

Strict Components

Are your components ready to be used in a `strict:true` application?



Strict by default

For all new Angular Applications the typescript flag `strict` is enabled by default.

Including full template checking of @Input / @Output types

```
Error: src/app/app.component.html:1:15 - error TS2322: Type 'string'
  is not assignable to type 'boolean'.
```

```
1 <app-display [disabled]="'WrongType'"></app-display>
```

The problem

Why do we have to talk about this?

Because surprisingly just setting the correct types isn't actually enough!

```
// This will not be enough for strict applications  
@Input()  
public disabled: boolean = false;
```

Our Component

```
export class DisplayComponent {  
  
  @Input()  
  public disabled: boolean = false;  
}
```

Explicit binding of a boolean value is ok

```
<app-display [disabled]="true" ></app-display>
```

But what about supporting plain attributes?

disabled is not boolean

```
<app-display disabled></app-display>
```

```
Error: app.component.html:1:14 - error TS2322: Type 'string' is not assignable to type 'boolean'.
```

Because this is actually equivalent to:

```
<app-display [disabled]=''' '></app-display>
```

Input Coercion

We need to perform Input Coercion to accept plain attributes as boolean.

There are two approaches depending on your Angular / Typescript version

- `ngAcceptInputType_` Angular v9-14
- Set / Get with different Types
 - Typescript [v4.3](#) / Angular v13+

ngAcceptInputType

Static property supported by Angular compiler that enables you to widen the accepted types of an Input property.

```
// Also accept the empty string in addition to boolean values  
static ngAcceptInputType_disabled: boolean | '';  
  
@Input()  
public disabled: boolean = false;
```

Supported Angular v9-14

ngAcceptInputType

```
static ngAcceptInputType_disabled: boolean | '';
```

This code now compiles with no errors.

```
<app-display disabled ></app-display>
```

But we have to remember to convert the empty string to true!

ngAcceptInputType Input Coercion via OnChanges

Convert as part of update from `ngOnChanges`

```
ngOnChanges(changes: SimpleChanges) {  
  if (changes.disabled) {  
    this._disabled = toBoolean(changes.disabled.currentValue);  
  }  
}  
  
toBoolean(value: boolean | string) {  
  this._disabled = (value === '') || value === true;  
}
```

ngAcceptInputType Examples

Works for other types too.

Accept Dates as strings for example

```
static ngAcceptInputType_date: Date | string;

@Input()
public date: Date;

ngOnChanges(changes: SimpleChanges) {
  if (changes.date) {
    this._date = toDate(changes.date.currentValue);
  }
}
```

Set / Get with different Types

Typescript 4.3 supports different Set and Get types

```
_disabled: boolean = false;

@Input()
get disabled(): boolean {
  return this._disabled;
}
set disabled(value: boolean | string) {
  this._disabled = toBoolean(value);
}
```

No need for `ngAcceptInputType` anymore hence its deprecation

What about async pipe?

Now what if our user wants to set disabled state from an Observable stream?

```
<app-display [disabled]="disabledStream$ | async"></app-display>
```

Another error!

```
Type 'null' is not assignable to type 'boolean | '''
```

Async pipe can return `null` if no value emitted yet.

Supporting async pipe with ngAcceptInputType_

Update our `ngAcceptInputType_disabled` to handle null too.

```
static ngAcceptInputType_disabled: boolean | '' | null;
```

Validate that your `toBoolean(value: any)` function handles null

Supporting async pipe with Set / Get

Update our setter to handle null.

```
set disabled(value: boolean | string | null) {  
  this._disabled = toBoolean(value);  
}
```

What if its not your component?

Non null assertion

```
<app-display [disabled]="(disabledStream$ | async)!"
```

Disable type checking with `$any()`

```
<app-display [disabled]="$any(disabledStream$ | async)"
```

Provided a default value

```
<app-display [disabled]="(disabledStream$ | async) || false"
```


Tweak tsConfig angularCompilerOptions

Stop strict null errors just for Inputs

```
"angularCompilerOptions": { "strictNullInputTypes": false }
```

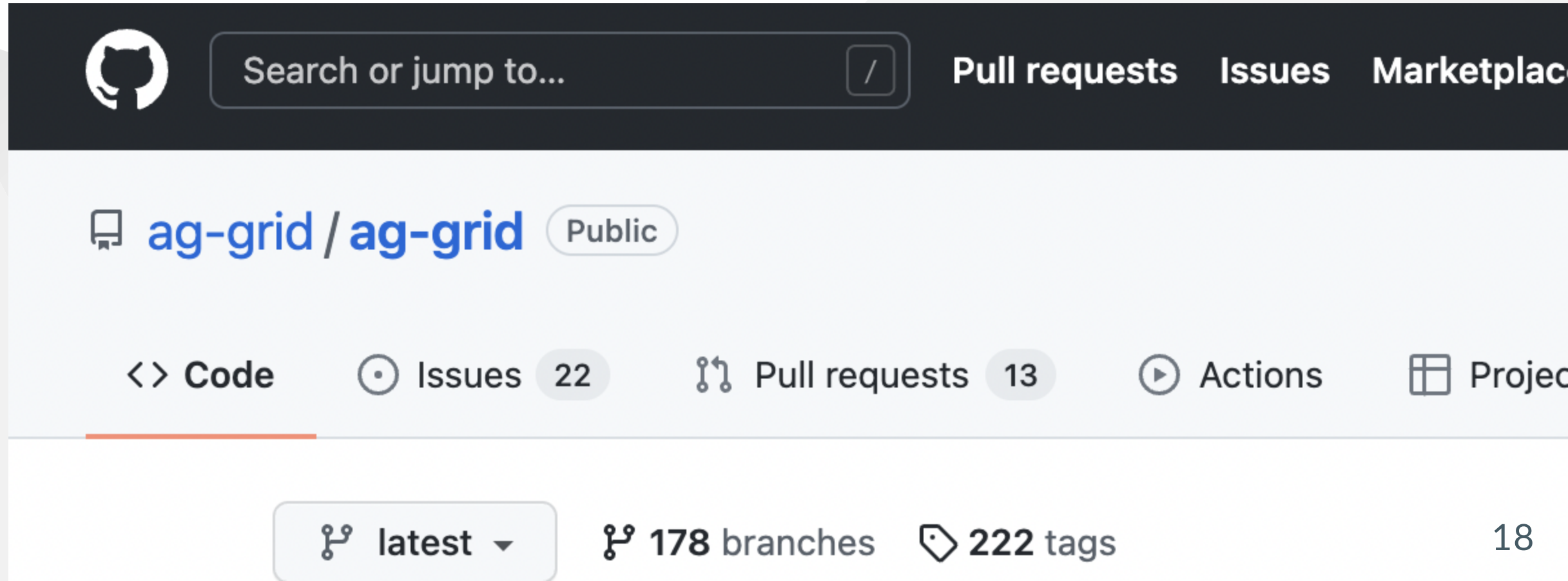
Turn off all input checking. (Not recommended)

```
"angularCompilerOptions": { "strictInputTypes": true }
```

Full list available under [Troubleshooting template errors](#)

What if its not your component?

Create an Issue for the library to fix their types! :)



The screenshot shows the GitHub interface for the `ag-grid/ag-grid` repository. At the top, there is a dark navigation bar with the GitHub logo, a search bar labeled "Search or jump to...", and links for "Pull requests", "Issues", and "Marketplace". Below this, the repository name `ag-grid / ag-grid` is displayed in blue, followed by a "Public" badge. A secondary navigation bar contains tabs for "Code", "Issues" (which is selected and highlighted with a red underline and a count of 22), "Pull requests" (with a count of 13), "Actions", and "Projects". At the bottom, there is a section showing the repository's state: a dropdown menu set to "latest", "178 branches", and "222 tags".

GitHub logo

Search or jump to...

Pull requests

Issues

Marketplace

ag-grid / ag-grid

Public

<> Code

Issues 22

Pull requests 13

Actions

Projects

latest

178 branches

222 tags

