to create a new environment with Python 3.9: `conda create --name myenv python=3.9` If you still encounter an error saying Python 3.9 is not available, add the `conda-forge` channel: `conda create --name myenv python=3.9 -c conda-forge` Step 3: Activate the Environment Activate the newly created environment: `conda activate myenv` Step 4: Verify Python Version Check the Python version to ensure Python 3.9 is installed: `python --version` Step 5: Install Additional Packages Install essential packages like `pip` if not already included: `conda install pip` Alternative: Use Mamba for Faster Package Installation If Conda installation fails or is slow, you can use `mamba`, a faster Conda alternative: `conda install mamba -n base -c conda-forge` `mamba create --name myenv python=3.9` These steps will ensure Python 3.9 is installed in a functional Conda environment with `pip` and other tools working correctly.

Co-Pilot, Verb Count: 16 , Adjective Count: 4

Co-Pilot Answer: Since Python 3.9 is not yet available in the default Conda channels, you can use the ``conda-forge`` channel to install it. Here's how you can do it: `$ conda create --name myenv python=3.9 -c conda-forge` This command will create a new environment named ``myenv`` with Python 3.9 installed from the ``conda-forge`` channel. After creating the environment, you can activate it and verify the installation: `$ conda activate myenv $ python --version` This should show Python 3.9.x, confirming that it's installed correctly. With this setup, you should have all the functionalities like ``pip`` working as expected.

Month: Sep-20

Question 1:

Tags: javascript, reactjs, react-router, react-router-dom

Question: I am getting the following error when running `npm run start` in the terminal. Attempted import error: 'Redirect' is not exported from 'react-router-dom'. I have reinstalled `node_modules`, `react-router-dom`, `react-router`. Also restarted the terminal and my computer, but the issue persists. My code: `import React from 'react'; import { Switch, Redirect } from 'react-router-dom'; import { RouteWithLayout } from './components'; import { Minimal as MinimalLayout } from './layouts'; import { Login as LoginView, Dashboard as DashboardView, NotFound as NotFoundView } from './views'; const Routes = () => { return (<Switch> <Redirect exact from="/" to="/dashboard" /> <RouteWithLayout component={routeProps => <LoginView {...routeProps} data={data} />} exact layout={MinimalLayout} path="/login" /> <Redirect to="/not-found" /> </Switch>); }; export default Routes; Here is my package.json imports: "react-router": "^6.0.0-beta.0", "react-router-dom": "^6.0.0-beta.0",`

Answer Score: 420

Stack Overflow answer, Verb Count: 1 , Adjective Count: 4

Stack Overflow Answer: For `react-router-dom` v6, simply replace `Redirect` with `Navigate`

```
import { Navigate } from 'react-router-dom'; . . . { component: () =>
<Navigate to="/404" /> }
```

GPT answer, Verb Count: 20 , Adjective Count: 13

GPT-4o Answer: The error occurs because in React Router v6 (including its beta versions), the `Redirect` component has been removed and replaced with the `Navigate` component. To fix this issue, you need to update your code to use `Navigate` instead of `Redirect`. Here's the updated code:

```
import React from 'react'; import { Switch, Route, Navigate } from
'react-router-dom'; import { RouteWithLayout } from './components'; import {
Minimal as MinimalLayout } from './layouts'; import { Login as LoginView,
Dashboard as DashboardView, NotFound as NotFoundView } from './views'; const
Routes = () => { return ( <Switch> <Route path="/" element={<Navigate
to="/dashboard" replace /> } /> <RouteWithLayout path="/login"
layout={MinimalLayout} component={routeProps => <LoginView {...routeProps}
data={data} /> } /> <Route path="*" element={<Navigate to="/not-found"
replace /> } /> </Switch> ); }; export default Routes; Note the following changes: 1.
Import Navigate instead of Redirect. 2. Replace <Redirect> with <Navigate> and use the to
and replace props for redirection. After making these changes, the error should be resolved.
```

Co-Pilot, Verb Count: 13 , Adjective Count: 12

Co-Pilot Answer: The error you're encountering is because in React Router v6, the `Redirect` component has been removed. Instead, you should use the `Navigate` component for navigation. Here is the updated code using `Navigate`:

```
import React from 'react'; import { Routes,
Route } from 'react-router-dom'; import { RouteWithLayout } from
'./components'; import { Minimal as MinimalLayout } from './layouts'; import
{ Login as LoginView, Dashboard as DashboardView, NotFound as NotFoundView }
from './views'; import { Navigate } from 'react-router-dom'; const AppRoutes
= () => { return ( <Routes> <Route path="/" element={<Navigate replace
to="/dashboard" /> } /> <Route path="/login" element={<RouteWithLayout
layout={MinimalLayout} component={LoginView} /> } /> <Route path="*"
element={<Navigate to="/not-found" /> } /> </Routes> ); }; export default
AppRoutes; Make sure to update your package.json to the latest versions of
react-router-dom: "react-router-dom": "^6.x" This change will resolve the issue by using
the updated navigation components in React Router v6.
```

Question 2:

Tags: python, matplotlib

Question: I have used for a long time small subroutines to format axes of charts I'm plotting. A couple of examples:

```
def format_y_label_thousands(): # format y-axis tick labels
formats ax = plt.gca() label_format = '{:,.0f}'
ax.set_yticklabels([label_format.format(x) for x in
ax.get_yticks().tolist()]) def format_y_label_percent(): # format y-axis
tick labels formats ax = plt.gca() label_format = '{:.1%}'
ax.set_yticklabels([label_format.format(x) for x in
ax.get_yticks().tolist()])
```

However, after an update to matplotlib yesterday, I get the following warning when calling any of these two functions: `UserWarning: FixedFormatter should only be used together with FixedLocator`

```
ax.set_yticklabels([label_format.format(x) for x in
ax.get_yticks().tolist()])
```

What is the reason for such a warning? I couldn't figure it out looking into matplotlib's documentation.

Answer Score: 78

Stack Overflow answer, Verb Count: 53 , Adjective Count: 12

Stack Overflow Answer: WORKAROUND: The way to avoid the warning is to use `FixedLocator` (that is part of `matplotlib.ticker`). Below I show a code to plot three charts. I format their axes in different ways. Note that the "set_ticks" silence the warning, but it changes the actual ticks locations/labels (it took me some time to figure out that `FixedLocator` uses the same info but keeps the ticks locations intact). You can play with the x/y's to see how each solution might affect the output.

```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.ticker as mticker
mpl.rcParams['font.size'] = 6.5
x = np.array(range(1000, 5000, 500))
y = 37*x
fig, [ax1, ax2, ax3] = plt.subplots(1,3)
ax1.plot(x,y, linewidth=5, color='green')
ax2.plot(x,y, linewidth=5, color='red')
ax3.plot(x,y, linewidth=5, color='blue')
label_format = '{:,.0f}' # nothing done to ax1 as it is a "control chart." #
# fixing yticks with "set_yticks"
ticks_loc = ax2.get_yticks().tolist()
ax2.set_yticks(ax1.get_yticks().tolist())
ax2.set_yticklabels([label_format.format(x) for x in ticks_loc]) # fixing
# yticks with matplotlib.ticker "FixedLocator"
ticks_loc = ax3.get_yticks().tolist()
ax3.yaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax3.set_yticklabels([label_format.format(x) for x in ticks_loc]) # fixing
# xticks with FixedLocator but also using MaxNLocator to avoid cramped
# x-labels
ax3.xaxis.set_major_locator(mticker.MaxNLocator(3))
ticks_loc = ax3.get_xticks().tolist()
ax3.xaxis.set_major_locator(mticker.FixedLocator(ticks_loc))
ax3.set_xticklabels([label_format.format(x) for x in ticks_loc])
fig.tight_layout()
plt.show()
```

OUTPUT CHARTS: Obviously, having a couple of idle lines of code like the one above (I'm basically getting the yticks or xticks and setting them again) only adds noise to my program. I would prefer that the warning was removed. However, look into some of the "bug reports" (from links on the comments above/below; the issue is not actually a bug: it is an update that is generating some issues), and the contributors that manage matplotlib have their reasons to keep the warning. OLDER VERSION OF MATPLOTLIB: If you use your Console to control critical outputs of your code (as I do), the warning messages might be problematic. Therefore, a way to delay having to deal with the issue is to downgrade matplotlib to version 3.2.2. I use Anaconda to manage my Python packages, and here is the command used to downgrade matplotlib: `conda install matplotlib=3.2.2` Not all listed versions might be available. For instance, couldn't install matplotlib 3.3.0 although it is listed on matplotlib's releases page: <https://github.com/matplotlib/matplotlib/releases>

GPT answer, Verb Count: 23 , Adjective Count: 10

GPT-4o Answer: The warning arises because `set_yticklabels` directly sets the tick labels without modifying the tick locations, which is only appropriate when used with a `FixedLocator`. Matplotlib now discourages this usage, encouraging instead the use of tick formatters to maintain consistency between tick labels and locations. To fix this issue, you should use `FuncFormatter` from the `matplotlib.ticker` module to set the format of tick labels. Here's how you can rewrite your functions:

```
from matplotlib.ticker import FuncFormatter
def format_y_label_thousands():
    ax = plt.gca()
    formatter = FuncFormatter(lambda x, _: f'{x:,.0f}')
    ax.yaxis.set_major_formatter(formatter)
def format_y_label_percent():
    ax = plt.gca()
    formatter = FuncFormatter(lambda x, _: f'{x:.1%}')
    ax.yaxis.set_major_formatter(formatter)
```

Explanation: 1. `FuncFormatter`: It allows you to dynamically format tick labels using a custom formatting function. 2. `set_major_formatter`: It applies the custom formatter to the major ticks of the axis, ensuring that the tick labels are automatically updated to match the tick locations. This approach adheres to the recommended usage in Matplotlib and will eliminate the warning.

Co-Pilot, Verb Count: 16 , Adjective Count: 4

Co-Pilot Answer: The warning "FixedFormatter should only be used together with FixedLocator" occurs because you are setting tick labels without explicitly setting the tick positions. To resolve this, you need to use `FixedLocator` along with `FixedFormatter`. Here's how you can update your code:

```
import matplotlib.pyplot as plt from matplotlib.ticker import FixedLocator, FixedFormatter
def format_y_label_thousands(): # format y-axis tick labels
    formats ax = plt.gca() label_format = '{:,.0f}'
    ax.set_yticks(FixedLocator(ax.get_yticks().tolist())) # Set tick positions
    ax.set_yticklabels([label_format.format(x) for x in ax.get_yticks().tolist()])
def format_y_label_percent(): # format y-axis tick labels
    formats ax = plt.gca() label_format = '{:.1%}'
    ax.set_yticks(FixedLocator(ax.get_yticks().tolist())) # Set tick positions
    ax.set_yticklabels([label_format.format(x) for x in ax.get_yticks().tolist()])
This should resolve the warning and ensure that your tick labels are correctly aligned with the tick positions.
```

Question 3:

Tags: python, youtube, download, python-3.x, youtube-dl

Question: I'm making a little graphic interface with Python 3 which should download a youtube video with its URL. I used the `youtube_dl` module for that. This is my code : `import youtube_dl # Youtube_dl is used for download the video ydl_opt = {"outtmpl" : "/videos/%(title)s.%(ext)s", "format": "bestaudio/best"}` # Here we give some advanced settings. `outtmpl` is used to define the path of the video that we are going to download `def operation(link):` """ Start the download operation """ `try: with youtube_dl.YoutubeDL(ydl_opt) as yd: # The method YoutubeDL() take one argument which is a dictionary for changing default settings video = yd.download([link]) # Start the download result.set("Your video has been downloaded !") except Exception: result.set("Sorry, we got an error.") operation("https://youtube.com/watch?v=...")` When I execute my code, I get this error: `ERROR: YouTube said: Unable to extract video data` I saw here that it was because it doesn't find any video info, how can I resolve this problem?

Answer Score: 208

Stack Overflow answer, Verb Count: 6 , Adjective Count: 4

Stack Overflow Answer: Updating youtube-dl helped me. Depending on the way you installed it, here are the commands: `youtube-dl --update` (self-update) `pip install -U youtube-dl` (via python) `brew upgrade youtube-dl` (macOS + homebrew) `choco upgrade youtube-dl` (Windows + Chocolatey)

GPT answer, Verb Count: 27 , Adjective Count: 18

GPT-4o Answer: The error occurs because the `youtube_dl` module is no longer actively maintained, and YouTube frequently updates its platform, which breaks the ability of `youtube_dl` to extract video data. To resolve this issue, you should use `yt-dlp`, a maintained fork of `youtube_dl` with additional features and better support for YouTube's updates. Here is how you can modify your code to use `yt-dlp` instead: `import yt_dlp as youtube_dl # Replace youtube_dl with yt_dlp ydl_opts = { "outtmpl": "/videos/%(title)s.%(ext)s", "format": "bestaudio/best" }` # Updated dictionary name for clarity `def operation(link):` """ Start the download operation """ `try: with youtube_dl.YoutubeDL(ydl_opts) as yd: # Use yt_dlp's YoutubeDL class yd.download([link]) # Start the download result.set("Your video has been`

```
downloaded!") except Exception as e: result.set(f"Sorry, we got an error: {str(e)}") operation("https://youtube.com/watch?v=...")
```

Steps to fix: 1. Install yt-dlp: `pip install yt-dlp` 2. Update the import statement: Replace `import youtube_dl` with `import yt_dlp as youtube_dl`. 3. Update your code: Ensure that all calls to `youtube_dl` are redirected to `yt_dlp`. This will solve the issue and allow your script to download YouTube videos successfully.

Co-Pilot, Verb Count: 15 , Adjective Count: 9

Co-Pilot Answer: It seems like the error "Unable to extract video data" is due to an outdated or broken version of `youtube_dl`. To resolve this, you should update `youtube_dl` to the latest version: `$ pip install --upgrade youtube_dl` If updating doesn't solve the issue, try reinstalling `youtube_dl`: `$ pip uninstall youtube_dl $ pip install youtube_dl` If the problem persists, consider using the `yt-dlp` module, which is an actively maintained fork of `youtube_dl`: `$ pip install --upgrade yt-dlp` Using `yt-dlp` should resolve the issue and allow you to download YouTube videos without errors.

Question 4:

Tags: java, final, jls, language-lawyer, effectively-final

Question: So far I thought that effectively final and final are more or less equivalent and that the JLS would treat them similar if not identical in the actual behavior. Then I found this contrived scenario:
`final int a = 97; System.out.println(true ? a : 'c'); // outputs a // versus`
`int a = 97; System.out.println(true ? a : 'c'); // outputs 97` Apparently, the JLS makes an important difference between the two here and I am not sure why. I read other threads like Difference between final and effectively final Effectively final variable vs final variable What does a variable being, "effectively final," mean? but they do not go into such detail. After all, on a broader level they appear to be pretty much equivalent. But digging deeper, they apparently differ. What is causing this behavior, can anyone provide some JLS definitions that explain this? Edit: I found another related scenario: `final String a = "a"; System.out.println(a + "b" == "ab"); // outputs true // versus`
`String a = "a"; System.out.println(a + "b" == "ab"); // outputs false` So the string interning also behaves differently here (I don't want to use this snippet in real code, just curious about the different behavior).

Answer Score: 65

Stack Overflow answer, Verb Count: 66 , Adjective Count: 121

Stack Overflow Answer: First of all, we are talking about local variables only. Effectively final does not apply to fields. This is important, since the semantics for `final` fields are very distinct and are subject to heavy compiler optimizations and memory model promises, see §17.5.1 on the semantics of final fields. On a surface level `final` and `effectively final` for local variables are indeed identical. However, the JLS makes a clear distinction between the two which actually has a wide range of effects in special situations like this. Premise From JLS-§4.12.4 about `final` variables: A constant variable is a `final` variable of primitive type or type `String` that is initialized with a constant expression (§15.29). Whether a variable is a constant variable or not may have implications with respect to class initialization (§12.4.1), binary compatibility (§13.1), reachability (§14.22), and definite assignment (§16.1.1). Since `int` is primitive, the variable `a` is such a constant variable. Further, from the same chapter about `effectively final`: Certain variables that are not declared final are instead considered effectively final: ... So from the way this is worded, it is clear that in the other example, `a` is not considered a constant variable, as it is not final, but only effectively final. Behavior Now that we have the distinction, let's look up what is going on and why the output is different. You are using the conditional operator `? :` here, so we have to check its definition. From JLS-§15.25: There are three kinds of conditional expressions, classified according to the second and third operand expressions: boolean conditional expressions, numeric conditional expressions, and reference conditional expressions. In this case, we

are talking about a numeric conditional expressions, from JLS-§15.25.2: The type of a numeric conditional expression is determined as follows: And that is the part where the two cases get classified differently. effectively final The version that is `effectively final` is matched by this rule: Otherwise, general numeric promotion (§5.6) is applied to the second and third operands, and the type of the conditional expression is the promoted type of the second and third operands. Which is the same behavior as if you would do `5 + 'd'`, i.e. `int + char`, which results in `int`. See JLS-§5.6 Numeric promotion determines the promoted type of all the expressions in a numeric context. The promoted type is chosen such that each expression can be converted to the promoted type, and, in the case of an arithmetic operation, the operation is defined for values of the promoted type. The order of expressions in a numeric context is not significant for numeric promotion. The rules are as follows: [...] Next, widening primitive conversion (§5.1.2) and narrowing primitive conversion (§5.1.3) are applied to some expressions, according to the following rules: In a numeric choice context, the following rules apply: If any expression is of type `int` and is not a constant expression (§15.29), then the promoted type is `int`, and other expressions that are not of type `int` undergo widening primitive conversion to `int`. So everything is promoted to `int` as `a` is an `int` already. That explains the output of 97. final The version with the `final` variable is matched by this rule: If one of the operands is of type `T` where `T` is `byte`, `short`, or `char`, and the other operand is a constant expression (§15.29) of type `int` whose value is representable in type `T`, then the type of the conditional expression is `T`. The final variable `a` is of type `int` and a constant expression (because it is `final`). It is representable as `char`, hence the outcome is of type `char`. That concludes the output `a`. String example The example with the string equality is based on the same core difference, `final` variables are treated as constant expression/variable, and `effectively final` is not. In Java, string interning is based on constant expressions, hence `"a" + "b" + "c" == "abc"` is `true` as well (don't use this construct in real code). See JLS-§3.10.5: Moreover, a string literal always refers to the same instance of class `String`. This is because string literals - or, more generally, strings that are the values of constant expressions (§15.29) - are "interned" so as to share unique instances, using the method `String.intern` (§12.5). Easy to overlook as it is primarily talking about literals, but it actually applies to constant expressions as well.

GPT answer, Verb Count: 37 , Adjective Count: 52

GPT-4o Answer: The key difference lies in how the Java Compiler and Java Language Specification (JLS) handle `final` and `effectively final` variables at compile-time. Here are the explanations for both scenarios: Scenario 1: Ternary Operator `final int a = 97; System.out.println(true ? a : 'c');` // outputs 'a' `int a = 97; System.out.println(true ? a : 'c');` // outputs 97 Explanation: When a variable is declared `final`, the compiler treats it as a constant, allowing constant folding and type promotion at compile time. In the ternary operator, if both branches are constants, the compiler evaluates the result using the type of the second operand. For the first case (`final int a`): - `a` is a constant, so the expression `true ? a : 'c'` evaluates as a `char` type with the value `'a'`. For the second case (`int a`): - `a` is not `final`, so it is treated as a variable. The result type of the ternary operator is determined by the common type between `int` and `char`, which is `int`. Scenario 2: String Concatenation and Interning `final String a = "a"; System.out.println(a + "b" == "ab");` // outputs true `String a = "a"; System.out.println(a + "b" == "ab");` // outputs false Explanation: When a `final` variable is used in a string concatenation, the compiler can perform constant folding and concatenate the strings at compile-time, which allows the result to be interned into the string pool. For the first case (`final String a`): - The compiler knows `a` is constant, so it concatenates `"a" + "b"` into `"ab"` at compile-time. The resulting string is interned, and the comparison with `"ab"` returns `true`. For the second case (`String a`): - `a` is not `final`, so the concatenation happens at runtime. The resulting string is not guaranteed to be interned, so the comparison with `"ab"` returns `false`. JLS References: 1. Effectively Final: A variable is considered `effectively final` if it is not modified after its initialization. This concept allows usage in lambda expressions and inner classes. 2. Final Constants: According to the JLS (§15.28), compile-time constants (variables declared `final` and initialized with constant expressions) enable optimizations like constant folding. 3. String Interning: The JLS (§3.10.5) specifies

that string literals are automatically interned, but runtime-generated strings are not guaranteed to be interned unless explicitly done with `String.intern()`.

Co-Pilot, Verb Count: 27 , Adjective Count: 39

Co-Pilot Answer: The difference between `final` and `effectively final` is subtle but significant in certain contexts. In the scenarios you provided, the Java Language Specification (JLS) distinguishes them based on how constant expressions and compile-time constants are evaluated. Here's the detailed explanation: When a variable is declared as `final`, it is treated as a constant expression if it is initialized with a compile-time constant. The JLS (Section 15.28) defines a constant expression as an expression that can be evaluated at compile time. This is why: `final int a = 97; System.out.println(true ? a : 'c');` // outputs 'a' In this case, `a` is a constant expression, and the ternary operator (`?:`) treats `a` as a constant. However, when the variable is not declared as `final` but is effectively final, it is not treated as a constant expression. Instead, it is evaluated at runtime: `int a = 97; System.out.println(true ? a : 'c');` // outputs 97 This difference is due to the JLS's distinction between compile-time constants and effectively final variables that are not explicitly declared as `final`. Regarding the string interning behavior: `final String a = "a"; System.out.println(a + "b" == "ab");` // outputs true In this case, the `final` variable `a` allows the compiler to treat `a + "b"` as a compile-time constant, which is interned by the compiler. Hence, the comparison is true. `String a = "a"; System.out.println(a + "b" == "ab");` // outputs false Here, `a` is not `final`, so the concatenation is performed at runtime, resulting in a new String object that is not interned. Therefore, the comparison is false. These examples illustrate the nuanced differences in behavior due to how the JLS treats `final` and effectively final variables in the context of compile-time constants and string interning.

Question 5:

Tags: javascript, typescript

Question: This is the new error that is coming in typescript code. I am not able to realize the logic behind it Documentation `/*When using the delete operator in strictNullChecks, the operand must now be any, unknown, never, or be optional (in that it contains undefined in the type). Otherwise, use of the delete operator is an error.*/ interface Thing { prop: string; } function f(x: Thing) { delete x.prop; // throws error = The operand of a 'delete' operator must be optional. }`

Answer Score: 262

Stack Overflow answer, Verb Count: 11 , Adjective Count: 16

Stack Overflow Answer: I am not able to realize the logic behind it The logic as I understand is the following: Interface `Thing` is a contract asking to have a (non-null, non-undefined) `prop` as a `string`. If one removes the property, then the contract is not implemented anymore. If you want it still valid when removed, just declare it as optional with a `?:` `prop?: string` I'm actually surprised that this was not causing error in earlier versions of TypeScript.

GPT answer, Verb Count: 19 , Adjective Count: 19

GPT-4o Answer: In TypeScript, when you declare an interface, it acts as a strict contract for the shape of an object. If you define an interface like: `interface Thing { prop: string; }` It means that any object conforming to `Thing` must have a `prop` property, and it must be a `string`. If you remove `prop` from an object, it violates the contract and TypeScript should flag this as an error. If you want the `prop` to be optional, you can define it with a `?` operator: `interface Thing { prop?: string; }` This makes `prop` optional, meaning the object can either have `prop` as a `string` or it can omit it entirely, without violating the contract. The fact that earlier versions of TypeScript didn't throw an error might be due to looser type-checking in those versions or specific compiler options. Modern TypeScript

versions enforce stricter checks to ensure code correctness and better type safety.

Co-Pilot, Verb Count: 22 , Adjective Count: 17

Co-Pilot Answer: The logic behind this behavior in TypeScript is rooted in how interfaces enforce structure and type safety. An interface in TypeScript acts as a contract, specifying that any object adhering to it must have certain properties with specified types. In your example, the `Thing` interface requires a property `prop` of type `string`. When you declare an interface like this: `interface Thing { prop: string; }` Any object of type `Thing` must include the `prop` property, and its value must be a `string`. Removing the property means the object no longer satisfies the interface, which TypeScript will flag as an error. If you want `prop` to be optional, you can use the `?` operator: `interface Thing { prop?: string; }` This way, `prop` can be omitted, and the object will still be considered as adhering to the `Thing` interface. This behavior is consistent with TypeScript's goal of providing strict type safety and helping developers catch potential bugs at compile time.