Pentalog

A FULL-STACK NEARSHORE IT GROUP SERVING COMPANIES FROM STARTUPS TO FORTUNE 500s

Angular Performance Tips

Alexandru Bereghici
Pentalog

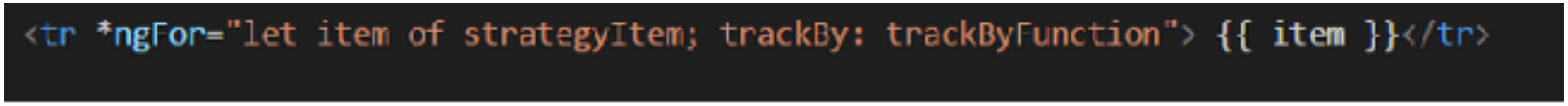
WHY PERFORMANCE MATTERS?

TOPICS

- NgFor - TrackBy
- Subscriptions
- Change Detection
- Pipes
- NgZone
- Computing Values
- Dynamic Imports
- Web Workers
- Webpack Bundle Analyser
- Angular CLI Budgets
- Lazy Loading
- Intersection Observer

NGFOR - TRACKBY

When Angular displays an array , it removes all the DOM elements that associated with the data and create them again. That means a lot of DOM manipulations especially in a case of
a big collection, and as we know, DOM manipulations are expensive.



```
<tr *ngFor="let item of strategyItem; trackBy: trackByFunction"> {{ item }}</tr>
```

WHEN YO USE TRACKBY

- Iterating over large array of objects collection
- In case your business logic might need to modify any of these elements through reordering, modifying specific item, deleting item, or adding a new one.

SUBSCRIPTIONS

- Every time a component or directive is destroyed, the subscription to observable remains active. So it is important to unsubscribe from it to release the memory in the system, otherwise, you will have memory leak.

SUBSCRIPTIONS

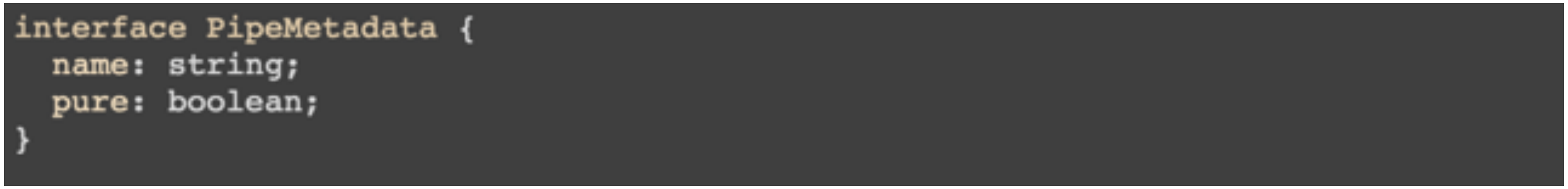
*PS: You don't need to worry about unsubscribing if you are using **AsyncPipe** because it unsubscribes automatically. Also, methods than take(n), takeWhile(predicate), first() and first(predicate) unsubscribes by their-self.*

MINIMIZE CHANGE DETECTIONS

By default, components in an Angular application undergo change detection with nearly every user interaction. But, Angular allows you take control of this process. You can indicate to Angular if a component subtree is up to date and exclude it from change detection.

PURE PIPES

As argument the @Pipe decorator accepts an object literal with the following format:



```
interface PipeMetadata {
  name: string;
  pure: boolean;
}
```

PURE PIPES

The pure flag indicates that the pipe is not dependent on any global state and does not produce side-effects. This means that the pipe will return the same output when invoked with the same input. This way Angular can cache the outputs for all the input parameters the pipe has been invoked with, and reuse them in order to not have to recompute them on each evaluation.

The default value of the pure property is true.

NGZONE

We can take advantage of Zone APIs, to execute our code outside the Angular zone, which will prevent Angular from running unnecessary change detection tasks.

COMPUTING VALUES IN THE TEMPLATE

Sometimes you need to transform a value that comes from the server to something you can display in the UI.
For example:

```
<table>
  <tr *ngFor="let skill of skills">{{skill.calcSomething(skill)}}</tr>
</table>
```

The problem is, because Angular needs to re-run your function in every change detection cycle, if the function performs expensive tasks, it can be costly.

DYNAMIC IMPORTS

TypeScript 2.4 added support for dynamic import() expressions, which allow us to asynchronously load and execute **ECMAScript modules** on demand.

This means that we can conditionally and lazily import other modules and libraries.

WEB WORKERS

JavaScript is a *single-threaded* language, meaning that all your operations are run by the very same thread. If you perform compute intensive tasks on the thread, the rest of the tasks will have to way for those to be finished, leading to sloppy UI rendering and whatnot.

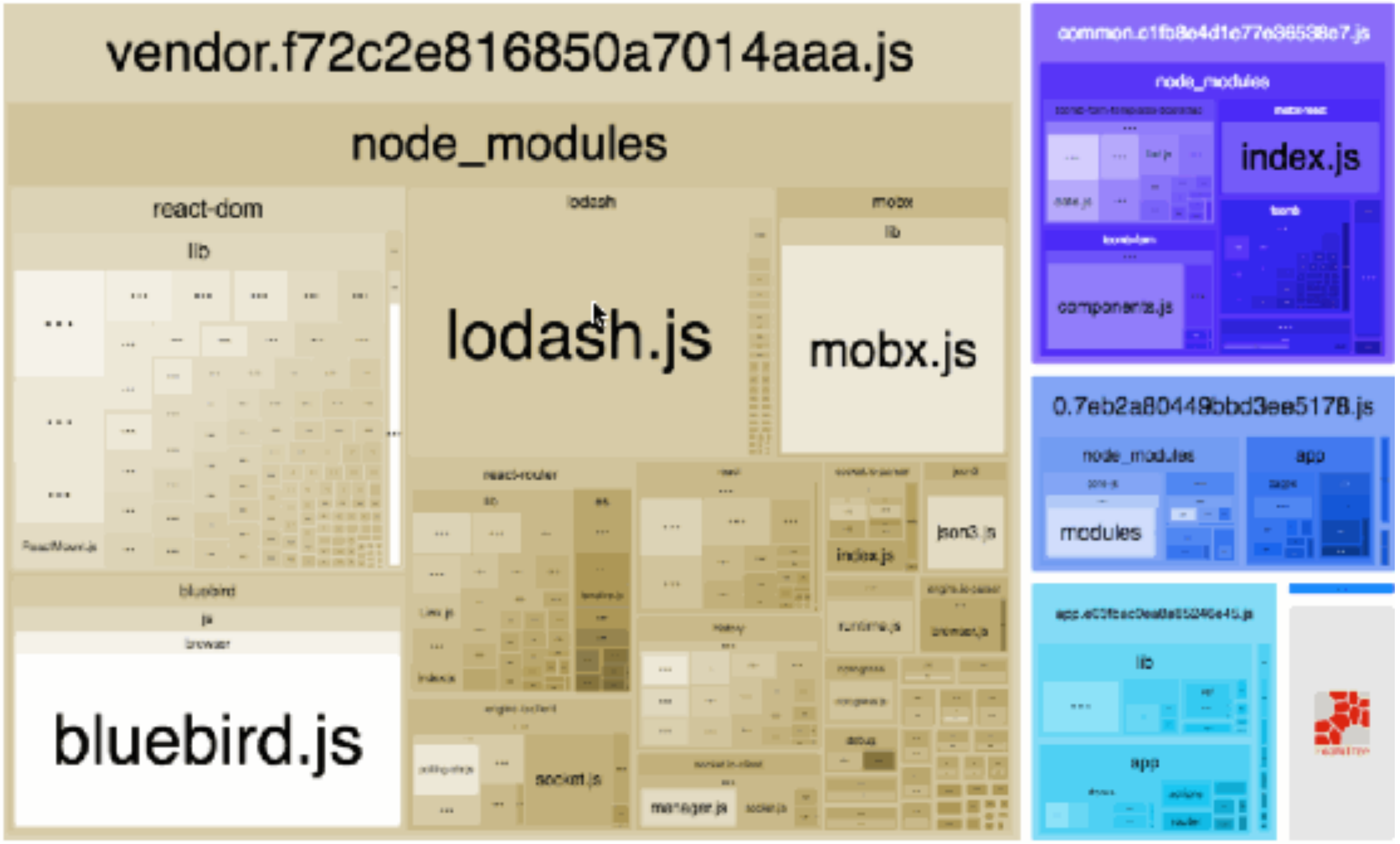
WEBPACK BUNDLE ANALYSES

It will create an interactive treemap visualization of the contents of all your bundles.

This module will help you:

1. Realize what's *really* inside your bundle
2. Find out what modules make up the most of its size
3. Find modules that got there by mistake
4. Optimize it!

WEBPACK BUNDLE ANALYSES

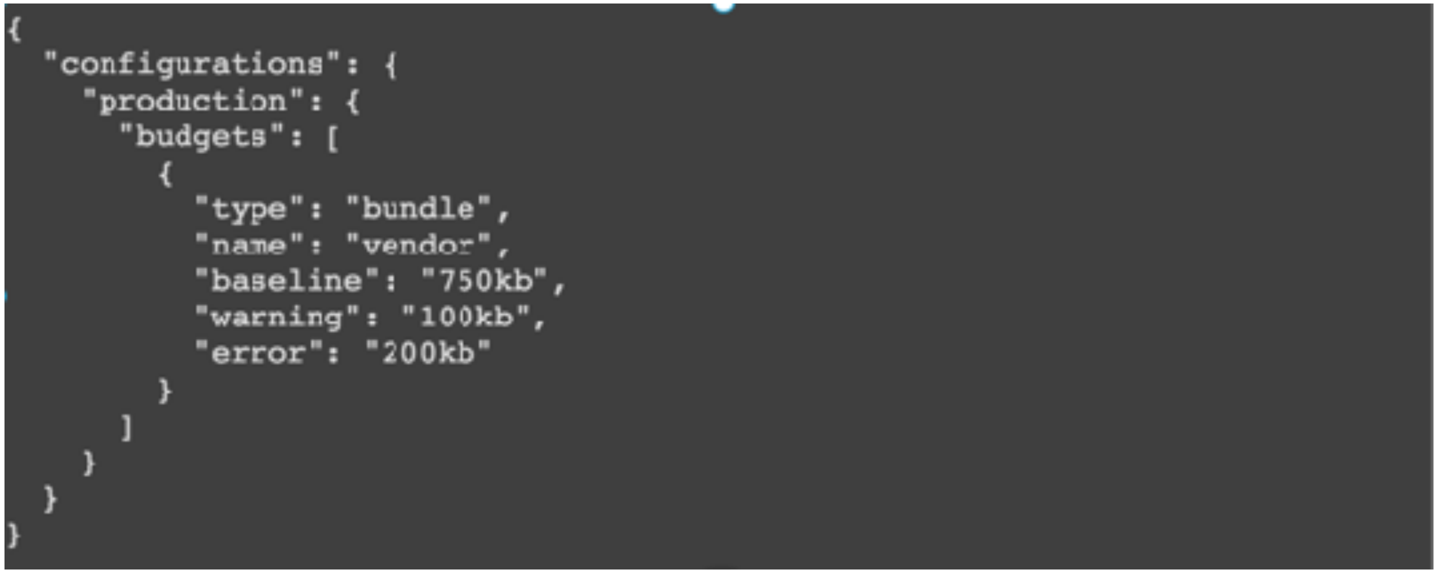


ANGULAR CLI BUDGETS

As applications grow in functionality, they also grow in size. Budgets is a feature in the Angular CLI which allows you to set budget thresholds in your configuration to ensure parts of your application stay within boundaries which you set

ANGULAR CLI BUDGETS

Angular budgets are defined in the angular.json file. Budgets are defined per project which makes sense because every app in a workspace has different needs.



```
{
  "configurations": {
    "production": {
      "budgets": [
        {
          "type": "bundle",
          "name": "vendor",
          "baseline": "750kb",
          "warning": "100kb",
          "error": "200kb"
        }
      ]
    }
  }
}
```

LAZY LOADING

In case the target application has a huge code base with hundreds of dependencies, the practices listed above may not help us reduce the bundle to a reasonable size (reasonable might be 100K or 2M, it again, completely depends on the business goals).

LAZY LOADING

Add a CanLoad guard that only loads the AdminModule once the user is logged in and attempts to access the admin feature area.

```
{
  path: 'admin',
  loadChildren: './admin/admin.module#AdminModule',
  canLoad: [AuthGuard]
},
```

LAZY LOADING IMAGES WITH INTERSECTION API

The Intersection Observer API provides a way to asynchronously observe changes in the intersection of a target element with an ancestor element or with a top-level document's viewport.

QUESTIONS ?