Google Summer of Code 2022

About You

My name is Jeevesh Garg. My major is Civil Engineering, and I am a second-year undergraduate at Indian Institute of Technology, Varanasi (IIT BHU). I am an active part of our institute's coding club (COPs), contribute and maintain many websites of our institute.

What project are you applying for?

Improve the frontend type system

Why are you interested in working with Oppia, and on your chosen project?

Because Oppia is such a vast community with contributions from all around the world, and because its aim is to "help anybody learn anything they desire in an effective way," I feel compelled to contribute more to its development. I'd like to contribute to this idea as well, so that everyone in the globe has access to high-quality, scalable courses. Working with Oppia taught me a lot in just a few months, and I plan to continue contributing to this fantastic community when the GSoC time finishes. The mentors have been quite helpful and have assisted me in improving my technical and coding abilities. I learned a lot about Open Source and project management while working on Oppia. As a result, I'd like to use this chance to create something meaningful for the company. I started with Oppia as part of a Hacktober fest, which sparked my interest in Open Source.

Working with the Angular framework under the Migration Team and LaCE Team over the last few months was a lot of fun. Migrate all interactions and Schema-based-editor with this project mentor, e.g. Mridul Setia and Srijan Reddy VasaI have learned a lot. I believe the initiative would provide more information on the subject. The developers play a vital part in the site's progress, and I believe that after the site is finished, the developers will be able to focus on other projects. I'd like to study more and enhance my knowledge now that I've spent time learning the principles, and this project is ideal for doing exactly that. It would allow me to solve problems involving strict typing that I have yet to encounter.

Prior experience

For the past 5 months, I've been contributing to oppia and working with Angular 8+, AngularJS and Python. Now I am familiar with the majority of the codebase. I also contributed to many projects of my institute, primarily working with frontend frameworks e.g. Angular/Vue/React and backend with Python(Django/SQL). In my spare time, I like doing competitive programming as a hobby.

1. In Oppia, I am a part of the angular migration team and LaCE team. Since November 2021, I've been contributing to the migration project on a regular basis and some current releases of the LaCE. My efforts include writing PRs, and assisting new devs as an Onboarding mentor.
   - I currently have 18 merged PRs and 3 open PRs.*(Last updated March 10th, 2022)*
   - Exposure: Python, TypeScript, Webpack, Karma & Jasmine, Angular, CI/CD.

2. I did an internship in the Training & Placement Cell of IIT(BHU). Maintain its main server and make a new portal for preference choice of participants from scratch.*(Code Repo is private)*
    - Exposure: Django Celery(Redis server), Python template, REST APIs, React.

3. Maintain many websites and apps backend end points of different projects of Indian Institute of Technology, Varanasi (IIT BHU).
    - COPs Hackalog: The home for Hackathons organized under COPs IIT(BHU). (4 PRs)
    - Lite-hai: App monitor all Extracurricular activities & Clubs information. (4 PRs)
    - Technex: Asia's oldest techno-management fest website. (40+ commits & 10 PR)
    - Exposure: Angular, Vue, REST APIs..

Link to my PRs for Oppia:

- #14774: Fix Part of #9749: Migrate and Redesign Drag and Drop interaction.
- #15025: Fixes #15005: Topic prerequisite skill checking is broken and prevents topic being published.
- #14546: Fix part of #9749: Migrate translation-suggestion-review-modal.
- #14766: Fix Part of #9749: Migrate End exploration interaction with separate backend-api-service
- #15089: Fix part of #10474: Make typescript checks strict for some files.

The complete list of PRs created by me is here.
I have opened 4 issues regarding bugs, and 1 regarding formatting, those can be found here.
I did volunteer work in migrating Schema-based-editor with Srijan Reddy Vasa , related PRs.

Project size

I am applying for a large project (~350 hours).

Project timeframe

June, 2022 to 12 September, 2022

Contact info and timezone(s)

Primary Email and Hangout: jeevesh.garg.cd.civ20@itbhu.ac.in
Secondary Email: gargjeevesh60@gmail.com
Contact Number: +91 9896270452
Github: @JeeveshGarg
TimeZone: Kolkata,India (GMT+5:30)
Preferred mode of Communication: Google Chat, Discord, Email.

Time commitment

- From the 13th of June through the 12th of September, I would be working on the GSoC project for 13 weeks.
- I would commit at least 3-4 hrs per day or 20 hrs per week during the coding period, or more if required.

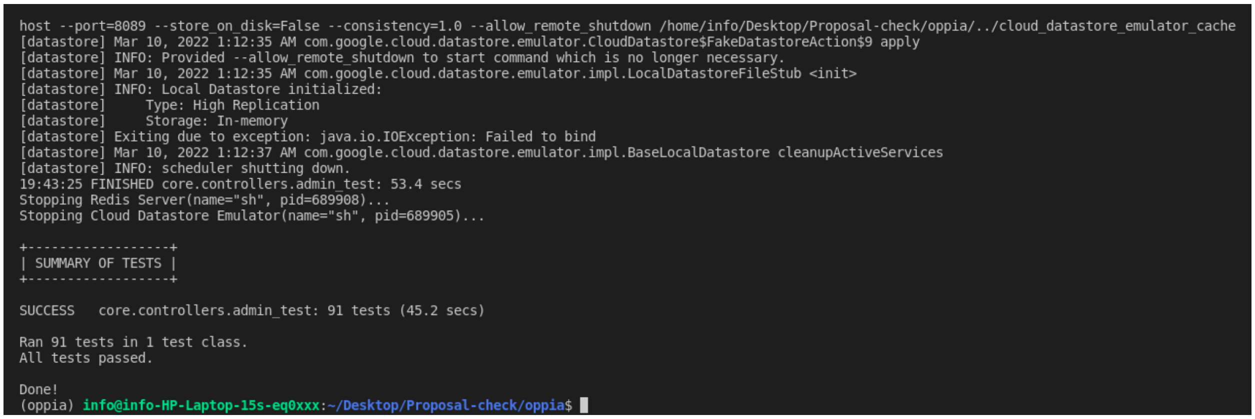
S. No	Dates (Week)	Days(Total)	Time Commitment
1.	13th June - 18th June	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
2.	20th June - 25th June	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
3.	27th June - 2nd July	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
4.	4th July - 9th July	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
5.	11th July - 16th July	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
6.	18th July - 23rd July	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
7.	25th July - 30th July	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
8.	1st Aug - 6th Aug	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
9.	8th Aug - 13th Aug (M1)	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
10.	15th Aug - 20th Aug	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
11.	22nd Aug - 27th Aug	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
12.	29th Aug - 3rd Sep	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
13.	5th Sep - 10th Sep	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
14.	12th Sep - 17th Sep	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
15.	19th Sep - 24th Sep	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
16.	26th Sep - 1st Oct	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
17.	3rd Oct - 8th Oct	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
18.	10th Oct - 15th Oct	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)
19.	16th Oct - 21st Oct (M2)	Mon - Sat(6)	3-4h/day - 20h/week (except Sun)

Total Working Days Estimated: 114
Total Working Hours Estimated: 350+ hours (This may alter depending on the situation)

Note: Last updated May 27th 2022*

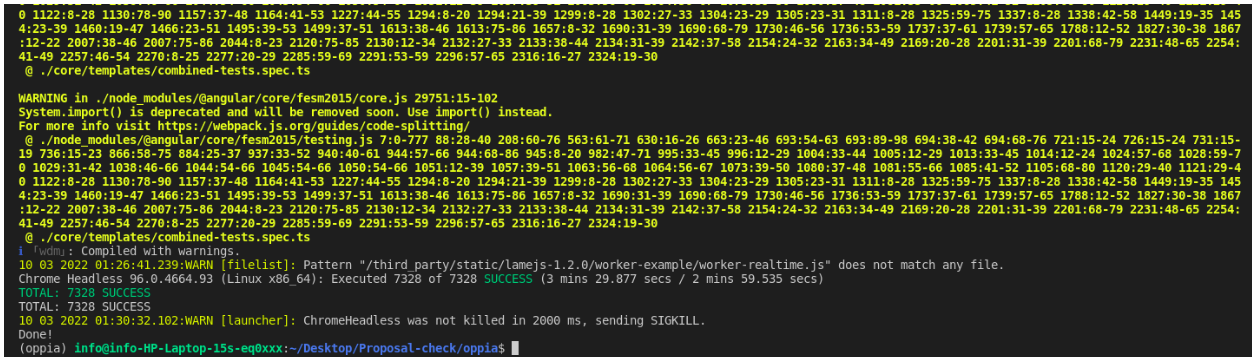
Essential Prerequisites

- I am able to run a single backend test target on my machine.



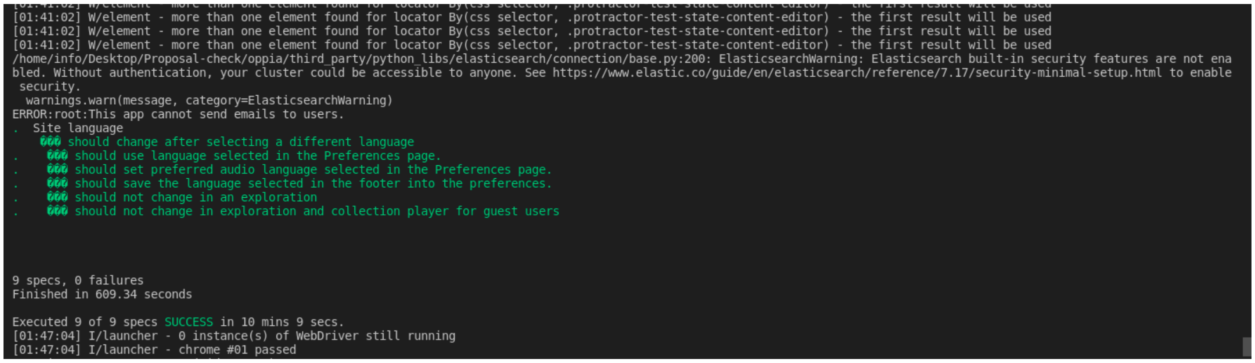
```
host --port=8089 --store_on_disk=False --consistency=1.0 --allow_remote_shutdown /home/info/Desktop/Proposal-check/oppia/../cloud_datastore_emulator_cache
[datastore] Mar 10, 2022 1:12:35 AM com.google.cloud.datastore.emulator.CloudDatastore$FakeDatastoreAction$9 apply
[datastore] INFO: Provided --allow_remote_shutdown to start command which is no longer necessary.
[datastore] Mar 10, 2022 1:12:35 AM com.google.cloud.datastore.emulator.impl.LocalDatastoreFileStub <init>
[datastore] INFO: Local Datastore initialized:
[datastore]     Type: High Replication
[datastore]     Storage: In-memory
[datastore] Exiting due to exception: java.io.IOException: Failed to bind
[datastore] Mar 10, 2022 1:12:37 AM com.google.cloud.datastore.emulator.impl.BaseLocalDatastore cleanupActiveServices
[datastore] INFO: scheduler shutting down.
19:43:25 FINISHED core.controllers.admin_test: 53.4 secs
Stopping Redis Server(name="sh", pid=689908)...
Stopping Cloud Datastore Emulator(name="sh", pid=689905)...

+------------------+
| SUMMARY OF TESTS |
+------------------+

SUCCESS   core.controllers.admin_test: 91 tests (45.2 secs)

Ran 91 tests in 1 test class.
All tests passed.

Done!
(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ 
```

- I am able to run all the frontend tests at once on my machine.



```
0 1122:8-28 1130:78-90 1157:37-48 1164:41-53 1227:44-55 1294:8-20 1294:21-39 1299:8-28 1302:27-33 1304:23-29 1305:23-31 1311:8-28 1325:59-75 1337:8-28 1338:42-58 1449:19-35 145
4:23-39 1460:19-47 1466:23-51 1495:39-53 1499:37-51 1613:38-46 1613:75-86 1657:8-32 1690:31-39 1690:68-79 1730:46-56 1736:53-59 1737:37-61 1739:57-65 1788:12-52 1827:30-38 1867
:12-22 2007:38-46 2007:75-86 2044:8-23 2120:75-85 2130:12-34 2132:27-33 2133:38-44 2134:31-39 2142:37-58 2154:24-32 2163:34-49 2169:20-28 2201:31-39 2201:68-79 2231:48-65 2254:
41-49 2257:46-54 2270:8-25 2277:20-29 2285:59-69 2291:53-59 2296:57-65 2316:16-27 2324:19-30
 @ ./core/templates/combined-tests.spec.ts

WARNING in ./node_modules/@angular/core/fesm2015/core.js 29751:15-102
System.import() is deprecated and will be removed soon. Use import() instead.
For more info visit https://webpack.js.org/guides/code-splitting/
 @ ./node_modules/@angular/core/fesm2015/testing.js 7:0-777 88:28-40 208:60-76 563:61-71 630:16-26 663:23-46 693:54-63 693:89-98 694:38-42 694:68-76 721:15-24 726:15-24 731:15-
19 736:15-23 866:58-75 884:25-37 937:33-52 940:40-61 944:57-66 944:68-86 945:8-20 982:47-71 995:33-45 996:12-29 1004:33-44 1005:12-29 1013:33-45 1014:12-24 1024:57-68 1028:59-7
0 1029:31-42 1038:46-66 1044:54-66 1045:54-66 1050:54-66 1051:12-39 1057:39-51 1063:56-68 1064:56-67 1073:39-50 1080:37-48 1081:55-66 1085:41-52 1105:68-80 1120:29-40 1121:29-4
0 1122:8-28 1130:78-90 1157:37-48 1164:41-53 1227:44-55 1294:8-20 1294:21-39 1299:8-28 1302:27-33 1304:23-29 1305:23-31 1311:8-28 1325:59-75 1337:8-28 1338:42-58 1449:19-35 145
4:23-39 1460:19-47 1466:23-51 1495:39-53 1499:37-51 1613:38-46 1613:75-86 1657:8-32 1690:31-39 1690:68-79 1730:46-56 1736:53-59 1737:37-61 1739:57-65 1788:12-52 1827:30-38 1867
:12-22 2007:38-46 2007:75-86 2044:8-23 2120:75-85 2130:12-34 2132:27-33 2133:38-44 2134:31-39 2142:37-58 2154:24-32 2163:34-49 2169:20-28 2201:31-39 2201:68-79 2231:48-65 2254:
41-49 2257:46-54 2270:8-25 2277:20-29 2285:59-69 2291:53-59 2296:57-65 2316:16-27 2324:19-30
 @ ./core/templates/combined-tests.spec.ts
ℹ ｢wdm｣: Compiled with warnings.
10 03 2022 01:26:41.239:WARN [filelist]: Pattern "/third_party/static/lamejs-1.2.0/worker-example/worker-realtime.js" does not match any file.
Chrome Headless 96.0.4664.93 (Linux x86_64): Executed 7328 of 7328 SUCCESS (3 mins 29.877 secs / 2 mins 59.535 secs)
TOTAL: 7328 SUCCESS
TOTAL: 7328 SUCCESS
10 03 2022 01:30:32.102:WARN [launcher]: ChromeHeadless was not killed in 2000 ms, sending SIGKILL.
Done!
(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ 
```

- I am able to run one suite of e2e tests on my machine.



```
[01:41:02] W/element - more than one element found for locator By(css selector, .protractor-test-state-content-editor) - the first result will be used
[01:41:02] W/element - more than one element found for locator By(css selector, .protractor-test-state-content-editor) - the first result will be used
[01:41:02] W/element - more than one element found for locator By(css selector, .protractor-test-state-content-editor) - the first result will be used
/home/info/Desktop/Proposal-check/oppia/third_party/python_libs/elasticsearch/connection/base.py:200: ElasticsearchWarning: Elasticsearch built-in security features are not ena
bled. Without authentication, your cluster could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.17/security-minimal-setup.html to enable
 security.
  warnings.warn(message, category=ElasticsearchWarning)
ERROR:root:This app cannot send emails to users.
.  Site language
    ��� should change after selecting a different language
.    ��� should use language selected in the Preferences page.
.    ��� should set preferred audio language selected in the Preferences page.
.    ��� should save the language selected in the footer into the preferences.
.    ��� should not change in an exploration
.    ��� should not change in exploration and collection player for guest users



9 specs, 0 failures
Finished in 609.34 seconds

Executed 9 of 9 specs SUCCESS in 10 mins 9 secs.
[01:47:04] I/launcher - 0 instance(s) of WebDriver still running
[01:47:04] I/launcher - chrome #01 passed
```

Other summer obligations

I don't have any summer plans right now.

Communication channels

I intend to communicate with my mentor on a weekly basis for progress updates and on an as-needed basis throughout the project.

- Google Chat
- Google Meet
- Zoom
- Gmail

Alternatively, the mentor might use whatever other platform he likes.

Proposal Details

Problem Statement

Link to PRD (or N/A if there isn't one)	**Improve the Frontend Type System**
Target Audience	**The Oppia Team's developers**
Core User Need	● The project's goal is to make typescript checks strict for 560 files (280 twin files) and to alter the typescript config file so that any newly added files must be typed strictly. This will result in less error-prone code, faster debugging and reworking, and improved execution performance. ● Customize the error log in terminal when we use the command "python -m scripts.typescript_checks --strict_checks". Currently running the command only displays error logs, but not the number of errors. Adding this feature will assist developers in identifying problems linked to strict typescript
What goals do we want the solution to achieve?	All assigned files must pass the typescript strict checks, which will give the code base greater structure, make it self-documenting, more legible, and will help in catching potential flaws.

Section 2.1: WHAT

Key User Stories and Tasks

#	Title	User Story Description (role, goal, motivation) *"As a …, I need …, so that …."*	Priority	List of tasks needed to achieve the goal (this is the "User Journey")	Links to mock prototypes, and/or PRD sections that spec out additional requirements.
1	Typescript strict typing	As an Oppia developer, I'd want the typescript files to be completely typed so that the code is self-documenting, causes fewer defects, and saves time debugging mistakes during runtime.	High	Change the typescript-config file and accompanying typescript checks.py to require strict typing for all newly added files.	N/A
				Apply strict typing to all 280 files and tests that relate to them. (560 files)	N/A
2	Number of errors in addition to the error logs in terminal	As an Oppia developer, I want to see the number of errors in addition to the error logs, so that I can focus on a specific one while dealing with a large number of faults.	Low	Change the typescript checks.py to customize error logs. And show a number of error logs.	N/A
3	remove all the usage of `unknown` type	As an Oppia developer, I want all variables to be properly typed, no type conversion used using unknown.	High	Add lint check in js_ts_linter.py to ensure no more unknown type is used.	N/A
				Remove unknown type from all files.	

Oppia is a fully typed codebase that uses Angular as the frontend framework and Typescript as the primary language. This means that variable assignment, procedure parameters, and function return values will all be clearly connected with a type.

Furthermore, because these type checks are implemented throughout the compilation process, exceptions and mistakes are more likely to occur. As a result, explicit typing makes code self-documenting, creates fewer problems, improves code understanding, and lowers the time spent on troubleshooting issues at runtime.

`strict` is a typescript compiler option that turns on the following set of rules (strict mode):

- **noImplicitAny**
  The implicit type 'any' cannot be used in variables or function parameters.
- **noImplicitThis**
  The context of 'this' cannot be implicitly defined.
- **strictNullChecks**
  Only if a value is explicitly designated as null or undefined may it be null or undefined.
- **strictPropertyInitialization**
  A function Object() { [native code] } [constructor] or property initializer must be used to initialize all class properties.
- **strictBindCallApply**
  Check the 'bind', 'call', and 'apply' functions more thoroughly.
- **strictFunctionTypes**
  Bivariate argument types aren't possible.

Enabling the following constraints reduces the likelihood of surprising results and makes the code more resilient. However, Oppia's code base presently fails to meet these tight requirements. As a result, the code is vulnerable to unexpected actions and failures.

The following activities must be performed to avoid this:

1. Typescript strict mode should be enabled for any new files being added.
2. Strict typing should be implemented to the code base's existing files.

Urgency Requirements

- In the first week of the GSoC coding cycle, certain early parts of this project (1st PR), where we implement severe strict typescript checks to all existing files and newly contributed files, must be merged into the repository. Because the whole later part of this project is dependent on this part and more importantly we are applying a mechanism in 1st PR that all newly-added files need to be strictly typed, so if any other contributor is making new files he must be aware of this new feature and make its file strictly typed.
- So, in order to keep other contributors in sync, merging this section under the urgency requirement would be ideal.

Connection to Existing Work

- Currently, In issue #10474 We add files in tsconfig-strict.json that need to be of a strict type, Which is no longer needed (as mentioned in the "How" section).

Other Requirements

None.

Section 2.2: HOW

Existing Status Quo

The Oppia codebase currently has two typescript config files; tsconfig.json and tsconfig-strict.json.

- **tsconfig-strict.json**: When the compiler option strict is set to true, the noImplicitAny, noImplicitThis, strictNullChecks, strictPropertyInitialization, strictBindCallApply, and strictFunctionTypes rules are enabled. It currently shows a list of specific files that have passed the typescript tests. Since the compiler would throw a huge number of errors if all of the files were covered at once, therefore this list has been updated incrementally after applying the severe restrictions to individual files. As a result of this setup, strict mode only executes on the files/directories listed in the tsconfig-strict.json.

  Command used: **python -m scripts.typescripts_checks --strict_checks**

- **tsconfig.json**: These stringent constraints are deactivated in tsconfig.json by setting noImplicitUseStrict to true in the compiler options. This config file presently contains all of the filegroups found in the "core," "extensions," and "typings" folders of the Oppia codebase. As a result, when this configuration is utilized, the strict mode is deactivated for all files in the directories.

  Command used: **python -m scripts.typescript_checks**

- **typescript_checks.py**: The script typescript checks.py is used to compile and check typescript. It builds the files according to the settings specified in a typescript config file. If the flag "--strict checks" is used, tsconfig-strict.json is used to build the files. The tsconfig.json file is used instead if the flag is deleted.

- **pre_push_hook.py:** This hook also performs the same action as above. It runs both the commands and checks for errors before pushing to the repository.

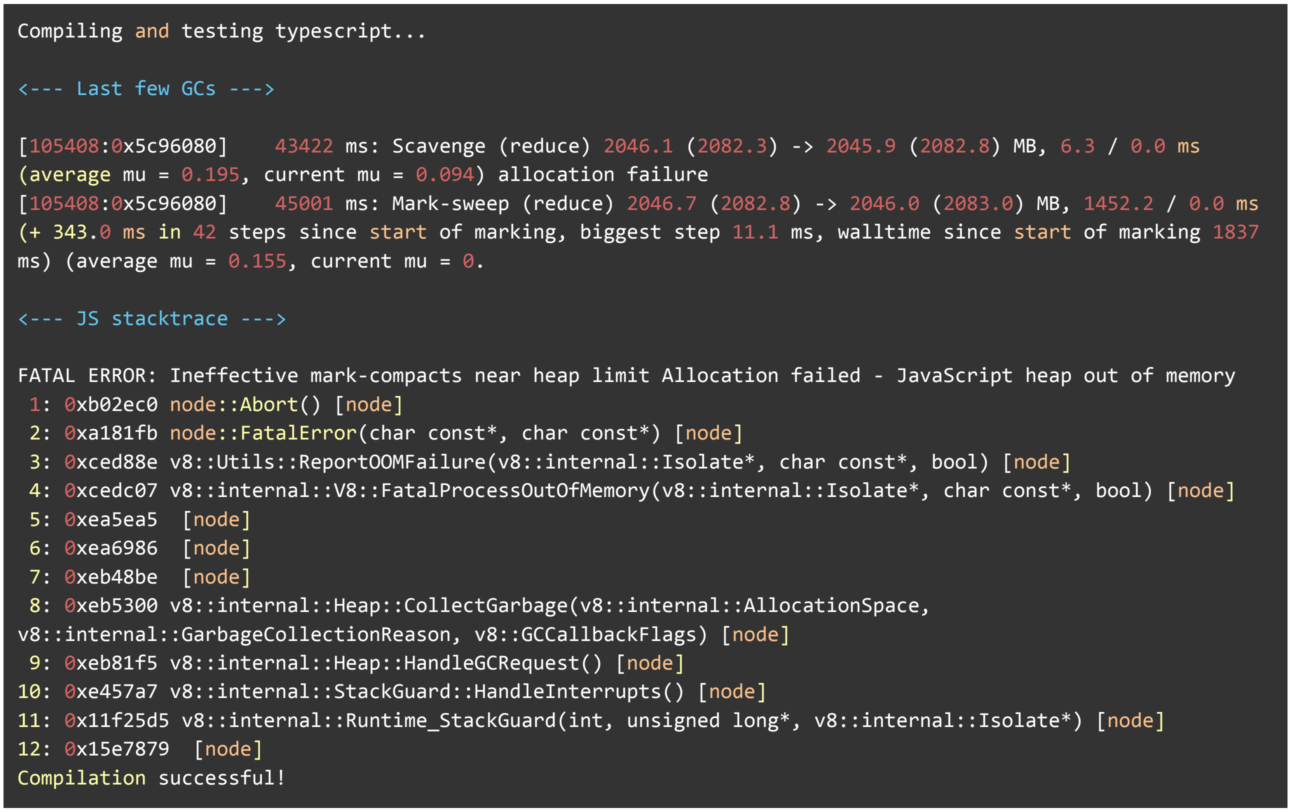
Solution Overview

The following is the project's main goal:

1. Change the **tsconfig-strict.json**, **tsconfig.json**, **typescript_checks.py**, **typescript_checks_test.py**. so that all newly-added files need to be strictly typed
2. Introduce typing for 560 files (this number also can include test files).
3. Customize the error logs by specifying the number of errors to be displayed.
4. Get rid of the type “unknown” and “as unknown as X” pattern from the files we are going to type and add a lint check to prevent usage of “unknown” in the future.

Goal 1: Enable typescript strict mode for all the newly added files.

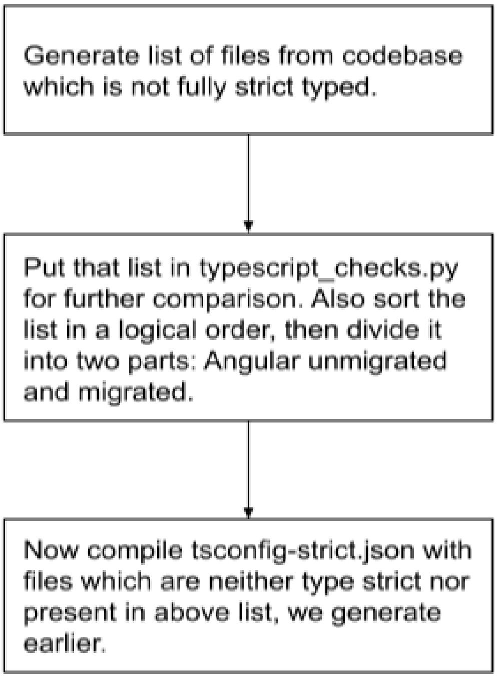
- My solution is similar to what we performed in the frontend testing using NOT_FULLY_COVERED_FILENAMES. Adding left files(unstrict typed files) to "exclude" isn't a smart idea because "include" would overwrite "exclude" and we can't compel new files to be strict. Also, if we just remove include list and use exclude only with left files, we are going to face runtime errors and out of memory issues.



```
Compiling and testing typescript...

<--- Last few GCs --->

[105408:0x5c96080]    43422 ms: Scavenge (reduce) 2046.1 (2082.3) -> 2045.9 (2082.8) MB, 6.3 / 0.0 ms
(average mu = 0.195, current mu = 0.094) allocation failure
[105408:0x5c96080]    45001 ms: Mark-sweep (reduce) 2046.7 (2082.8) -> 2046.0 (2083.0) MB, 1452.2 / 0.0 ms
(+ 343.0 ms in 42 steps since start of marking, biggest step 11.1 ms, walltime since start of marking 1837
ms) (average mu = 0.155, current mu = 0.

<--- JS stacktrace --->

FATAL ERROR: Ineffective mark-compacts near heap limit Allocation failed - JavaScript heap out of memory
 1: 0xb02ec0 node::Abort() [node]
 2: 0xa181fb node::FatalError(char const*, char const*) [node]
 3: 0xced88e v8::Utils::ReportOOMFailure(v8::internal::Isolate*, char const*, bool) [node]
 4: 0xcedc07 v8::internal::V8::FatalProcessOutOfMemory(v8::internal::Isolate*, char const*, bool) [node]
 5: 0xea5ea5  [node]
 6: 0xea6986  [node]
 7: 0xeb48be  [node]
 8: 0xeb5300 v8::internal::Heap::CollectGarbage(v8::internal::AllocationSpace,
v8::internal::GarbageCollectionReason, v8::GCCallbackFlags) [node]
 9: 0xeb81f5 v8::internal::Heap::HandleGCRequest() [node]
10: 0xe457a7 v8::internal::StackGuard::HandleInterrupts() [node]
11: 0x11f25d5 v8::internal::Runtime_StackGuard(int, unsigned long*, v8::internal::Isolate*) [node]
12: 0x15e7879  [node]
Compilation successful!
```

- If a new file is added, we'll check if it's in the NOT_FULLY_COVERED_FILENAMES, and if it isn't, we'll display a strict typescript error.
- So, anytime we execute **python -m scripts.typescripts_checks --strict_checks**, we intend to show a strict typescript error for all files that are not contained in the NOT_FULLY_COVERED_FILENAMES.

Steps: How are we going to enforce strictness for newly inserted files



Generate list of files from codebase which is not fully strict typed.

Put that list in typescript_checks.py for further comparison. Also sort the list in a logical order, then divide it into two parts: Angular unmigrated and migrated.

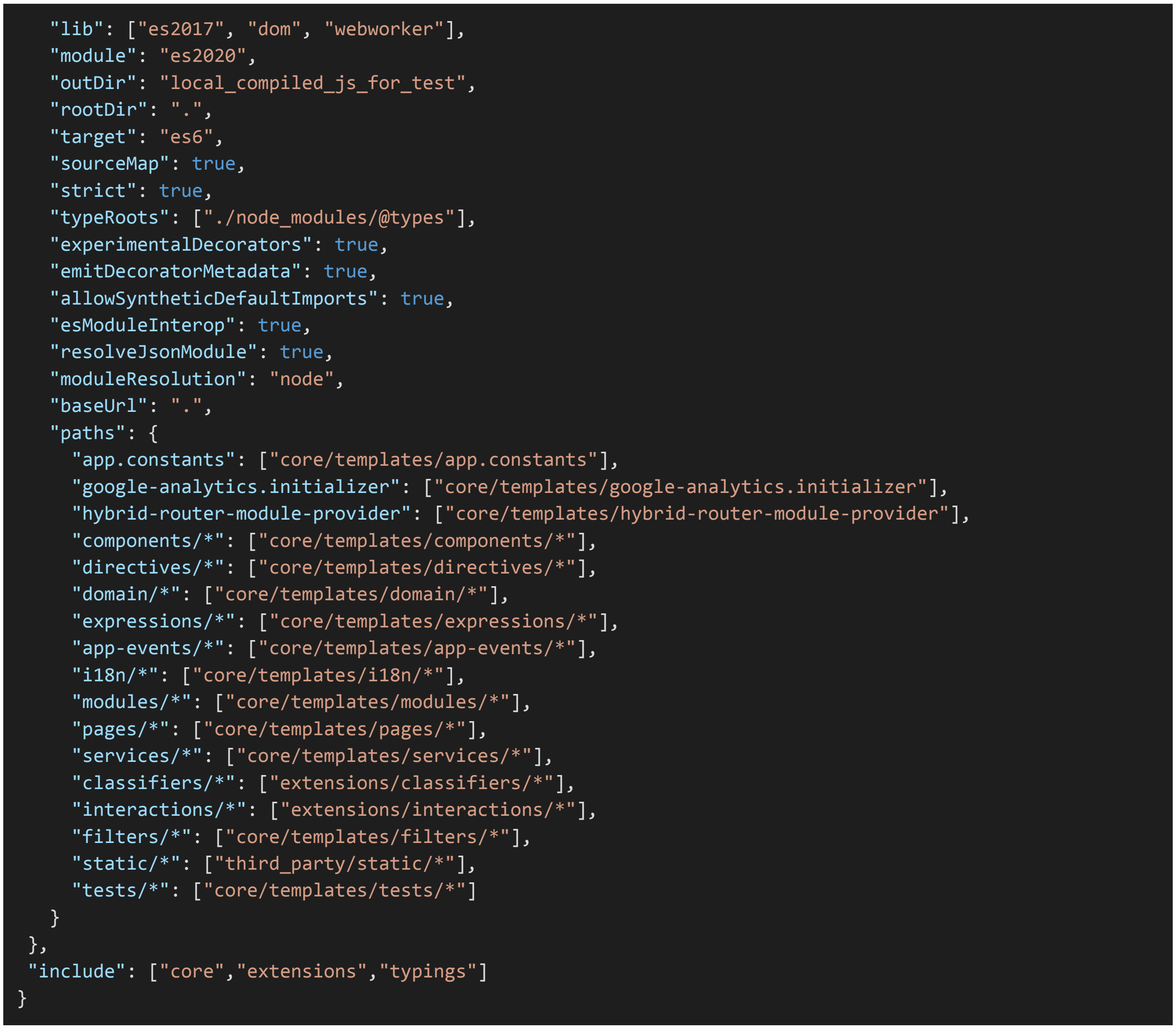
Now compile tsconfig-strict.json with files which are neither type strict nor present in above list, we generate earlier.

This is how we plan to compile tsconfig-strict.json with files that aren't typescript or in the left files list. As a result, those files are fresh/newly additions.

Step 1: Generate list of files from codebase which are not strictly typed.

Remove “files”, “exclude” property and make include as "include": ["core",   "extensions", "typings"] in tsconfig-strict.json. Now tsconfig-strict.json will look like.

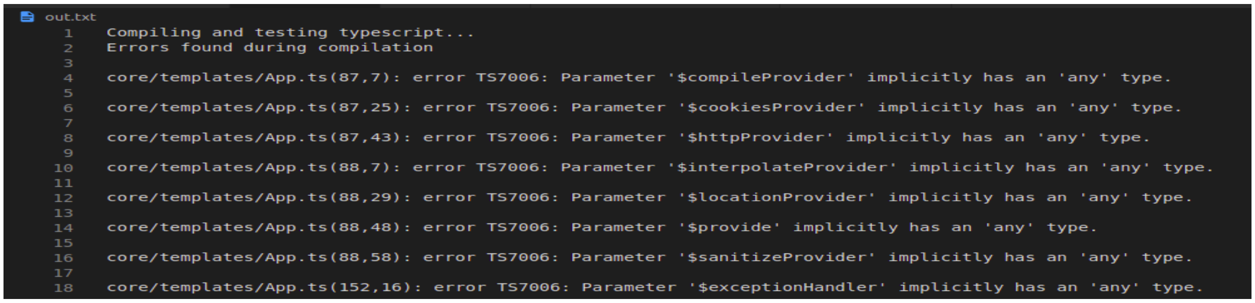
```
{
 "compilerOptions": {
   "allowJs": true,
   "skipLibCheck": true,
   "downlevelIteration": true,
```



```
    "lib": ["es2017", "dom", "webworker"],
    "module": "es2020",
    "outDir": "local_compiled_js_for_test",
    "rootDir": ".",
    "target": "es6",
    "sourceMap": true,
    "strict": true,
    "typeRoots": ["./node_modules/@types"],
    "experimentalDecorators": true,
    "emitDecoratorMetadata": true,
    "allowSyntheticDefaultImports": true,
    "esModuleInterop": true,
    "resolveJsonModule": true,
    "moduleResolution": "node",
    "baseUrl": ".",
    "paths": {
      "app.constants": ["core/templates/app.constants"],
      "google-analytics.initializer": ["core/templates/google-analytics.initializer"],
      "hybrid-router-module-provider": ["core/templates/hybrid-router-module-provider"],
      "components/*": ["core/templates/components/*"],
      "directives/*": ["core/templates/directives/*"],
      "domain/*": ["core/templates/domain/*"],
      "expressions/*": ["core/templates/expressions/*"],
      "app-events/*": ["core/templates/app-events/*"],
      "i18n/*": ["core/templates/i18n/*"],
      "modules/*": ["core/templates/modules/*"],
      "pages/*": ["core/templates/pages/*"],
      "services/*": ["core/templates/services/*"],
      "classifiers/*": ["extensions/classifiers/*"],
      "interactions/*": ["extensions/interactions/*"],
      "filters/*": ["core/templates/filters/*"],
      "static/*": ["third_party/static/*"],
      "tests/*": ["core/templates/tests/*"]
    }
  },
  "include": ["core","extensions","typings"]
}
```

Log the error in a text file named out.txt for better understanding. Run command

python -m scripts.typescript_checks --strict_checks > out.txt

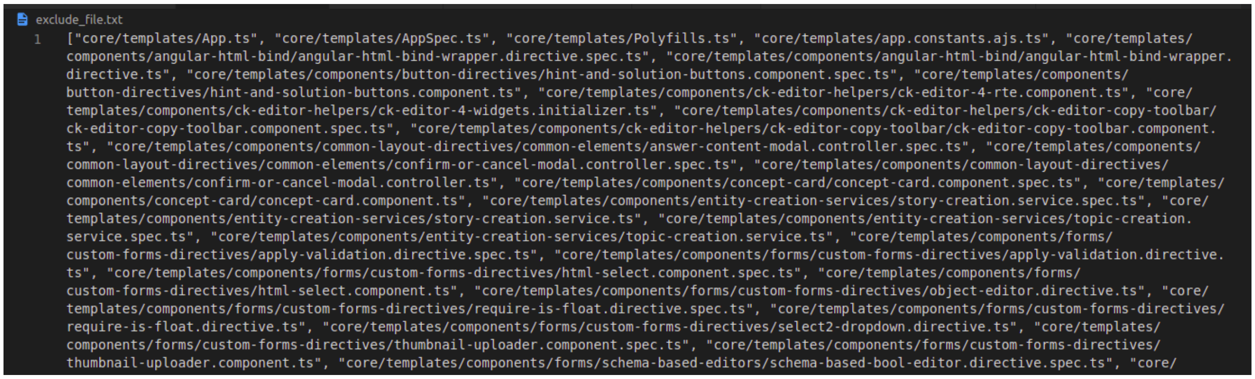


```
Compiling and testing typescript...
Errors found during compilation

core/templates/App.ts(87,7): error TS7006: Parameter '$compileProvider' implicitly has an 'any' type.

core/templates/App.ts(87,25): error TS7006: Parameter '$cookiesProvider' implicitly has an 'any' type.

core/templates/App.ts(87,43): error TS7006: Parameter '$httpProvider' implicitly has an 'any' type.

core/templates/App.ts(88,7): error TS7006: Parameter '$interpolateProvider' implicitly has an 'any' type.

core/templates/App.ts(88,29): error TS7006: Parameter '$locationProvider' implicitly has an 'any' type.

core/templates/App.ts(88,48): error TS7006: Parameter '$provide' implicitly has an 'any' type.

core/templates/App.ts(88,58): error TS7006: Parameter '$sanitizeProvider' implicitly has an 'any' type.

core/templates/App.ts(152,16): error TS7006: Parameter '$exceptionHandler' implicitly has an 'any' type.
```

Use python script to clean output. Make a new file named **out.py** at the root level..

```
import json

with open("out.txt") as f:
   errors = f.readlines()

errors = [x.strip() for x in errors]
# Remove the empty lines and error explanation lines.
prefixes = ("core", "extension", "typings")
errors = [x for x in errors if x.startswith(prefixes)]
# Remove error explanation lines.
errors = [x.split("(",1)[0] for x in errors]
# Remove the dublin core prefixes.
errors = list(dict.fromkeys(errors))
errors = sorted(errors)

print(json.dumps(errors))
```

Run command **python out.py > exclude_file.txt** to see all the files which are not strictly typed in typescript currently.



```
["core/templates/App.ts", "core/templates/AppSpec.ts", "core/templates/Polyfills.ts", "core/templates/app.constants.ajs.ts", "core/templates/
components/angular-html-bind/angular-html-bind-wrapper.directive.spec.ts", "core/templates/components/angular-html-bind/angular-html-bind-wrapper.
directive.ts", "core/templates/components/button-directives/hint-and-solution-buttons.component.spec.ts", "core/templates/components/
button-directives/hint-and-solution-buttons.component.ts", "core/templates/components/ck-editor-helpers/ck-editor-4-rte.component.ts", "core/
templates/components/ck-editor-helpers/ck-editor-4-widgets.initializer.ts", "core/templates/components/ck-editor-helpers/ck-editor-copy-toolbar/
ck-editor-copy-toolbar.component.spec.ts", "core/templates/components/ck-editor-helpers/ck-editor-copy-toolbar/ck-editor-copy-toolbar.component.
ts", "core/templates/components/common-layout-directives/common-elements/answer-content-modal.controller.spec.ts", "core/templates/components/
common-layout-directives/common-elements/confirm-or-cancel-modal.controller.spec.ts", "core/templates/components/common-layout-directives/
common-elements/confirm-or-cancel-modal.controller.ts", "core/templates/components/concept-card/concept-card.component.spec.ts", "core/templates/
components/concept-card/concept-card.component.ts", "core/templates/components/entity-creation-services/story-creation.service.spec.ts", "core/
templates/components/entity-creation-services/story-creation.service.ts", "core/templates/components/entity-creation-services/topic-creation.
service.spec.ts", "core/templates/components/entity-creation-services/topic-creation.service.ts", "core/templates/components/forms/
custom-forms-directives/apply-validation.directive.spec.ts", "core/templates/components/forms/custom-forms-directives/apply-validation.directive.
ts", "core/templates/components/forms/custom-forms-directives/html-select.component.spec.ts", "core/templates/components/forms/
custom-forms-directives/html-select.component.ts", "core/templates/components/forms/custom-forms-directives/object-editor.directive.ts", "core/
templates/components/forms/custom-forms-directives/require-is-float.directive.spec.ts", "core/templates/components/forms/custom-forms-directives/
require-is-float.directive.ts", "core/templates/components/forms/custom-forms-directives/select2-dropdown.directive.ts", "core/templates/
components/forms/custom-forms-directives/thumbnail-uploader.component.spec.ts", "core/templates/components/forms/custom-forms-directives/
thumbnail-uploader.component.ts", "core/templates/components/forms/schema-based-editors/schema-based-bool-editor.directive.spec.ts", "core/
```

Step 2: Put that list in **typescript_checks.py** for further comparison. Example:

```
# Contains the name of all files that are not strictly typed.
# This list must be kept up-to-date; the changes (only remove) should be done
# manually.
# Please keep the list in alphabetical order.
# NOTE TO DEVELOPERS: do not add any new files to this list without asking
# @vojtechjelinek first.
NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH = [
   "core/templates/Polyfills.ts",
   "core/templates/app.constants.ajs.ts",
   "core/templates/App.ts",
   "core/templates/AppSpec.ts",
]
```

Actual number of unstrict typed files that are in Angular are **453**

Actual number of unstrict typed files that are in AngularJS are **334**

Total number of files this list contains is **787**

Github Gist of the above list.

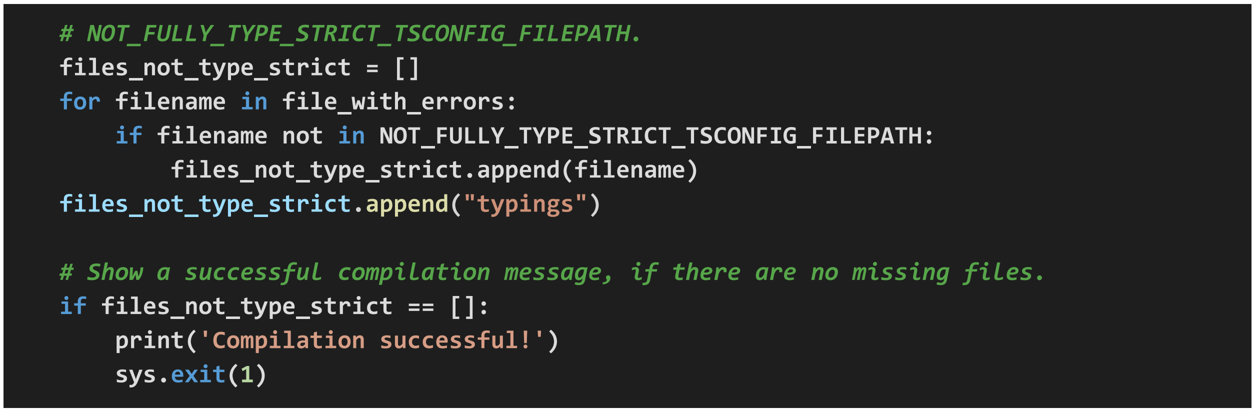
Step 3: Now compile tsconfig-strict.json with files which are neither type strict nor present in the above list, we generated earlier.

We can find out new unstrict typed files by comparing errors created by typescript_checks.py for **"core," "extensions," and "typings"** (which always contain newly added files) with files that are not strictly typed currently, which we generate in **step 2**. In typescript_checks.py, use the same python script of **step 1** we generate in out.py to get all files (which always contain newly added files).

```
os.environ['PATH'] = '%s/bin:' % common.NODE_PATH + os.environ['PATH']
validate_compiled_js_dir()

if os.path.exists(COMPILED_JS_DIR):
    shutil.rmtree(COMPILED_JS_DIR)

print('Compiling and testing typescript...')
cmd = ['./node_modules/typescript/bin/tsc', '--project', config_path]
process = subprocess.Popen(cmd, stdout=subprocess.PIPE, encoding='utf-8')

if os.path.exists(COMPILED_JS_DIR):
    shutil.rmtree(COMPILED_JS_DIR)

# Generate a list of files that are not strict typescript.
error_messages = list(iter(process.stdout.readline, ''))

errors = [x.strip() for x in error_messages]
# Remove the empty lines and error explanation lines.
prefixes = ("core", "extension", "typings")
errors = [x for x in errors if x.startswith(prefixes)]
# Remove error explanation lines.
errors = [x.split("(",1)[0] for x in errors]
# Remove the dublin core prefixes.
errors = list(dict.fromkeys(errors))
file_with_errors = sorted(errors)
```

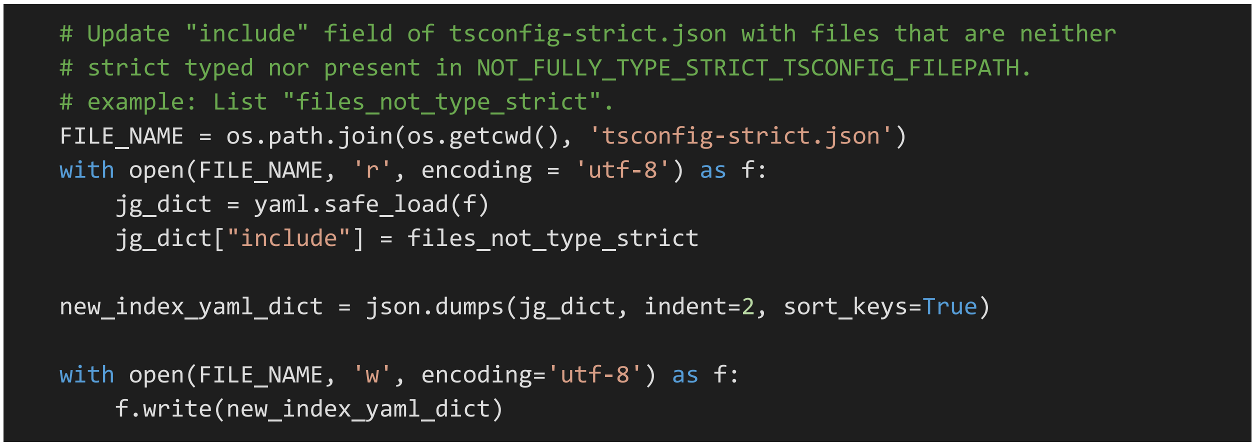
Store files path names which are newly added or files which are neither type strict nor present in **NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH.** Also save the "typings" folder for global imports such as "angular"; if we don't, we'll receive issues like "Using angular without declaration". Show a successful compilation message, if there are no missing files.

```
# List of missing files that are neither strict typed nor present in
```



```
# NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH.
files_not_type_strict = []
for filename in file_with_errors:
    if filename not in NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH:
        files_not_type_strict.append(filename)
files_not_type_strict.append("typings")

# Show a successful compilation message, if there are no missing files.
if files_not_type_strict == []:
    print('Compilation successful!')
    sys.exit(1)
```

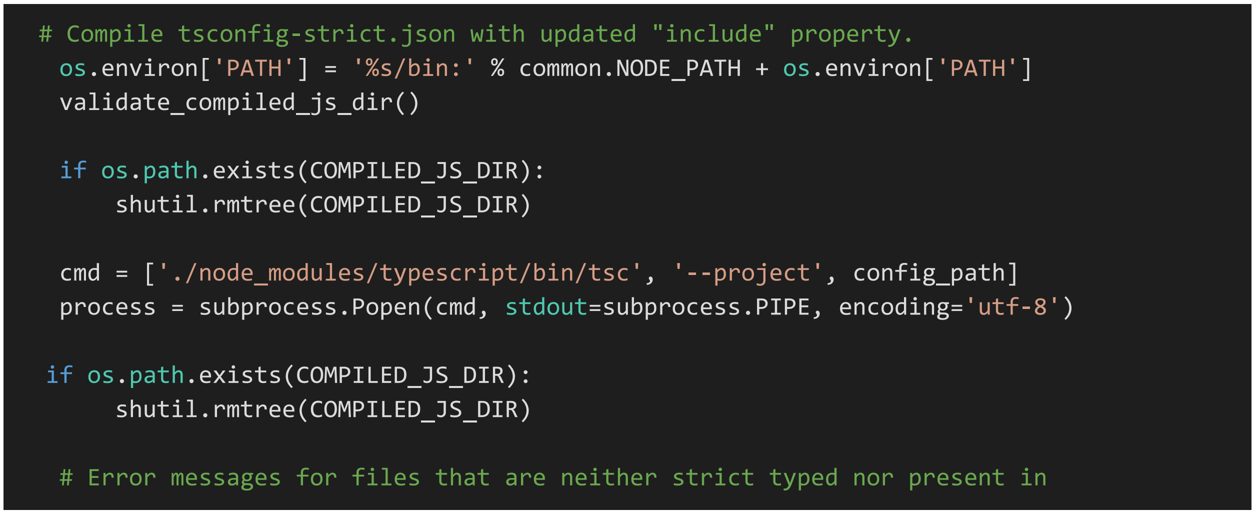
Till now we have generated error logs for **"include": ["core", "extensions", "typings"]**, but the main goal is to generate error logs for **files_not_type_strict.** So update **“include”** key of **tsconfig-strict.json** with file paths present in files_not_type_strict.



```
# Update "include" field of tsconfig-strict.json with files that are neither
# strict typed nor present in NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH.
# example: List "files_not_type_strict".
FILE_NAME = os.path.join(os.getcwd(), 'tsconfig-strict.json')
with open(FILE_NAME, 'r', encoding = 'utf-8') as f:
    jg_dict = yaml.safe_load(f)
    jg_dict["include"] = files_not_type_strict

new_index_yaml_dict = json.dumps(jg_dict, indent=2, sort_keys=True)

with open(FILE_NAME, 'w', encoding='utf-8') as f:
    f.write(new_index_yaml_dict)
```

*Note: Don’t forget to import “yaml”. Now **tsconfig-strict.json** is updated*

Run the revised tsconfig-strict.json again to create error logs.



```
# Compile tsconfig-strict.json with updated "include" property.
 os.environ['PATH'] = '%s/bin:' % common.NODE_PATH + os.environ['PATH']
 validate_compiled_js_dir()

 if os.path.exists(COMPILED_JS_DIR):
     shutil.rmtree(COMPILED_JS_DIR)

 cmd = ['./node_modules/typescript/bin/tsc', '--project', config_path]
 process = subprocess.Popen(cmd, stdout=subprocess.PIPE, encoding='utf-8')

if os.path.exists(COMPILED_JS_DIR):
     shutil.rmtree(COMPILED_JS_DIR)

 # Error messages for files that are neither strict typed nor present in
 # NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH.
 error_messages = list(iter(process.stdout.readline, ''))
```

Update “include” before printing error logs in the terminal. Because we need include as **"include": ["core", "extensions", "typings"],**for the next run.

```
# Update tsconfig-strict.json and set to its initial "include" state
# example "include": ["core", "extensions", "typings"].
with open(FILE_NAME, 'r', encoding = 'utf-8') as f:
    jg_dict = yaml.safe_load(f)
    jg_dict["include"] = ["core", "extensions", "typings"]

new_index_yaml_dict = json.dumps(jg_dict, indent=2, sort_keys=True)

with open(FILE_NAME, 'w', encoding='utf-8') as f:
    f.write(new_index_yaml_dict)

if error_messages:
    print('Errors found during compilation\n')
    print('\n'.join(error_messages))
    sys.exit(1)
else:
    print('Compilation successful!')
```

Remove NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH and above all changes we did in typescript_checks.py once we finished with strict typing in the whole codebase. *(not related with this proposal).*

Implementation Approach

Now we can choose file from “NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH” and do strict typings as we did earlier. Briefly explain in GOAL 2.

My method has a number of advantages over the usual method of including a file path in the include field of tsconfig-strict.json.

Advantages:

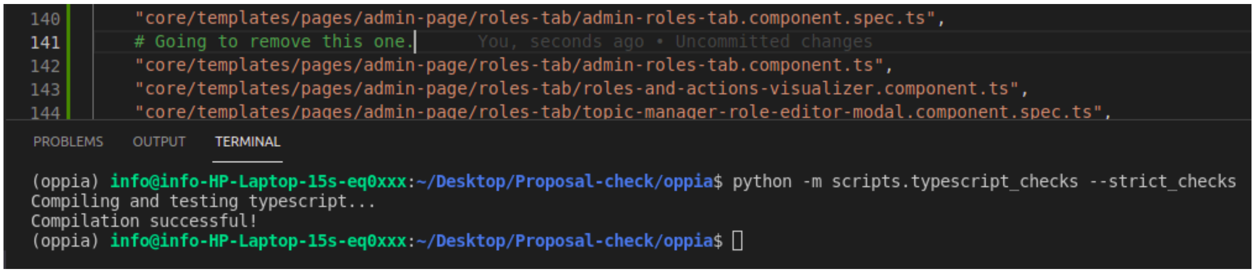
- Now files are not dependent on ts config files anymore.
- Automatically check strict typing in newly added files.
- Achieve strict typing in “core”, “extensions”, “typings” from starting.
- Follow the transient property of strictness.
- If a developer is working on a large PR, he or she may double-check the strict typescript of any of his finished files while working on PR locally. (Breaking down large projects into smaller working sections).
- New developers no longer have to worry about adding, deleting, or changing files in the ts config file. Work independently without any hindrance.

- Get proper list of unstrict typed files.
- Most Important: If you are working on newly added files, It will help you to detect errors before pushing code to repo of your each and every file. (Breaking down large projects into smaller working sections).

Goal 2: Introduce typing for 560 files (this number also can include test files).

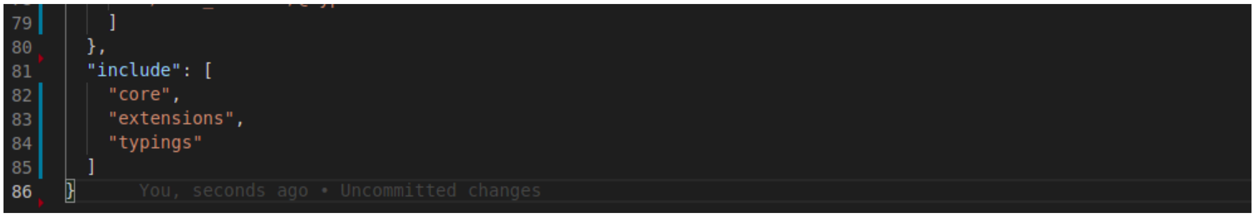
We now need to delete the file path name from the list of all files that are now showing strict typescript problems, as per the prior approach. example In typescirpt checks.py, the NOT_FULLY_TYPE_STRICT_TSCONFIG_PATH.

Before removal of any file.



```
    "core/templates/pages/admin-page/roles-tab/admin-roles-tab.component.spec.ts",
    # Going to remove this one.       You, seconds ago • Uncommitted changes
    "core/templates/pages/admin-page/roles-tab/admin-roles-tab.component.ts",
    "core/templates/pages/admin-page/roles-tab/roles-and-actions-visualizer.component.ts",
    "core/templates/pages/admin-page/roles-tab/topic-manager-role-editor-modal.component.spec.ts",
```

```
(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ python -m scripts.typescript_checks --strict_checks
Compiling and testing typescript...
Compilation successful!
(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ 
```

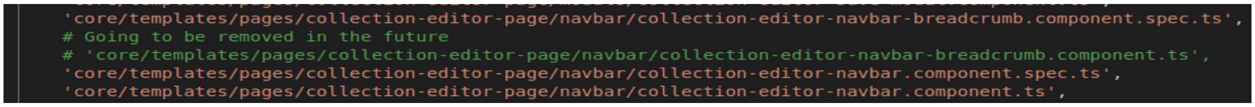


```
    ]
  },
  "include": [
    "core",
    "extensions",
    "typings"
  ]
}       You, seconds ago • Uncommitted changes
```

As you can see, there were no errors in the entire codebase. Because we haven't removed any of the file path names from NOT_FULLY_TYPE_STRICT_TSCONFIG_PATH.

After removal of file

"core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts": File remove from NOT_FULLY_TYPE_STRICT_TSCONFIG_PATH.

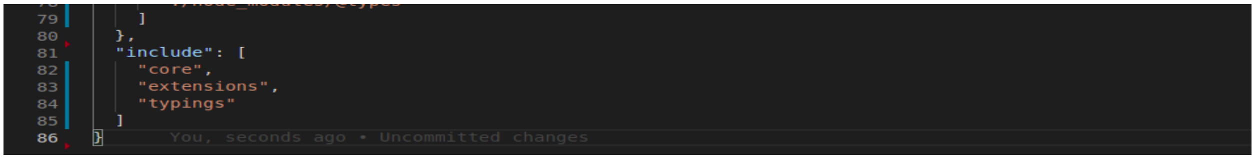


```
    'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.spec.ts',
    # Going to be removed in the future
    # 'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts',
    'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar.component.spec.ts',
    'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar.component.ts',
```

Errors:

```
(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ python -m scripts.typescript_checks
--strict_checks
Compiling and testing typescript...

core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts(46,3): error
```

```
TS2564: Property 'activeTabName' has no initializer and is not definitely assigned in the constructor.

core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts(47,3): error
TS2564: Property 'collection' has no initializer and is not definitely assigned in the constructor.

core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts(56,12): error
TS7053: Element implicitly has an 'any' type because expression of type 'string' can't be used to index type '{ edit:
string; settings: string; stats: string; history: string; }'.

  No index signature with a parameter of type 'string' was found on type '{ edit: string; settings: string; stats:
string; history: string; }'.

core/templates/pages/collection-editor-page/services/collection-editor-routing.service.ts(32,11): error TS2564:
Property '_activeTabName' has no initializer and is not definitely assigned in the constructor.

core/templates/pages/collection-editor-page/services/collection-editor-state.service.ts(184,7): error TS2345: Argument
of type 'string | null' is not assignable to parameter of type 'string'.

  Type 'null' is not assignable to type 'string'.

7 Errors found during compilation.
```



```
        ./node_modules/@types
      ]
    },
    "include": [
      "core",
      "extensions",
      "typings"
    ]
}       You, seconds ago • Uncommitted changes
```

As we can see, we are encountering errors regarding our file chosen and its dependencies. Basically the strictness is transient, so if A → B → C. And we say that A needs to be strictly typed both B and C need to be strictly typed, that's why we need to solve errors regarding dependencies as well.

Important Note:
What will happen if someone adds a new file that contains a lot of imports and due to that it produces a lot of TS errors?

For this we can simply add file paths which are showing errors in our list **NOT_FULLY_TYPE_STRICT_TSCONFIG_PATH** present in typescirpt_checks.py. And later solve its dependencies first then the file which contain lot of imports.

(some of file paths may be present already).

```
# Contains the name of all files that are not strictly typed.
# This list must be kept up-to-date; the changes (only remove) should be done
# manually.
# Please keep the list in alphabetical order and divided into two parts:
# 1. The files that are in Angular.
# 2. The files that are in AngularJS.
# NOTE TO DEVELOPERS: do not add any new files to this list without asking
# @vojtechjelinek first.
NOT_FULLY_TYPE_STRICT_TSCONFIG_FILEPATH = [
    # Files that are in Angular.
```

Approach for Resolving Errors:

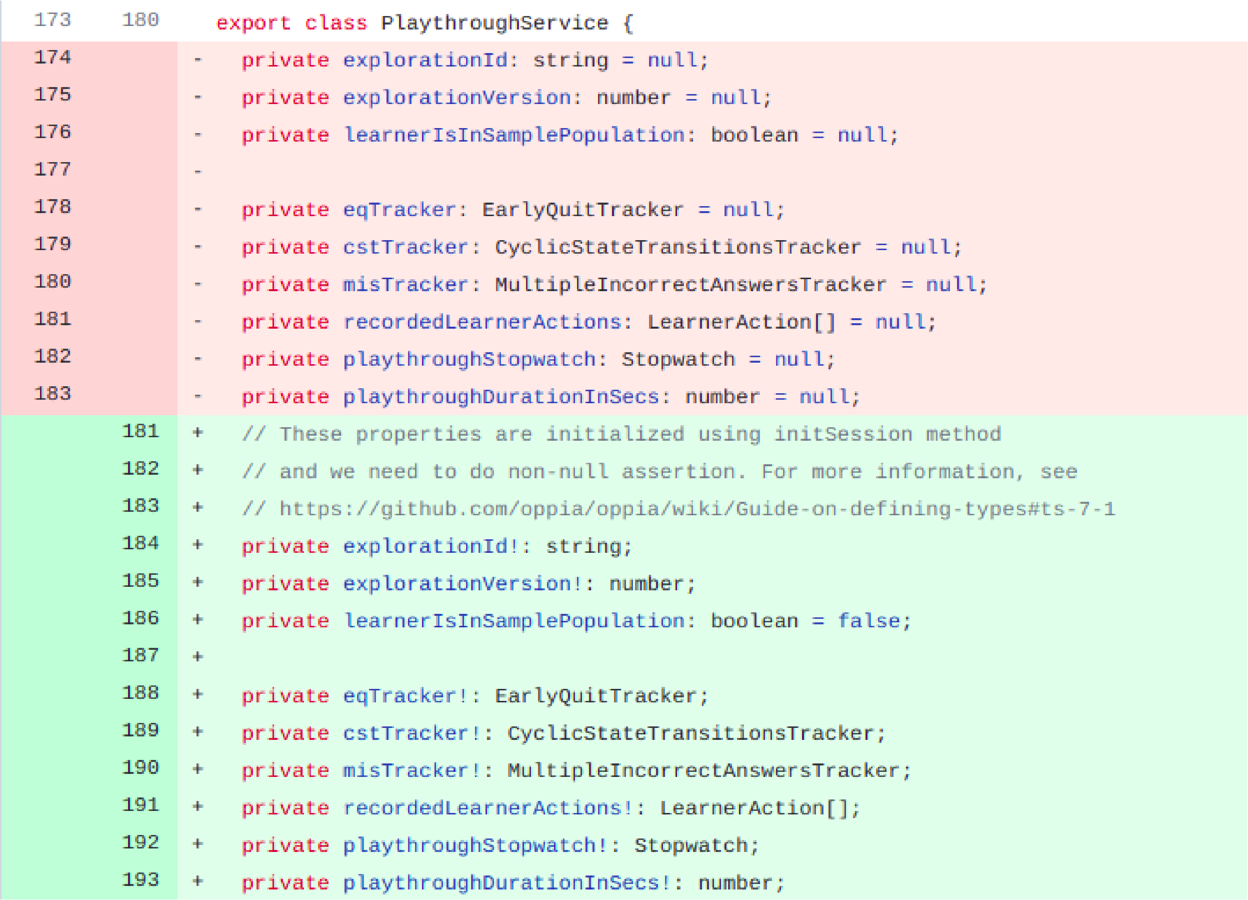
I analyzed out.txt which i generated earlier for loging all errors using:
python -m scripts.typescript_checks --strict_checks > out.txt

Most of the encountered errors are present in this guide.

Other cases I encountered are mentioned below:

Type 1: **Null assignments to defined variables.**

The process of typing isn't just a mechanical process, it needs some proper analysis of what's going on. So, null assignments can be removed from the places where not required. Since using null means that there's a leak somewhere in the logic , therefore it would be worth checking whether that leak can be patched.



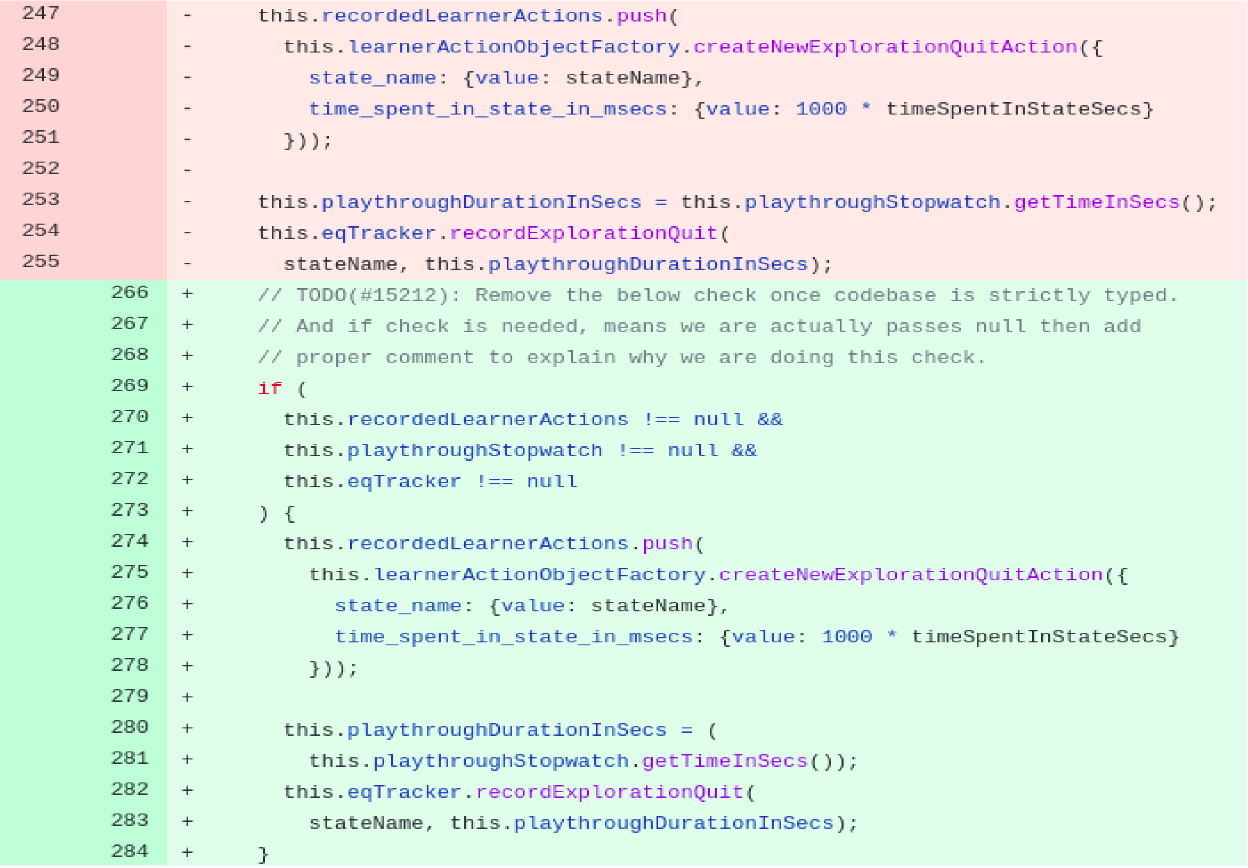
```
  export class PlaythroughService {
-   private explorationId: string = null;
-   private explorationVersion: number = null;
-   private learnerIsInSamplePopulation: boolean = null;
-
-   private eqTracker: EarlyQuitTracker = null;
-   private cstTracker: CyclicStateTransitionsTracker = null;
-   private misTracker: MultipleIncorrectAnswersTracker = null;
-   private recordedLearnerActions: LearnerAction[] = null;
-   private playthroughStopwatch: Stopwatch = null;
-   private playthroughDurationInSecs: number = null;
+   // These properties are initialized using initSession method
+   // and we need to do non-null assertion. For more information, see
+   // https://github.com/oppia/oppia/wiki/Guide-on-defining-types#ts-7-1
+   private explorationId!: string;
+   private explorationVersion!: number;
+   private learnerIsInSamplePopulation: boolean = false;
+
+   private eqTracker!: EarlyQuitTracker;
+   private cstTracker!: CyclicStateTransitionsTracker;
+   private misTracker!: MultipleIncorrectAnswersTracker;
+   private recordedLearnerActions!: LearnerAction[];
+   private playthroughStopwatch!: Stopwatch;
+   private playthroughDurationInSecs!: number;
```

Note: Comment Regarding non-null assertion varies according to situations. Like we may also initialize properties using Angular lifecycle hooks, private methods or with some other specific function.

You can use a non-null (!) assertion operator which asserts that the object is non-null and non-undefined. It solves the strictPropertyInitialization rule that enforces properties must be assigned either in the constructor or with a proper initializer.

But we need to use non-null (!) safely as it may be the case when these values would not be filled either in constructor or ngOnInit or it is not initialized properly before actual use of it.

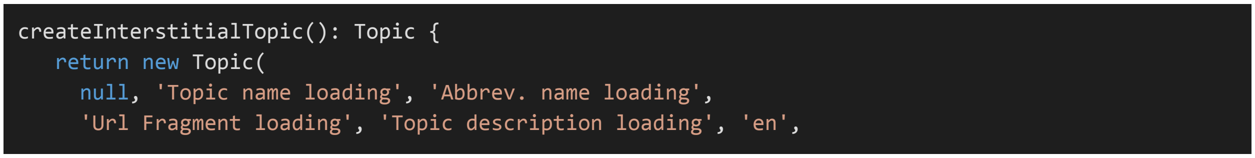
So you can add the non-null checks and then also add the if checks in the code. The reason for that is we cannot be sure in some non-strict file that if null is passed as an argument or if a value is used before it is initialized.In both of these cases we can do lite refactoring otherwise it can make code prone to runtime errors.



```
-     this.recordedLearnerActions.push(
-       this.learnerActionObjectFactory.createNewExplorationQuitAction({
-         state_name: {value: stateName},
-         time_spent_in_state_in_msecs: {value: 1000 * timeSpentInStateSecs}
-       }));
-
-     this.playthroughDurationInSecs = this.playthroughStopwatch.getTimeInSecs();
-     this.eqTracker.recordExplorationQuit(
-       stateName, this.playthroughDurationInSecs);
+     // TODO(#15212): Remove the below check once codebase is strictly typed.
+     // And if check is needed, means we are actually passes null then add
+     // proper comment to explain why we are doing this check.
+     if (
+       this.recordedLearnerActions !== null &&
+       this.playthroughStopwatch !== null &&
+       this.eqTracker !== null
+     ) {
+       this.recordedLearnerActions.push(
+         this.learnerActionObjectFactory.createNewExplorationQuitAction({
+           state_name: {value: stateName},
+           time_spent_in_state_in_msecs: {value: 1000 * timeSpentInStateSecs}
+         }));
+
+       this.playthroughDurationInSecs = (
+         this.playthroughStopwatch.getTimeInSecs());
+       this.eqTracker.recordExplorationQuit(
+         stateName, this.playthroughDurationInSecs);
+     }
```

We should also add TODO(#15212) comments before the if check that will say that these if checks can be removed after we have strictly typed the whole codebase.

Also there are numerous places in the codebase where null is assigned as a type to a non-null variable with no origin of null, implying that these properties will never be null because the backend will always provide a non-null value. It basically signifies that we need to fix the value leak we're getting from the backend.

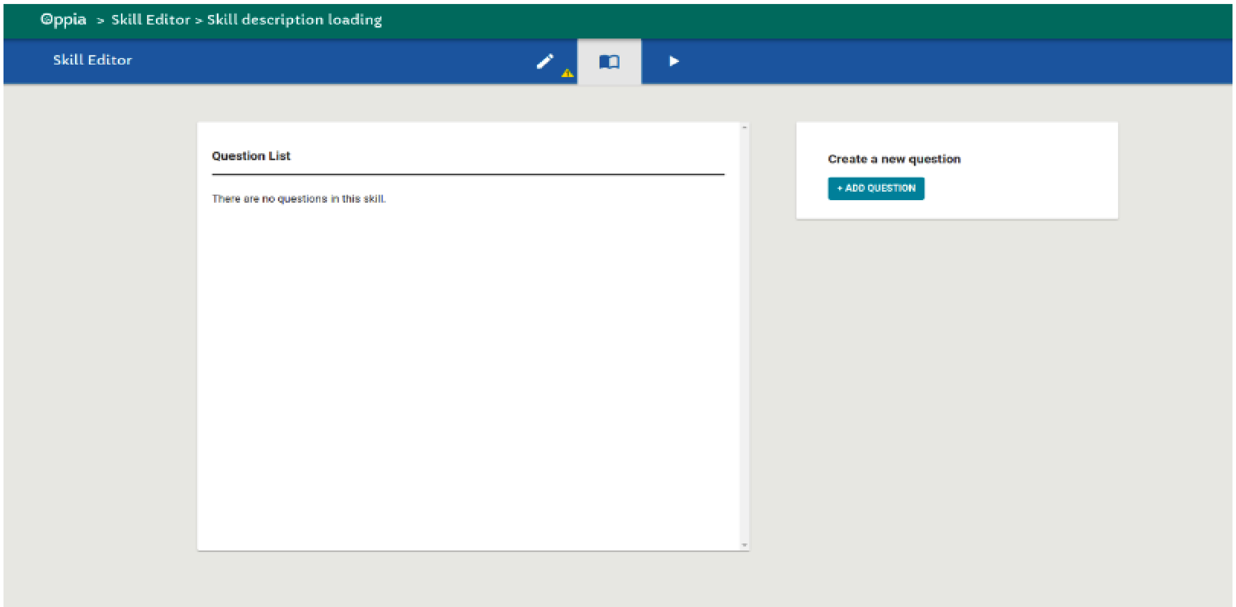


```
createInterstitialTopic(): Topic {
   return new Topic(
     null, 'Topic name loading', 'Abbrev. name loading',
     'Url Fragment loading', 'Topic description loading', 'en',
    [], [], [], 1, 1, [], null, '', {},
    this.storyReferenceObjectFactory, false, '', ''
  );
}
```

Potential Solution

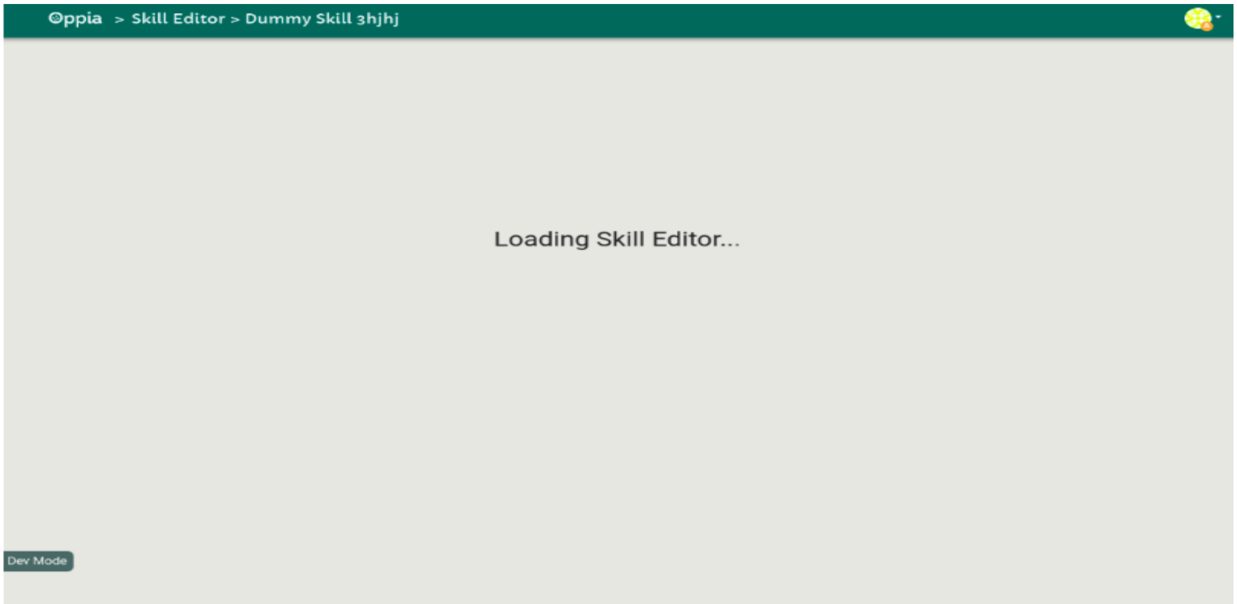
Remove the use of interstitial objects by refactoring code to make use of the LoaderService which displays a loading message until the object has been fetched from the backend.

Currently, placeholders (interstitial objects) are created until the loading completes



There are no questions in this skill.

We would like a loader to show up instead of those placeholders.



Loading Skill Editor...

We also have issue #13637 to deal with all these interstitial objects.
Reference PR: #13638

So, we need to understand the full flow of null, before actually adding or removing into types. There is no proper pattern of using null, which defines all cases. So there is no procedure that will be used to decide which case a particular instance belongs to. But for make ease in review and to understand flow of null we can do following things:

For **adding** null as type:

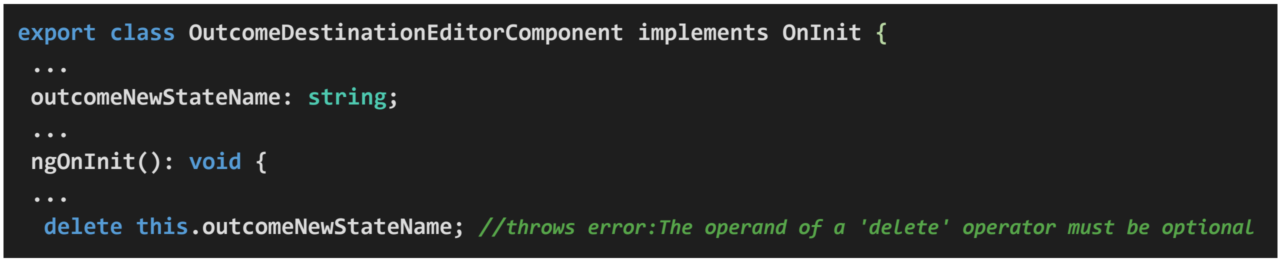
1. Add code comment, why we are using null here and what its significance is, above null type.
2. And it is also important not to carry null flow forward, so stop this by throwing a new error where it is not required.
3. Examples: XYZ will be null if the user has not input any value. (one of very common case uses in codebase).
4. Examples: Null comes from the backend.

For **removing** of null as type:

1. Will make a comment inside PR as a self-reviewer before actual review, why i remove this. This will save time and bring more clarity to reviewers.
2. **Note:** There is one common case where I see null used when it shouldn't be, which is as a temporary value when the page hasn't fully loaded yet and the init function hasn't run yet. In those cases we are going to use non-null assertion "!" and add a comment explaining the same.

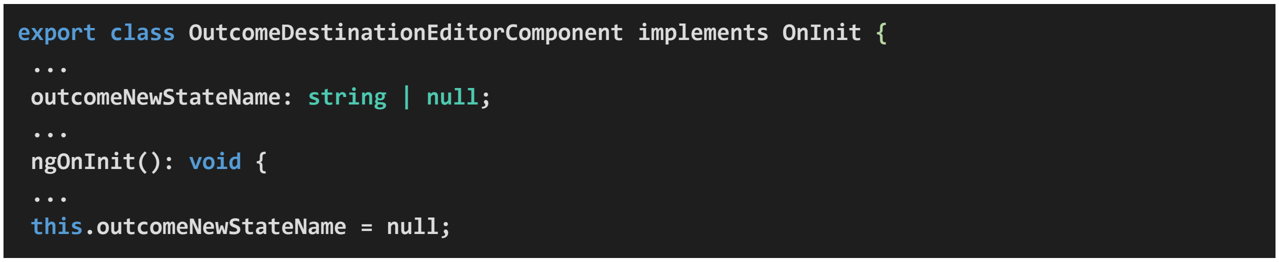
Type 2: The operand of a 'delete' operator must be optional.

This error occurs when using the delete operator in strictNullChecks. Because the property is no longer implemented after deletion, set its type to undefined. As a result, the property must be undefined; otherwise, the delete operator leads to an error.



```
export class OutcomeDestinationEditorComponent implements OnInit {
 ...
 outcomeNewStateName: string;
 ...
 ngOnInit(): void {
 ...
  delete this.outcomeNewStateName; //throws error:The operand of a 'delete' operator must be optional
```

One of the potential fix to this error is to assign `undefined` but generally we try to avoid unnecessary use of `undefined`. Because usage of `undefined` does not represent or give proper information about the actual situation.
As a result, in this case, we choose to use `null` rather than `undefined`. Because the delete operator makes a property unusable, `null` is a better representation of emptiness, unusable or property is now no longer implemented.



```
export class OutcomeDestinationEditorComponent implements OnInit {
 ...
 outcomeNewStateName: string | null;
 ...
 ngOnInit(): void {
 ...
 this.outcomeNewStateName = null;
```

Order of picking files

There are 453 files in Angular that are not strict typed. And we'll start with them because making AngularJs files strict is out of our scope and can only be done once they've been converted to Angular.

Out of 453 files, 300 have no dependencies, 147 have dependencies, and 6 can only be strictly typed when the entire code base has been strictly typed.

The order in which we will select files is as follows: first, select files that have no dependencies. And give more preference to services.

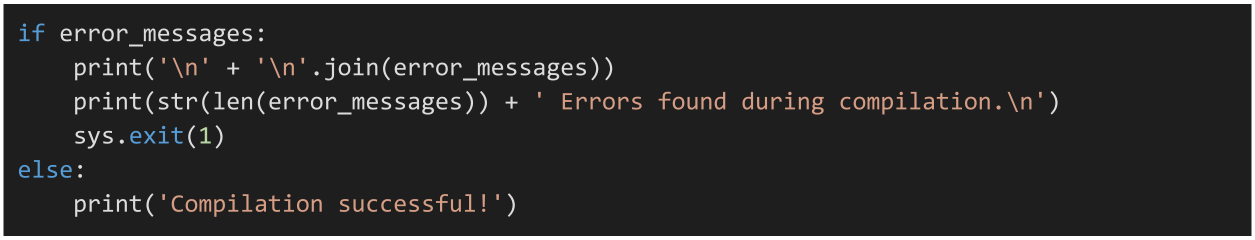
Reason:

- If we choose a file with a lot of dependencies, we'll have to fix a strict typescript issue and conduct some refactoring of the dependencies file first, before we can debug the refactored part. We virtually always strive to stick to our fundamental concept of breaking down big projects into smaller working sections.
- As almost every file has error **Null assignments to defined variables.** So in a file with no dependencies we can easily understand the flow of null, means using NULL makes sense or it is just a leak in information**.**
- Due to the large number of files with no dependencies (almost 300), I will be able to cover a large portion of the project from the start.

Goal 3: Number of errors in addition to the error logs in the terminal.

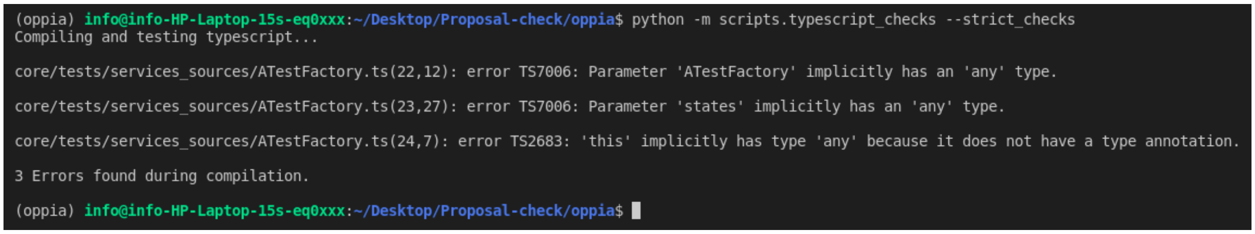
I want as much information as possible as a developer. Let's say we're dealing with a large number of faults and wish to focus on a specific one.

Change the print statement of error logs.



```
if error_messages:
    print('\n' + '\n'.join(error_messages))
    print(str(len(error_messages)) + ' Errors found during compilation.\n')
    sys.exit(1)
else:
    print('Compilation successful!')
```

Error logs:



```
(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ python -m scripts.typescript_checks --strict_checks
Compiling and testing typescript...

core/tests/services_sources/ATestFactory.ts(22,12): error TS7006: Parameter 'ATestFactory' implicitly has an 'any' type.

core/tests/services_sources/ATestFactory.ts(23,27): error TS7006: Parameter 'states' implicitly has an 'any' type.

core/tests/services_sources/ATestFactory.ts(24,7): error TS2683: 'this' implicitly has type 'any' because it does not have a type annotation.

3 Errors found during compilation.

(oppia) info@info-HP-Laptop-15s-eq0xxx:~/Desktop/Proposal-check/oppia$ 
```

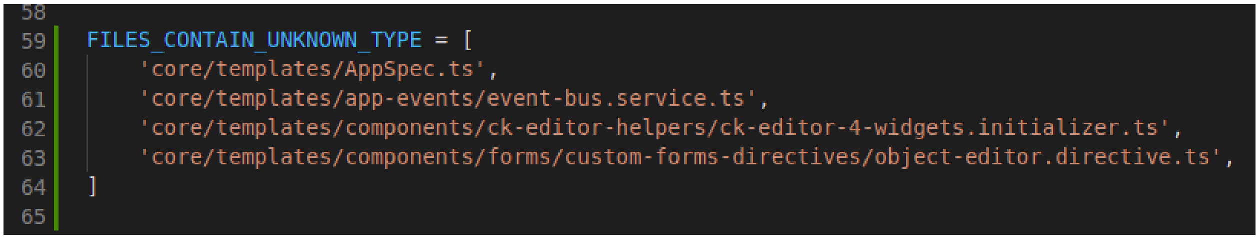
Goal 4: Get rid of the type “unknown” and “as unknown as X” pattern from the files we are going to type and add a lint check to prevent usage of “unknown” in the future.

Add lint check so that no new file with unknown type is added to the codebase. Also at places where we can't refactor to remove unknown we are going to add a ignore comment.

add lint check in **scripts/linters/js_ts_linter.py.**

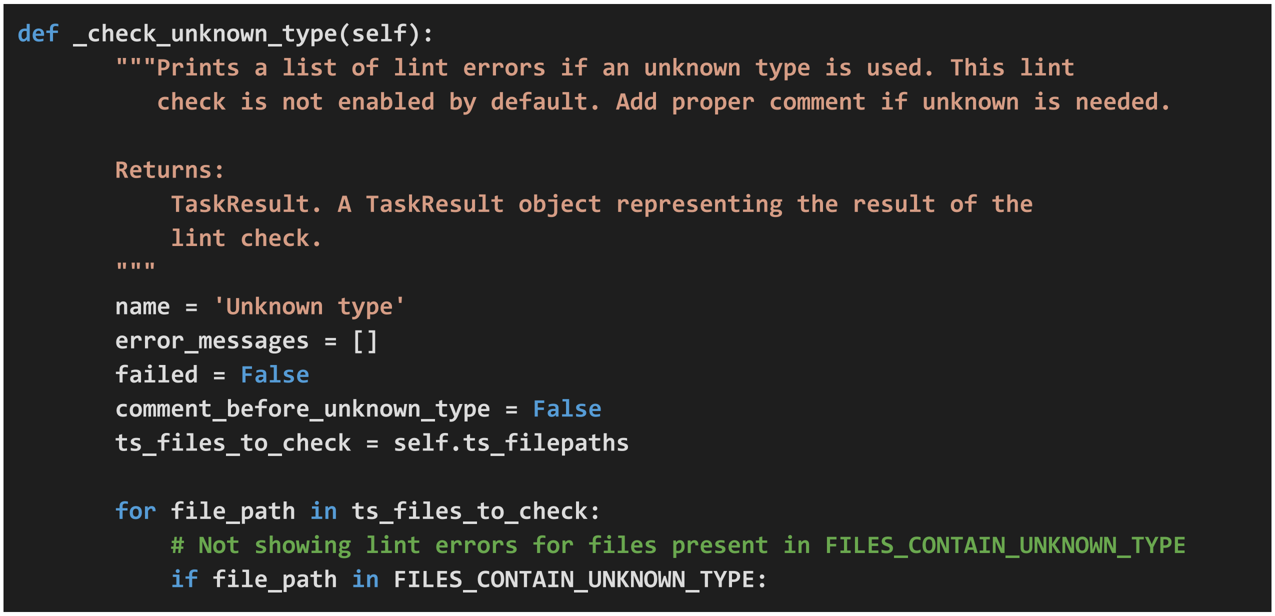
FILES_CONTAIN_UNKNOWN_TYPE: Contain files which are currently having unknown type. Remove files path from which we want to remove unknown type, similar to what we are doing while enabling force typescript strict in the newly added files.

Example:

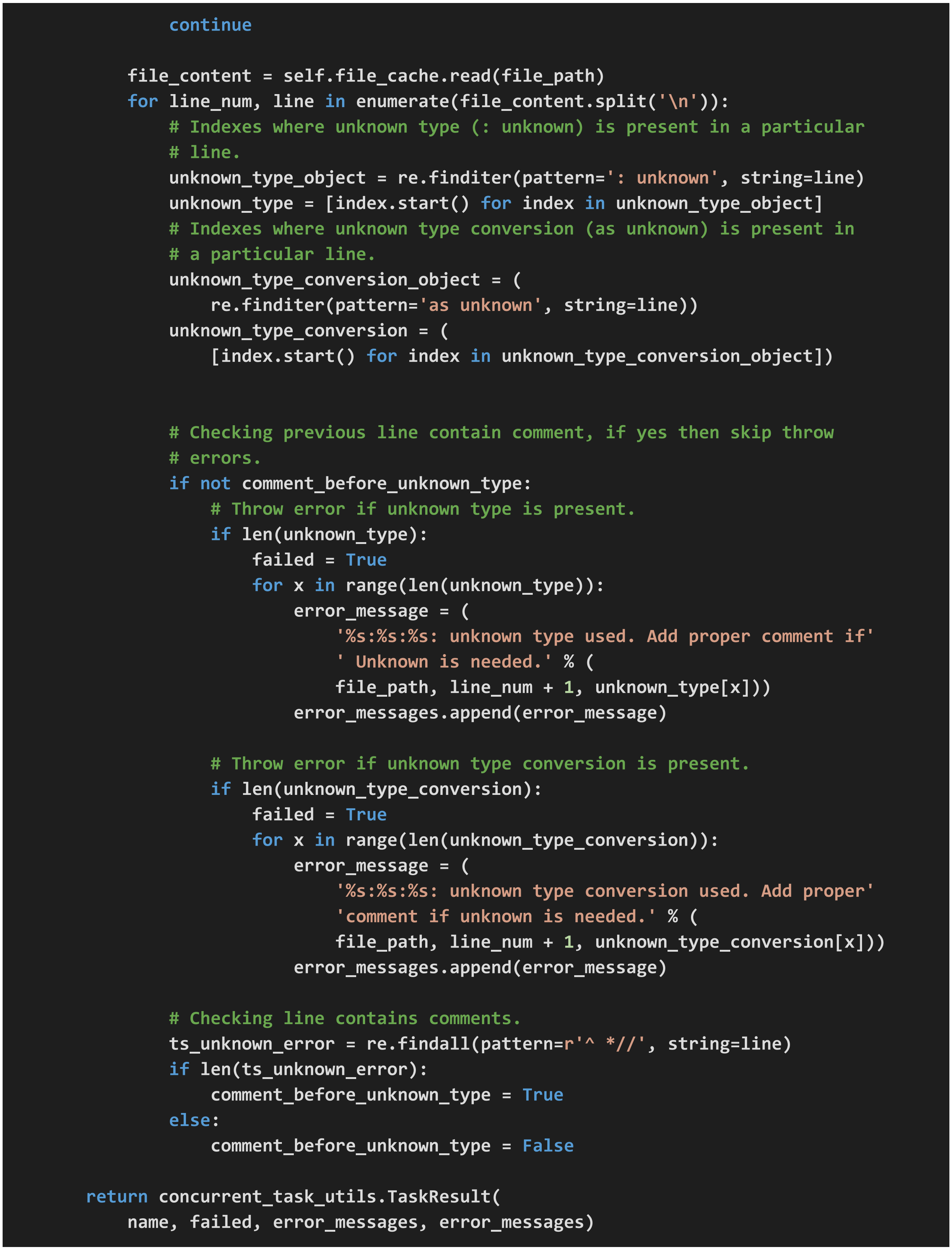


```
FILES_CONTAIN_UNKNOWN_TYPE = [
    'core/templates/AppSpec.ts',
    'core/templates/app-events/event-bus.service.ts',
    'core/templates/components/ck-editor-helpers/ck-editor-4-widgets.initializer.ts',
    'core/templates/components/forms/custom-forms-directives/object-editor.directive.ts',
]
```

Lint check:



```
def _check_unknown_type(self):
       """Prints a list of lint errors if an unknown type is used. This lint
          check is not enabled by default. Add proper comment if unknown is needed.

       Returns:
           TaskResult. A TaskResult object representing the result of the
           lint check.
       """
       name = 'Unknown type'
       error_messages = []
       failed = False
       comment_before_unknown_type = False
       ts_files_to_check = self.ts_filepaths

       for file_path in ts_files_to_check:
           # Not showing lint errors for files present in FILES_CONTAIN_UNKNOWN_TYPE
           if file_path in FILES_CONTAIN_UNKNOWN_TYPE:
```

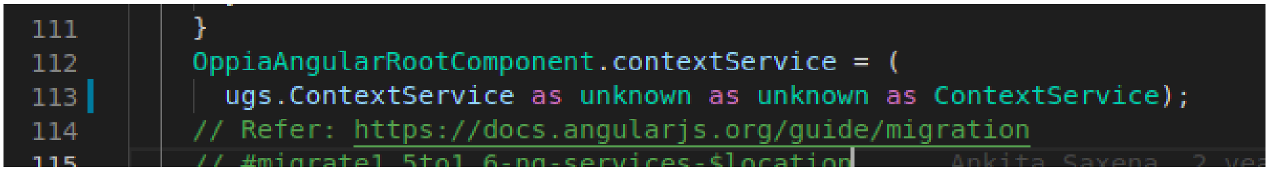


```
                continue

            file_content = self.file_cache.read(file_path)
            for line_num, line in enumerate(file_content.split('\n')):
                # Indexes where unknown type (: unknown) is present in a particular
                # line.
                unknown_type_object = re.finditer(pattern=': unknown', string=line)
                unknown_type = [index.start() for index in unknown_type_object]
                # Indexes where unknown type conversion (as unknown) is present in
                # a particular line.
                unknown_type_conversion_object = (
                    re.finditer(pattern='as unknown', string=line))
                unknown_type_conversion = (
                    [index.start() for index in unknown_type_conversion_object])


                # Checking previous line contain comment, if yes then skip throw
                # errors.
                if not comment_before_unknown_type:
                    # Throw error if unknown type is present.
                    if len(unknown_type):
                        failed = True
                        for x in range(len(unknown_type)):
                            error_message = (
                                '%s:%s:%s: unknown type used. Add proper comment if'
                                ' Unknown is needed.' % (
                                file_path, line_num + 1, unknown_type[x]))
                            error_messages.append(error_message)

                    # Throw error if unknown type conversion is present.
                    if len(unknown_type_conversion):
                        failed = True
                        for x in range(len(unknown_type_conversion)):
                            error_message = (
                                '%s:%s:%s: unknown type conversion used. Add proper'
                                'comment if unknown is needed.' % (
                                file_path, line_num + 1, unknown_type_conversion[x]))
                            error_messages.append(error_message)

                # Checking line contains comments.
                ts_unknown_error = re.findall(pattern=r'^ *//', string=line)
                if len(ts_unknown_error):
                    comment_before_unknown_type = True
                else:
                    comment_before_unknown_type = False

        return concurrent_task_utils.TaskResult(
            name, failed, error_messages, error_messages)
```

Add check in perform_all_lint_checks():

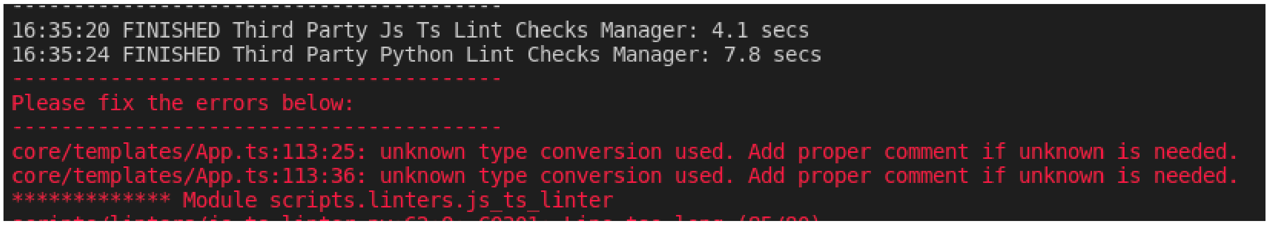
```
def perform_all_lint_checks(self):
        ...
        linter_stdout.append(self._check_constants_declaration())
        linter_stdout.append(self._check_angular_services_index())
        # Addition of unknown type check
        linter_stdout.append(self._check_unknown_type())

        ...
        return linter_stdout
```

Remove file core/templates/App.ts containing unknowns:



```
111      }
112      OppiaAngularRootComponent.contextService = (
113        ugs.ContextService as unknown as unknown as ContextService);
114      // Refer: https://docs.angularjs.org/guide/migration
115      // #migrate1.5to1.6-ng-services-$location          Ankita Saxena, 2 yea
```

Errors log:



```
----------------------------------------
16:35:20 FINISHED Third Party Js Ts Lint Checks Manager: 4.1 secs
16:35:24 FINISHED Third Party Python Lint Checks Manager: 7.8 secs
----------------------------------------
Please fix the errors below:
----------------------------------------
core/templates/App.ts:113:25: unknown type conversion used. Add proper comment if unknown is needed.
core/templates/App.ts:113:36: unknown type conversion used. Add proper comment if unknown is needed.
************* Module scripts.linters.js_ts_linter
scripts/linters/js_ts_linter.py:63:0: C0301: Line too long (85/80)
```

We're currently utilising unknown as a type in a lot of places in our codebase. We use an explicit conversion to type unknown to convert the variable to the desired type. Instead of explicitly converting that variable to that type, this can be fixed simply assigning the correct type to it.

Type 1: We use some property/key of actual type in variable in several frontend testing scripts, forcing us to use unknown. It is simple to solve by creating a complete type variable.

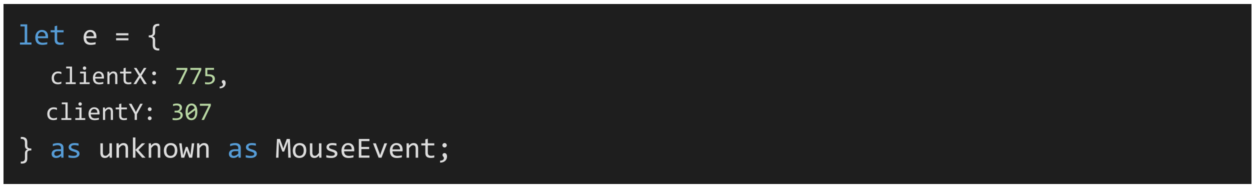
```
const lostChanges = [{
   cmd: 'add_state',
   state_name: 'State name',
 } as unknown as LostChange];
```

Creating complete type variable using createNew function of lostChangeObjectFactory

```
component.lostChanges = [lostChangeObjectFactory.createNew({
     cmd: 'add_state',
     state_name: 'State name',
   })];
```

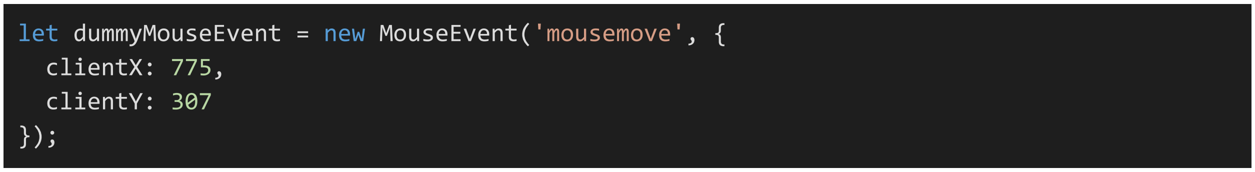
Similar things happen with MouseEvents, KeyboardEvents in order to test some functions.

Unknown with MouseEvent



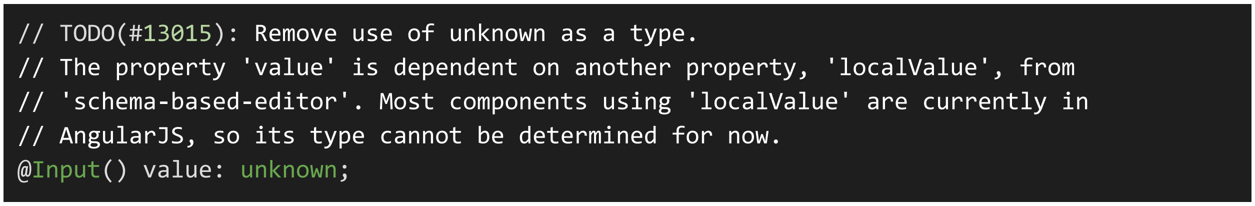
```
let e = {
  clientX: 775,
  clientY: 307
} as unknown as MouseEvent;
```

Solution:



```
let dummyMouseEvent = new MouseEvent('mousemove', {
  clientX: 775,
  clientY: 307
});
```

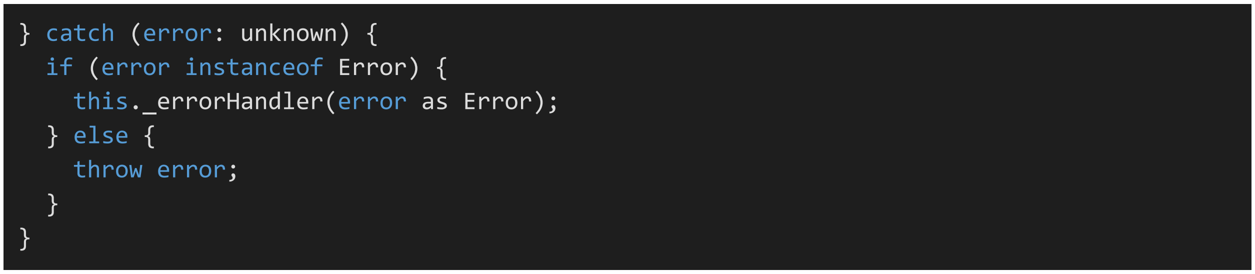
Type 2: Many of the components in AngularJs that pass values and are used by the @Input() attribute are of type unknown, therefore their type cannot be determined for the time being.



```
// TODO(#13015): Remove use of unknown as a type.
// The property 'value' is dependent on another property, 'localValue', from
// 'schema-based-editor'. Most components using 'localValue' are currently in
// AngularJS, so its type cannot be determined for now.
@Input() value: unknown;
```

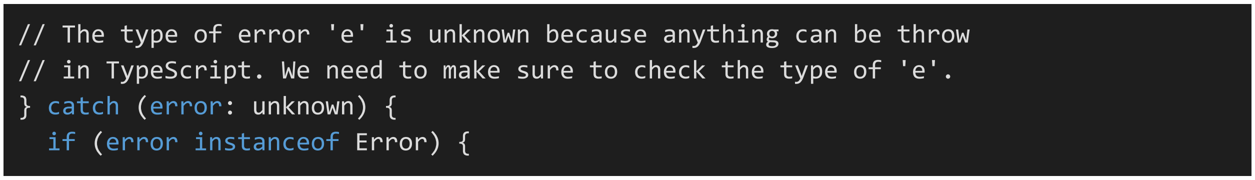
Solution: Add TODO(#13015) for now and solve once we are done with migration.

Type 3: Rare exceptions where unknown is needed (like in error handling), there should be appropriate type guards and it should be supplemented with proper comments.



```
} catch (error: unknown) {
  if (error instanceof Error) {
    this._errorHandler(error as Error);
  } else {
    throw error;
  }
}
```

Solution:



```
// The type of error 'e' is unknown because anything can be throw
// in TypeScript. We need to make sure to check the type of 'e'.
} catch (error: unknown) {
  if (error instanceof Error) {
    this._errorHandler(error as Error);
  } else {
    throw error;
  }
}
```

Here it is incorrect to use “throw error”, because the else case is not being taken care of properly -- if “else” case happens, we should do something with “error”, else throw a proper error message to prevent run-time errors.

```
// The type of error 'e' is unknown because anything can be throw
// in TypeScript. We need to make sure to check the type of 'e'.
} catch (error: unknown) {
  if (error instanceof Error) {
    this._errorHandler(error as Error);
  } else {
    throw new Error('Unexpected error response.');
  }
}
```

Files containing unknown:

All of the files are covered in issue #13015.

Third-party Libraries

No third party Libraries need to be added.

“Service” Dependencies

No dependencies.

Impact on Other Oppia Teams

Angular Migration: Strict typing is now enabled by default. It will highlight errors related to strict typing whenever we change file names during migration, for example, from directive.ts to component.ts (similar to spec files). As component.ts is a newly added file, it will act as such. When there is no need to alter the file name, such as during services and a file with the name component.ts exists, we will not encounter any errors. That files will be going to cover by removing their name from NOT_FULLY_COVERED_FILENAMES.

No impact on other teams, just strict typing is now enabled by default. Check strict typing of any of your newly added files locally, no other requirements like push code, adding, removing, updating files path in tsconfig-strict.json etc.

Note: If we are strictly typing a file while another contributor is working on the same file but refactoring something else on his/her branch, then our code might affect the other individual working on the same file after it is merged into develop. This can be avoided by keeping a specific branch up to date with develop branch right before merging.

Impact from Other Oppia Teams

Angular Migration: If the entire migration has been completed, and all files from NOT_FULLY_COVERED_FILENAMES have been completed, we may simply delete tsconfig-strict.json and use only tsconfig.json with strict as true. This can be done by replacing the path tsconfig-strict.json with tsconfig.json and then removing tsconfig-strict.json.

No impact from other teams.

Risks and mitigations

Potential Risk	Mitigation
During typing in many places we need to refactor code, which may lead to some non-working frontend changes.	Debug refactor files thoroughly and add video proof on the PR description of the file in which refactoring is done.

Implementation Approach

Firstly apply strict typing in the whole code base ,customize error log and Add lint check so that no new file with unknown type is added to the codebase. After that, start choosing files from NOT_FULLY_COVERED_FILENAMES and do strict typing. At last remove unknown from files which are not present in NOT_FULLY_COVERED_FILENAMES(files already strict but using unknowns)

Examining the locations where we can obtain Angular files:

Directories	No. of files
core/template/*.ts	2
core/templates/components/	87
core/templates/domain/	24
core/templates/filters/	6
core/templates/pages/	221
core/templates/services/	17
core/templates/tests/	1
core/tests/	1

extensions/interactions/	59
extensions/objects/	8
extensions/rich_text_components/	4

Github Gist of above list.

Dependencies table of directories

core/template/*.ts

File path	Dependencies
1. Polyfills.ts 2. app.constants.ajs.ts	N/A

core/templates/components/

File path	Dependencies
1. Ck-editor-4-widgets.initializer.ts 2. Ck-editor-copy-toolbar.component.spec.ts 3. Ck-editor-copy-toolbar.component.ts 4. Concept-card.component.spec.ts 5. Concept-card.component.ts 6. Topic-creation.service.spec.ts 7. Topic-creation.service.ts 8. apply-validation.directive.spec.ts 9. apply-validation.directive.ts 10. object-editor.directive.ts 11. Thumbnail-uploader.component.spec.ts 12. Thumbnail-uploader.component.ts 13. schema-based-bool-editor.component.spec.ts 14. Schema-based-bool-editor.component.ts 15. schema-based-choices-editor.component.spec.ts 16. Schema-based-choices-editor.component.ts 17. schema-based-custom-editor.component.spec.ts 18. Schema-based-custom-editor.component.ts 19. Schema-based-dict-editor.component.spec.ts 20. Schema-based-dict-editor.component.ts 21. Schema-based-editor.component.spec.ts 22. Schema-based-editor.component.ts 23. Schema-based-expression-editor.component.spec.ts 24. Schema-based-expression-editor.component.ts 25. Schema-based-float-editor.component.spec.ts 26. Schema-based-float-editor.component.ts 27. Schema-based-html-editor.component.spec.ts 28. Schema-based-html-editor.component.ts 29. Schema-based-int-editor.component.spec.ts 30. Schema-based-int-editor.component.ts	N/A

31. Schema-based-list-editor.component.spec.ts 32. Schema-based-list-editor.component.ts 33. Schema-based-unicode-editor.component.spec.ts 34. schema-based-unicode-editor.component.ts 35. Question-editor-save-modal.component.ts 36. Question-misconception-editor.component.spec.ts 37. Question-misconception-editor.component.ts 38. Question-misconception-selector.component.spec.ts 39. Question-misconception-selector.component.ts 40. Rating-display.component.ts 41. Review-material-editor.component.ts 42. Save-pending-changes-modal.component.ts 43. Score-ring.component.ts 44. Select-skill-modal.component.spec.ts 45. stale-tab-info-modal.component.ts 46. Hint-editor.component.spec.ts 47. Hint-editor.component.ts 48. Outcome-destination-editor.component.spec.ts 49. Outcome-destination-editor.component.ts 50. Outcome-editor.component.spec.ts 51. outcome-editor.component.ts 52. Response-header.component.spec.ts 53. Response-header.component.ts 54. Solution-editor.component.spec.ts 55. Solution-editor.component.ts 56. Solution-explanation-editor.component.ts 57. State-hints-editor.component.spec.ts 58. State-hints-editor.component.ts 59. State-interaction-editor.component.spec.ts 60. State-interaction-editor.component.ts 61. State-solution-editor.component.spec.ts 62. State-solution-editor.component.ts 63. Completion-graph.component.ts 64. Learner-story-summary-tile.component.spec.ts 65. Learner-story-summary-tile.component.ts 66. Learner-topic-goals-summary-tile.component.spec.ts 67. learner-topic-goals-summary-tile.component.ts 68. Learner-topic-summary-tile.component.spec.ts 69. Learner-topic-summary-tile.component.ts 70. Story-summary-tile.component.spec.ts 71. Subtopic-summary-tile.component.spec.ts 72. Subtopic-summary-tile.component.ts 73. Topic-summary-tile.component.ts 74. unsaved-changes-status-info-modal.component.ts	
1. hint-and-solution-buttons.component.spec.ts 2. hint-and-solution-buttons.component.ts	● display-hint-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● playthrough.service.ts

1. ck-editor-4-rte.component.ts	● ck-editor-4-widgets.initializer.ts ● oppia-angular-root.component.ts
1. oppia-angular-root.component.spec.ts 2. oppia-angular-root.component.ts	● ck-editor-4-widgets.initializer.ts
1. tag-misconception-modal-component.spec.ts 2. tag-misconception-modal-component.ts	● question-misconception-editor.component.ts
1. rubrics-editor.component.spec.ts 2. rubrics-editor.component.ts	● angular-html-bind-wrapper.directive.ts ● schema-based-editor.directive.ts
1. state-content-editor.component.spec.ts 2. state-content-editor.component.ts	● change-list.service.ts
1. state-skill-editor.component.spec.ts 2. state-skill-editor.component.ts	● story-editor-state.service.ts

core/templates/domain/

File path	Dependencies
1. Collection-update.service.spec.ts 2. Collection-update.service.ts 3. StatesObjectFactorySpec.ts 4. Param-metadata.model.spec.ts 5. Read-only-exploration-backend-api.service.spec.ts 6. Pretest-question-backend-api.service.spec.ts 7. Question-backend-api.service.spec.ts 8. Skill-update.service.spec.ts 9. Skill-update.service.ts 10. State-card.model.spec.ts 11. Learner-answer-info.model.ts 12. Editable-story-backend-api.service.spec.ts 13. SubtopicPage.model.spec.ts 14. Learner-topic-summary.model.spec.ts 15. Newly-created-story.model.spec.ts 16. Topic-rights.model.spec.ts 17. Topic-update.service.spec.ts 18. Topic-update.service.ts	N/A
1. Editable-collection-backend-api.service.spec.ts 2. Undo-redo.service.spec.ts	Contain UpgradedServices.ts
1. Editable-exploration-backend-api.service.spec.ts 2. Editable-question-backend-api.service.spec.ts	● ck-editor-4-widgets.initializer.ts ● topic-creation.service.ts ● collection-update.service.ts ● editable-exploration-backend-api.service.spec.ts ● skill-update.service.ts ● story-update.service.ts

	● topic-update.service.ts ● format-timer.pipe.ts ● collection-editor-routing.service.ts ● collection-editor-state.service.ts ● collection-linearizer.service.ts ● collection-player-page.component.ts ● contribution-and-review-backend-api.service.ts ● contribution-and-review.service.ts ● contribution-opportunities.service.ts ● translate-text.service.ts ● responses.service.ts ● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-language-code.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts ● exploration-states.service.ts ● exploration-tags.service.ts ● graph-data.service.ts ● user-email-preferences.service.ts ● translation-topic.service.ts ● display-hint-modal.component.ts ● flag-exploration-modal.component.ts ● refresher-exploration-confirmation-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-local-nav-backend-api.service.ts ● learner-view-rating.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● story-editor-state.service.ts ● create-new-skill-modal.service.ts ● subtopic-validation.service.ts ● topic-editor-routing.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts ● angular-services.index.ts ● exploration-improvements-task-registry.service.ts ● oppia-rte-parser.service.ts ● playthrough.service.ts ● state-interaction-stats.service.ts ● state-top-answers-stats.service.ts ● unit-test-utils.ajs.ts
1. Story-update.service.spec.ts 2. Story-update.service.ts	● story-editor-state.service.ts

core/templates/filters/

File path	Dependencies
1. Format-timer.pipe.ts 2. Remove-duplicates-in-array.pipe.spec.ts 3. Get-abbreviated-text.pipe.spec.ts 4. Replace-inputs-with-ellipses.pipe.spec.ts 5. Truncate-at-first-ellipsis.pipe.spec.ts 6. underscores-to-camel-case.pipe.spec.ts	N/A

core/templates/pages/

File path	Dependencies
admin-page	
1. admin-roles-tab.component.spec.ts 2. admin-roles-tab.component.ts	● topic-manager-role-editor-modal.component.ts
blog-dashboard-page	
1. blog-dashboard-tile.component.ts 2. Blog-post-editor.component.spec.ts 3. Blog-post-editor.component.ts 4. blog-card-preview-modal.component.spec.ts 5. Blog-card-preview-modal.component.ts	N/A
classroom-page	
1. classroom-page.module.ts	● topic-summary-tile.component.ts
collection-editor-page	
1. Collection-editor-save-modal.component.ts 2. Collection-editor-routing.service.ts 3. Collection-editor-state.service.spec.ts 4. Collection-editor-state.service.ts	N/A
1. Collection-editor-tab.component.spec.ts 2. Collection-editor-tab.component.ts 3. Collection-linearizer.service.spec.ts 4. Collection-linearizer.service.ts 5. Collection-details-editor.component.spec.ts 6. Collection-details-editor.component.ts	● collection-update.service.ts
1. collection-permissions-card.component.ts	● collection-editor-state.service.ts
1. collection-node-creator.component.spec.ts 2. collection-node-creator.component.ts	● collection-update.service.ts ● collection-linearizer.service.ts
1. collection-editor-pre-publish-modal.component.spec.ts	● collection-update.service.ts ● collection-editor-state.service.ts

2. collection-editor-pre-publish-modal.component.ts	
1. collection-editor-navbar-breadcrumb.component.spec.ts 2. collection-editor-navbar-breadcrumb.component.ts	● collection-editor-routing.service.ts ● collection-editor-state.service.ts
1. collection-editor-navbar.component.spec.ts 2. collection-editor-navbar.component.ts	● collection-update.service.ts ● collection-editor-pre-publish-modal.component.ts
1. collection-node-editor.component.spec.ts 2. collection-node-editor.component.ts	● collection-update.service.ts ● collection-editor-state.service.ts ● collection-linearizer.service.ts
collection-player-page	
1. Collection-local-nav.component.ts 2. Collection-navbar.component.ts 3. Collection-node-list.component.ts 4. Collection-player-page.component.spec.ts 5. Collection-player-page.component.ts	N/A
contributor-dashboard-admin-page	
1. Contributor-dashboard-admin-navbar.component.ts 2. Contributor-dashboard-admin-backend-api.service.spec.ts	N/A
contributor-dashboard-page	
1. translation-opportunities.component.spec.ts	N/A
2. Translation-modal.component.spec.ts 3. Translation-modal.component.ts	● ck-editor-4-widgets.initializer.ts ● oppia-angular-root.component.ts ● translate-text.service.ts
1. translate-text.service.spec.ts 2. translate-text.service.ts	● ck-editor-4-widgets.initializer.ts ● oppia-angular-root.component.ts
creator-dashboard-page	
1. creator-dashboard-page.component.spec.ts 2. creator-dashboard-page.component.ts	N/A
donate-page	
1. Donate-page.component.spec.ts	N/A
error-pages	
1. Error-page-root.component.ts 2. Error-page.component.ts	N/A
exploration-editor-page	

1. Changes-in-human-readable-form.component.spec.ts 2. Changes-in-human-readable-form.component.ts 3. Responses.service.spec.ts 4. Responses.service.ts 5. Solution-verification.service.spec.ts 6. Customize-interaction-modal.component.spec.ts 7. Customize-interaction-modal.component.ts 8. Revert-exploration-modal.component.ts 9. Preview-set-parameters-modal.component.ts 10. Autosave-info-modals.service.spec.ts 11. Change-list.service.spec.ts 12. Change-list.service.ts 13. Exploration-data.service.spec.ts 14. exploration-rights.service.spec.ts 15. Exploration-rights.service.ts 16. Exploration-tags.service.ts 17. Exploration-title.service.spec.ts 18. Graph-data.service.ts 19. User-email-preferences.service.ts 20. Moderator-unpublish-exploration-modal.component.ts 21. Translation-tab-busy-modal.component.ts 22. welcome-translation-modal.component.ts	N/A
1. Translation-topic.service.spec.ts 2. Translation-topic.service.ts	● contribution-opportunities.service.ts
1. State-param-changes-editor.component.spec.ts 2. State-param-changes-editor.component.ts 3. Exploration-editor-tab.component.spec.ts 4. Exploration-editor-tab.component.ts 5. param-changes-editor.component.spec.ts 6. Param-changes-editor.component.ts	● ck-editor-4-widgets.initializer.ts ● topic-creation.service.ts ● collection-update.servicets ● skill-update.service.ts ● story-update.service.ts ● topic-update.service.ts ● format-timer.pipe.ts ● collection-editor-routing.service.ts ● collection-editor-state.service.ts ● collection-linearizer.service.ts ● collection-player-page.component.ts ● contribution-and-review-backend-api.service.ts ● contribution-and-review.service.ts ● contribution-opportunities.service.ts ● translate-text.service.ts ● responses.service.ts ● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-language-code.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts ● exploration-states.service.ts ● exploration-tags.service.ts ● graph-data.service.ts ● user-email-preferences.service.ts

	● translation-topic.service.ts ● display-hint-modal.component.ts ● flag-exploration-modal.component.ts ● refresher-exploration-confirmation-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-local-nav-backend-api.service.ts ● learner-view-rating.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● story-editor-state.service.ts ● create-new-skill-modal.service.ts ● subtopic-validation.service.ts ● topic-editor-routing.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts ● angular-services.index.ts ● exploration-improvements-task-registry.service.ts ● oppia-rte-parser.service.ts ● playthrough.service.ts ● state-interaction-stats.service.ts ● state-top-answers-stats.service.ts ● unit-test-utils.ajs.ts
1. exploration-init-state-name.service.spec.ts 2. Exploration-init-state-name.service.ts 3. exploration-language-code.service.spec.ts 4. Exploration-language-code.service.ts 5. exploration-correctness-feedback.service.ts	● change-list.service.ts ● exploration-property.service.ts
1. exploration-property.service.spec.ts 2. exploration-property.service.ts	● change-list.service.ts
1. exploration-states.service.ts	● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts
1. preview-summary-tile-modal.component.ts	● change-list.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts
exploration-player-page	
1. supplemental-card.component.spec.ts 2. Supplemental-card.component.ts 3. display-hint-modal.component.spec.ts	N/A

4. Display-hint-modal.component.ts 5. Flag-exploration-modal.component.ts 6. Answer-classification.service.spec.ts 7. extract-image-filenames-from-model.service.spec.ts 8. Extract-image-filenames-from-model.service.ts 9. learner-answer-info.service.spec.ts 10. Learner-answer-info.service.ts 11. question-player-engine.service.spec.ts 12. Question-player-engine.service.ts 13. state-classifier-mapping.service.spec.ts	
1. exploration-engine.service.spec.ts 2. exploration-engine.service.ts	● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● stats-reporting.service.ts ● playthrough.service.ts
1. feedback-popup.component.spec.ts 2. Feedback-popup.component.ts 3. learner-local-nav.component.spec.ts 4. Learner-local-nav.component.ts 5. progress-nav.component.spec.ts 6. Progress-nav.component.ts 7. Learner-answer-info-card.component.ts 8. ratings-and-recommendations.component.spec.ts 9. Ratings-and-recommendations.component.ts 10. Refresher-exploration-confirmation-modal.component.ts 11. exploration-player-state.service.spec.ts 12. Exploration-player-state.service.ts 13. learner-view-rating.service.spec.ts 14. learner-view-rating.service.ts	● exploration-engine.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● stats-reporting.service.ts ● playthrough.service.ts
1. image-preloader.service.spec.ts 2. image-preloader.service.ts	● extract-image-filenames-from-model.service.ts
1. learner-local-nav-backend-api.service.ts	● flag-exploration-modal.component.ts
1. stats-reporting.service.spec.ts 2. stats-reporting.service.ts	● playthrough.service.ts
1. content-language-selector.component.spec.ts 2. content-language-selector.component.ts	● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts
1. learner-view-info.component.spec.ts 2. learner-view-info.component.ts	● stats-reporting.service.ts ● playthrough.service.ts
1. conversation-skin.component.spec.ts 2. conversation-skin.component.ts	● collection-player-page.component.ts ● exploration-engine.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● stats-reporting.service.ts ● playthrough.service.ts

1. tutor-card.component.spec.ts 2. Tutor-card.component.ts 3. Exploration-footer.component.spec.ts 4. exploration-footer.component.ts	● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-answer-info.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● playthrough.service.ts
landing-page	
1. Topic-landing-page.component.spec.ts 2. topic-landing-page.component.ts	N/A
learner-dashboard-page	
1. Community-lessons-tab.component.spec.ts 2. Community-lessons-tab.component.ts 3. goals-tab.component.spec.ts 4. Goals-tab.component.ts 5. Home-tab.component.ts 6. learner-dashboard-page.component.spec.ts 7. Learner-dashboard-page.component.ts 8. progress-tab.component.spec.ts 9. progress-tab.component.ts	N/A
library-page	
1. Library-footer.component.ts 2. Search-bar.component.spec.ts 3. activity-tiles-infinity-grid.component.spec.ts 4. activity-tiles-infinity-grid.component.ts 5. search-results.component.ts	N/A
oppia-root	
1. can-access-splash-page.guard.spec.ts 2. can-access-splash-page.guard.ts	N/A
1. App.routing.module.ts 2. app.module.ts	So many dependencies
profile-page	
1. profile-page.component.spec.ts 2. profile-page.component.ts	N/A
release-coordinator-page	
1. Cancel-beam-job-dialog.component.spec.ts 2. Cancel-beam-job-dialog.component.ts 3. Start-new-beam-job-dialog.component.spec.ts 4. Start-new-beam-job-dialog.component.ts	N/A

5. View-beam-job-output-dialog.component.spec.ts 6. View-beam-job-output-dialog.component.ts 7. Release-coordinator-navbar.component.spec.ts 8. Release-coordinator-navbar.component.ts 9. Release-coordinator-backend-api.service.spec.ts	
1. beam-jobs-tab.component.spec.ts 2. beam-jobs-tab.component.ts	● cancel-beam-job-dialog.component.ts ● start-new-beam-job-dialog.component.ts ● view-beam-job-output-dialog.component.ts
signup-page	
1. signup-page.component.spec.ts 2. signup-page.component.ts	N/A
skill-editor-page	
1. Skill-preview-modal.component.ts 2. skill-editor-state.service.spec.ts	N/A
1. worked-example-editor.component.spec.ts 2. Worked-example-editor.component.ts 3. skill-description-editor.component.spec.ts 4. Skill-description-editor.component.ts 5. misconception-editor.component.spec.ts 6. Misconception-editor.component.ts 7. skill-misconceptions-editor.component.spec.ts 8. Skill-misconceptions-editor.component.ts 9. skill-rubrics-editor.component.ts	● skill-update.service.ts
1. skill-concept-card-editor.component.spec.ts 2. skill-concept-card-editor.component.ts	● skill-update.service.ts ● skill-preview-modal.component.ts
1. skill-prerequisite-skills-editor.component.spec.ts 2. skill-prerequisite-skills-editor.component.ts	● select-skill-modal.component.ts ● skill-update.service.ts
1. skill-editor-staleness-detection.service.ts	● Stale-tab-info-modal.component.ts ● unsaved-changes-status-info-modal.component.ts
splash-page	
1. splash-page.component.spec.ts 2. splash-page.component.ts	N/A
story-editor-page	
1. story-editor-save-modal.component.ts	N/A
1. Story-editor-navbar-breadcrumb.component.spec.ts 2. Story-editor-navbar-breadcrumb.component.ts 3. Story-editor-navbar.component.spec.ts 4. Story-editor-navbar.component.ts	● save-pending-changes-modal.component.ts ● story-editor-state.service.ts ● story-editor-save-modal.component.ts

1. story-preview-tab.component.spec.ts 2. story-preview-tab.component.ts	● story-editor-state.service.ts
1. story-editor-staleness-detection.service.ts	● Stale-tab-info-modal.component.ts ● unsaved-changes-status-info-modal.component.ts
1. story-editor-state.service.spec.ts 2. story-editor-state.service.ts	● ck-editor-4-widgets.initializer.ts ● topic-creation.service.ts ● collection-update.servicets ● skill-update.service.ts ● story-update.service.ts ● topic-update.service.ts ● format-timer.pipe.ts ● collection-editor-routing.service.ts ● collection-editor-state.service.ts ● collection-linearizer.service.ts ● collection-player-page.component.ts ● contribution-and-review-backend-api.service.ts ● contribution-and-review.service.ts ● contribution-opportunities.service.ts ● translate-text.service.ts ● responses.service.ts ● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-language-code.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts ● exploration-states.service.ts ● exploration-tags.service.ts ● graph-data.service.ts ● user-email-preferences.service.ts ● translation-topic.service.ts ● display-hint-modal.component.ts ● flag-exploration-modal.component.ts ● refresher-exploration-confirmation-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-local-nav-backend-api.service.ts ● learner-view-rating.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● story-editor-state.service.ts ● create-new-skill-modal.service.ts ● subtopic-validation.service.ts ● topic-editor-routing.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts ● angular-services.index.ts

	● exploration-improvements-task-registry.service.ts ● oppia-rte-parser.service.ts ● playthrough.service.ts ● state-interaction-stats.service.ts ● state-top-answers-stats.service.ts ● unit-test-utils.ajs.ts
story-viewer-page	
1. Story-viewer-navbar-breadcrumb.component.ts 2. Story-viewer-navbar-pre-logo-action.component.ts 3. story-viewer-page.component.spec.ts 4. story-viewer-page.component.ts	N/A
subtopic-viewer-page	
1. Subtopic-viewer-navbar-breadcrumb.component.spec.ts 2. Subtopic-viewer-navbar-breadcrumb.component.ts 3. Subtopic-viewer-navbar-pre-logo-action.component.ts 4. subtopic-viewer-page.component.ts	N/A
teach-page	
1. teach-page.component.spec.ts 2. teach-page.component.ts	N/A
topic-editor-page	
1. Questions-list-select-skill-and-difficulty-modal.component.spec.ts 2. Questions-list-select-skill-and-difficulty-modal.component.ts 3. Topic-editor-save-modal.component.ts 4. Topic-editor-routing.service.spec.ts 5. Topic-editor-routing.service.ts 6. Topic-editor-state.service.spec.ts 7. Topic-editor-state.service.ts	N/A
1. Change-subtopic-assignment-modal.component.ts 2. create-new-subtopic-modal.component.spec.ts 3. create-new-subtopic-modal.component.ts	● topic-update.service.ts ● subtopic-validation.service.ts ● topic-editor-state.service.ts
1. questions-opportunities-select-difficulty-modal.component.spec.ts 2. questions-opportunities-select-difficulty-modal.component.ts	● extract-image-filenames-from-model.service.ts
1. Topic-editor-navbar-breadcrumb.component.ts 2. subtopic-preview-tab.component.spec.ts 3. subtopic-preview-tab.component.ts	● topic-editor-routing.service.ts ● topic-editor-state.service.ts

1. topic-preview-tab.component.spec.ts 2. Topic-preview-tab.component.ts 3. subtopic-validation.service.ts	● topic-editor-state.service.ts
1. create-new-skill-modal.service.spec.ts 2. create-new-skill-modal.service.ts	● create-new-skill-modal.component.ts
topic-viewer-page	
1. Topic-viewer-stories-list.component.ts 2. Subtopics-list.component.ts 3. Topic-viewer-page.component.spec.ts 4. topic-viewer-navbar-breadcrumb.component.spec.ts	N/A
topics-and-skills-dashboard-page	
1. Create-new-skill-modal.component.ts 2. delete-topic-modal.component.ts	N/A
1. topics-list.component.spec.ts 2. topics-list.component.ts	● delete-topic-modal.component.ts
1. topics-and-skills-dashboard-page.component.spec.ts 2. topics-and-skills-dashboard-page.component.ts	● topic-creation.service.ts ● create-new-skill-modal.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts

core/templates/services/

File path	**Dependencies**
1. oppia-rte-parser.service.spec.ts 2. Oppia-rte-parser.service.ts 3. playthrough.service.spec.ts 4. Playthrough.service.ts 5. Promo-bar-backend-api.service.spec.ts 6. Question-validation.service.spec.ts 7. Questions-list.service.spec.ts 8. state-interaction-stats.service.spec.ts 9. State-interaction-stats.service.ts 10. state-top-answers-stats.service.spec.ts 11. state-top-answers-stats.service.ts	N/A
1. exploration-improvements.service.spec.ts 2. exploration-improvements.service.ts	● ck-editor-4-widgets.initializer.ts ● topic-creation.service.ts ● collection-update.servicets ● skill-update.service.ts ● story-update.service.ts ● topic-update.service.ts ● format-timer.pipe.ts ● collection-editor-routing.service.ts

	● collection-editor-state.service.ts ● collection-linearizer.service.ts ● collection-player-page.component.ts ● contribution-and-review-backend-api.service.ts ● contribution-and-review.service.ts ● contribution-opportunities.service.ts ● translate-text.service.ts ● responses.service.ts ● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-language-code.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts ● exploration-states.service.ts ● exploration-tags.service.ts ● graph-data.service.ts ● user-email-preferences.service.ts ● translation-topic.service.ts ● display-hint-modal.component.ts ● flag-exploration-modal.component.ts ● refresher-exploration-confirmation-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-local-nav-backend-api.service.ts ● learner-view-rating.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● story-editor-state.service.ts ● create-new-skill-modal.service.ts ● subtopic-validation.service.ts ● topic-editor-routing.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts ● angular-services.index.ts ● exploration-improvements-task-registry.service.ts ● oppia-rte-parser.service.ts ● playthrough.service.ts ● state-interaction-stats.service.ts ● state-top-answers-stats.service.ts ● unit-test-utils.ajs.ts
1. UpgradedServices.ts 2. Angular-services.index.ts 3. Exploration-features.service.spec.ts 4. exploration-improvements-task-registry.service.ts	So many dependencies

core/templates/tests/

File path	Dependencies
1. unit-test-utils.ajs.ts	● ck-editor-4-widgets.initializer.ts ● topic-creation.service.ts ● collection-update.servicets ● skill-update.service.ts ● story-update.service.ts ● topic-update.service.ts ● format-timer.pipe.ts ● collection-editor-routing.service.ts ● collection-editor-state.service.ts ● collection-linearizer.service.ts ● collection-player-page.component.ts ● contribution-and-review-backend-api.service.ts ● contribution-and-review.service.ts ● contribution-opportunities.service.ts ● translate-text.service.ts ● responses.service.ts ● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-language-code.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts ● exploration-states.service.ts ● exploration-tags.service.ts ● graph-data.service.ts ● user-email-preferences.service.ts ● translation-topic.service.ts ● display-hint-modal.component.ts ● flag-exploration-modal.component.ts ● refresher-exploration-confirmation-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-local-nav-backend-api.service.ts ● learner-view-rating.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● story-editor-state.service.ts ● create-new-skill-modal.service.ts ● subtopic-validation.service.ts ● topic-editor-routing.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts ● angular-services.index.ts ● exploration-improvements-task-registry.service.ts ● oppia-rte-parser.service.ts

	● playthrough.service.ts ● state-interaction-stats.service.ts ● state-top-answers-stats.service.ts

core/tests/

File path	Dependencies
1. python-program.tokenizer.spec.ts	N/A

extensions/interactions/

File path	Dependencies
1. oppia-interactive-algebraic-expression-input.component.spec.ts 2. Oppia-interactive-algebraic-expression-input.component.ts 3. oppia-interactive-code-repl.component.spec.ts 4. Oppia-interactive-code-repl.component.ts 5. oppia-interactive-continue.component.spec.ts 6. Oppia-interactive-continue.component.ts 7. oppia-interactive-drag-and-drop-sort-input.component.spec.ts 8. Oppia-interactive-drag-and-drop-sort-input.component.ts 9. Oppia-response-drag-and-drop-sort-input.component.ts 10. Oppia-short-response-drag-and-drop-sort-input.component.ts 11. oppia-interactive-fraction-input.component.spec.ts 12. Oppia-interactive-fraction-input.component.ts 13. graph-viz.component.spec.ts 14. Graph-viz.component.ts 15. oppia-interactive-graph-input.component.spec.ts 16. Oppia-interactive-graph-input.component.ts 17. Oppia-response-graph-input.component.spec.ts 18. Oppia-short-response-graph-input.component.spec.ts 19. Oppia-response-image-click-input.component.spec.ts 20. oppia-interactive-interactive-map.component.spec.ts 21. Oppia-interactive-interactive-map.component.ts 22. oppia-interactive-item-selection-input.component.spec.ts 23. Oppia-interactive-item-selection-input.component.ts 24. Oppia-response-item-selection-input.component.ts 25. Oppia-short-response-item-selection-input.component.ts 26. oppia-interactive-math-equation-input.component.spec.ts	N/A

27. Oppia-interactive-math-equation-input.component.ts 28. Oppia-response-math-equation-input.component.ts 29. Oppia-short-response-math-equation-input.component.ts 30. oppia-interactive-multiple-choice-input.component.spec.ts 31. Oppia-interactive-multiple-choice-input.component.ts 32. oppia-interactive-number-with-units.component.spec.ts 33. Oppia-interactive-number-with-units.component.ts 34. Oppia-response-number-with-units.component.ts 35. Oppia-short-response-number-with-units.component.ts 36. Oppia-response-numeric-expression-input.component.ts 37. Oppia-short-response-numeric-expression-input.component.ts 38. oppia-interactive-numeric-input.component.spec.ts 39. Oppia-interactive-numeric-input.component.ts 40. Oppia-response-numeric-input.component.ts 41. Oppia-short-response-numeric-input.component.ts 42. Oppia-response-pencil-code-editor.component.ts 43. Oppia-short-response-pencil-code-editor.component.ts 44. oppia-interactive-ratio-expression-input.component.spec.ts 45. Oppia-interactive-ratio-expression-input.component.ts 46. Oppia-response-ratio-expression-input.component.ts 47. Oppia-short-response-ratio-expression-input.component.ts 48. oppia-interactive-set-input.component.spec.ts 49. Oppia-interactive-set-input.component.ts 50. oppia-interactive-text-input.component.spec.ts 51. Oppia-interactive-text-input.component.ts 52. Oppia-response-text-input.component.spec.ts 53. oppia-short-response-text-input.component.spec.ts	
1. oppia-interactive-image-click-input.component.spec.ts 2. oppia-interactive-image-click-input.component.ts	● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts
1. oppia-interactive-numeric-expression-input.component.spec.ts 2. oppia-interactive-numeric-expression-input.component.ts	● ck-editor-4-widgets.initializer.ts ● topic-creation.service.ts ● collection-update.service.ts ● skill-update.service.ts ● story-update.service.ts ● topic-update.service.ts ● collection-editor-routing.service.ts ● collection-editor-state.service.ts ● collection-linearizer.service.ts ● collection-player-page.component.ts ● contribution-and-review-backend-api.service.ts

	● contribution-and-review.service.ts ● contribution-opportunities.service.ts ● translate-text.service.ts ● responses.service.ts ● change-list.service.ts ● exploration-init-state-name.service.ts ● exploration-language-code.service.ts ● exploration-property.service.ts ● exploration-rights.service.ts ● exploration-states.service.ts ● exploration-tags.service.ts ● graph-data.service.ts ● user-email-preferences.service.ts ● translation-topic.service.ts ● display-hint-modal.component.ts ● flag-exploration-modal.component.ts ● refresher-exploration-confirmation-modal.component.ts ● exploration-engine.service.ts ● exploration-player-state.service.ts ● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts ● learner-local-nav-backend-api.service.ts ● learner-view-rating.service.ts ● question-player-engine.service.ts ● stats-reporting.service.ts ● story-editor-state.service.ts ● create-new-skill-modal.service.ts ● subtopic-validation.service.ts ● topic-editor-routing.service.ts ● topic-editor-state.service.ts ● create-new-skill-modal.component.ts ● angular-services.index.ts ● exploration-improvements-task-registry.service.ts ● oppia-rte-parser.service.ts ● playthrough.service.ts ● state-interaction-stats.service.ts ● state-top-answers-stats.service.ts ● unit-test-utils.ajs.ts
1. extensions/interactions/base-validator.spec.ts 2. extensions/interactions/rules.spec.ts	● contain UpgradedServices.ts

extensions/objects/

File path	Dependencies
1. Coord-two-dim-editor.component.spec.ts 2. List-of-sets-of-translatable-html-content-ids-editor.component.ts	N/A

3. skill-selector-editor.component.spec.ts 4. Skill-selector-editor.component.ts	
1. image-editor.component.spec.ts 2. Image-editor.component.ts 3. svg-editor.component.spec.ts 4. svg-editor.component.ts	● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts

extensions/rich_text_components/

File path	**Dependencies**
1. rte-output-display.component.ts	N/A
1. Oppia-noninteractive-math.component.ts 2. oppia-noninteractive-image.component.spec.ts 3. oppia-noninteractive-image.component.ts	● extract-image-filenames-from-model.service.ts ● image-preloader.service.ts

Web frontend changes

We'll require frontend adjustments while removing null to fix the information leak. For Example: Remove the use of interstitial objects by refactoring code to make use of the LoaderService.

Documentation changes

- May include more typings error in Oppia wiki pages
- Need to change whole description of issue #10474.

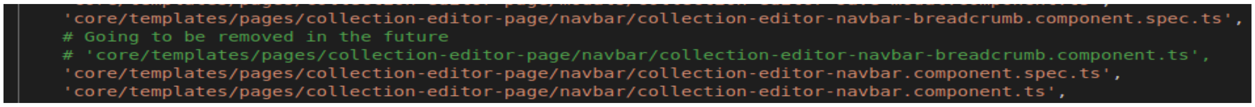
Testing Plan

Typescript-strict check

command: python -m scripts.typescript_checks --strict_checks

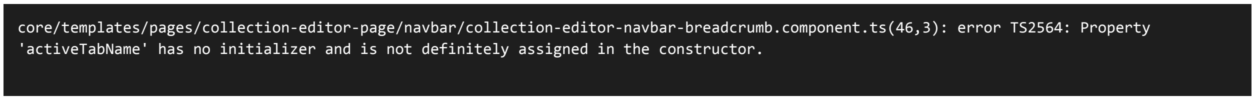
To eliminate typescript strict errors from any file, delete the file name from NOT_FULLY_COVERED_FILENAMES and run command to see errors.

Example:
"core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts": File remove from NOT_FULLY_COVERED_FILENAMES.



```
    'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.spec.ts',
    # Going to be removed in the future
    # 'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts',
    'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar.component.spec.ts',
    'core/templates/pages/collection-editor-page/navbar/collection-editor-navbar.component.ts',
```

Errors:



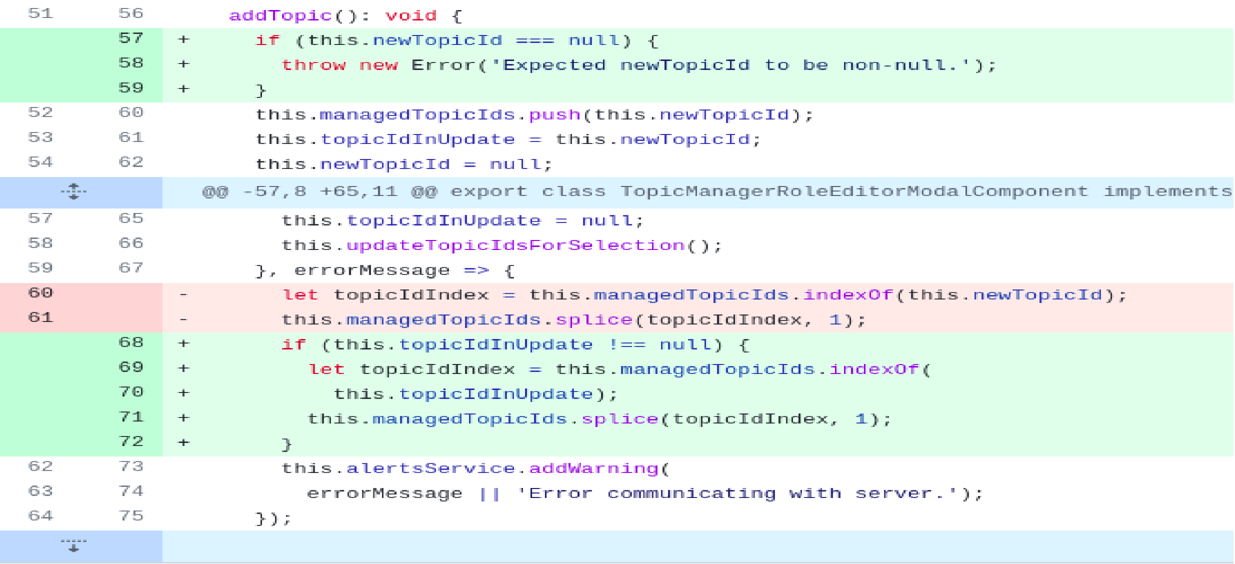
```
core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts(46,3): error TS2564: Property
'activeTabName' has no initializer and is not definitely assigned in the constructor.
```

```
core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts(47,3): error TS2564: Property
'collection' has no initializer and is not definitely assigned in the constructor.

core/templates/pages/collection-editor-page/navbar/collection-editor-navbar-breadcrumb.component.ts(56,12): error TS7053: Element
implicitly has an 'any' type because expression of type 'string' can't be used to index type '{ edit: string; settings: string; stats:
string; history: string; }'.

  No index signature with a parameter of type 'string' was found on type '{ edit: string; settings: string; stats: string; history:
string; }'.

  Type 'null' is not assignable to type 'string'.

7 Errors found during compilation.
```

Frontend Testing:
command: python -m scripts.run_frontend_tests --run_minified_tests --skip_install --check_coverage

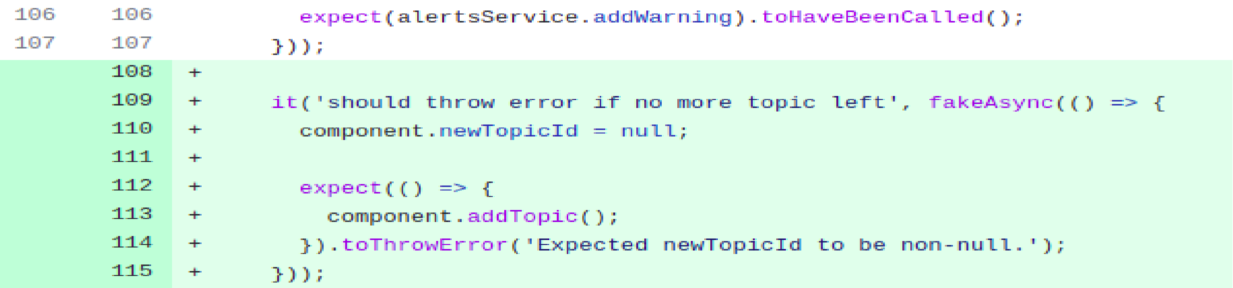
May need to add more tests during refactoring of usage of null.

Example: We shouldn't always carry explicitly defined nulls pattern forward everywhere; instead, use null just when it's required, and block it by raising an error in the remainder of the code. It is necessary to increase the level of quality checks on the code. So add frontend tests related to that refactored code. Below is very good example of the above statement.



```
   addTopic(): void {
+    if (this.newTopicId === null) {
+      throw new Error('Expected newTopicId to be non-null.');
+    }
     this.managedTopicIds.push(this.newTopicId);
     this.topicIdInUpdate = this.newTopicId;
     this.newTopicId = null;
@@ -57,8 +65,11 @@ export class TopicManagerRoleEditorModalComponent implements
       this.topicIdInUpdate = null;
       this.updateTopicIdsForSelection();
     }, errorMessage => {
-      let topicIdIndex = this.managedTopicIds.indexOf(this.newTopicId);
-      this.managedTopicIds.splice(topicIdIndex, 1);
+      if (this.topicIdInUpdate !== null) {
+        let topicIdIndex = this.managedTopicIds.indexOf(
+          this.topicIdInUpdate);
+        this.managedTopicIds.splice(topicIdIndex, 1);
+      }
       this.alertsService.addWarning(
         errorMessage || 'Error communicating with server.');
     });
```

Frontend test:



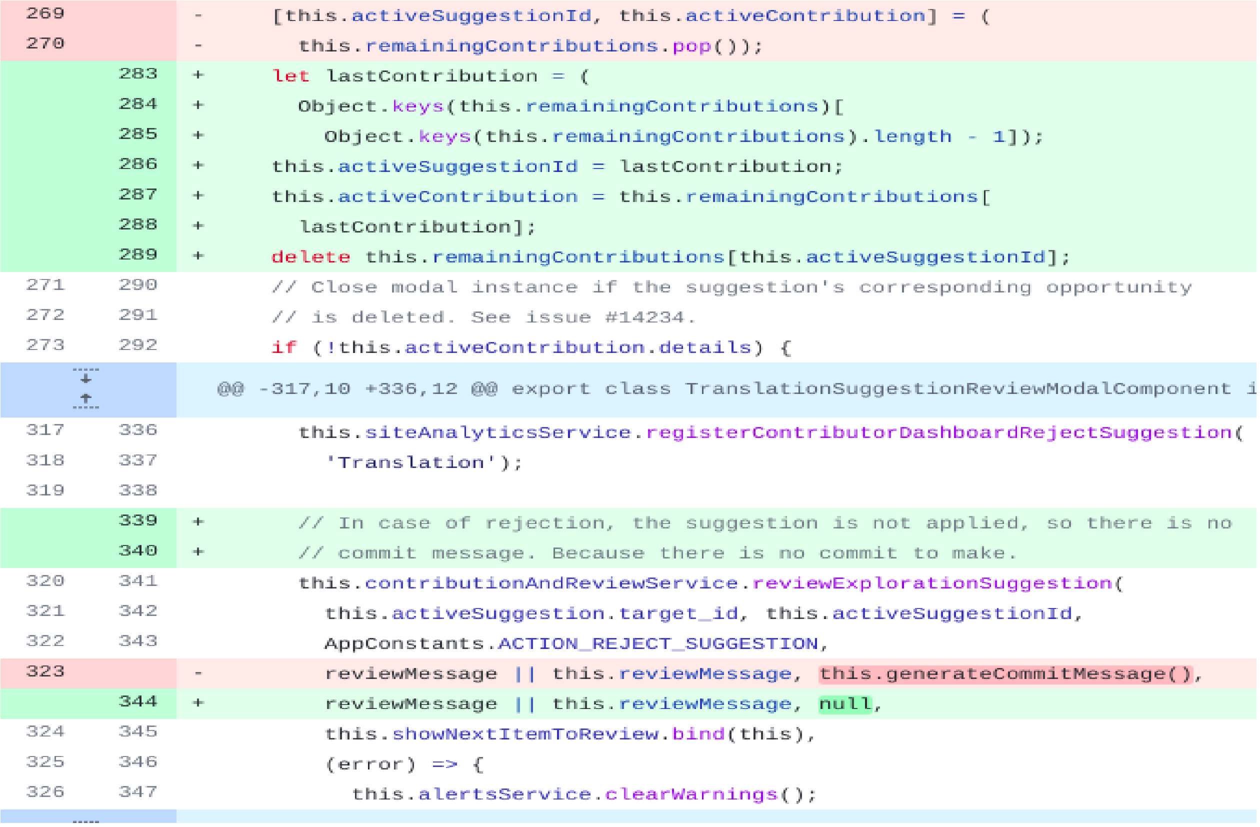
```
      expect(alertsService.addWarning).toHaveBeenCalled();
    }));
+
+   it('should throw error if no more topic left', fakeAsync(() => {
+     component.newTopicId = null;
+
+     expect(() => {
+       component.addTopic();
+     }).toThrowError('Expected newTopicId to be non-null.');
+   }));
```

E2e Testing:
command: python -m scripts.run_frontend_tests --run_minified_tests --skip_install --check_coverage

The process of typing isn't just a mechanical process, it needs some proper analysis of what's going on. So it is possible that we'll need to refactor code while typing, which could cause E2E to fail, therefore we'll need to make sure the UI is up to date and all e2e's are passing.

Case: In this case, there is a chance that the e2e test will fail. If we are strictly typing a file while another contributor is working on the same file but refactoring something else on his/her branch, then our code might affect the other individual working on the same file after it is merged into develop. This can be avoided by keeping a specific branch up to date with develop branch right before merging.

Example: Below case where we have risk of E2e test failure, as we refactor code to remove function and use null instead. As per this conversation.



```
-     [this.activeSuggestionId, this.activeContribution] = (
-       this.remainingContributions.pop());
+     let lastContribution = (
+       Object.keys(this.remainingContributions)[
+         Object.keys(this.remainingContributions).length - 1]);
+     this.activeSuggestionId = lastContribution;
+     this.activeContribution = this.remainingContributions[
+       lastContribution];
+     delete this.remainingContributions[this.activeSuggestionId];
      // Close modal instance if the suggestion's corresponding opportunity
      // is deleted. See issue #14234.
      if (!this.activeContribution.details) {
@@ -317,10 +336,12 @@ export class TranslationSuggestionReviewModalComponent i
        this.siteAnalyticsService.registerContributorDashboardRejectSuggestion(
          'Translation');

+       // In case of rejection, the suggestion is not applied, so there is no
+       // commit message. Because there is no commit to make.
        this.contributionAndReviewService.reviewExplorationSuggestion(
          this.activeSuggestion.target_id, this.activeSuggestionId,
          AppConstants.ACTION_REJECT_SUGGESTION,
-         reviewMessage || this.reviewMessage, this.generateCommitMessage(),
+         reviewMessage || this.reviewMessage, null,
          this.showNextItemToReview.bind(this),
          (error) => {
            this.alertsService.clearWarnings();
```

To avoid E2e failure and UI breakdown, properly debug the refactor part and add video proof in the PR description for the same.

Feature testing

Does this feature include non-trivial user-facing changes? **NO**

Implementation Plan

Milestone Table.

Milestone 1: Change the TypeScript config file so that all newly-added files need to be strictly typed. Introduce strict typing for 280 frontend files (this number also can include test files).

No.	Description of PR / action	Prereq PR numbers	Target date for PR creation	Target date for PR to be merged
1.1	1. Enable strict typing for all newly added files. 2. Error logs.	N/A	15 June	18 June
1.2	Make TS checks strict for: 1. Question-editor-save-modal.component.ts 2. Question-misconception-editor.component.spec.ts 3. Question-misconception-editor.component.ts 4. Question-misconception-selector.component.spec.ts 5. Question-misconception-selector.component.ts 6. Rating-display.component.ts 7. Review-material-editor.component.ts 8. Save-pending-changes-modal.component.ts 9. Score-ring.component.ts 10. Select-skill-modal.component.spec.ts 11. stale-tab-info-modal.component.ts 12. Hint-editor.component.spec.ts 13. Hint-editor.component.ts 14. Outcome-destination-editor.component.spec.ts 15. Outcome-destination-editor.component.ts 16. Outcome-editor.component.spec.ts 17. outcome-editor.component.ts 18. Response-header.component.spec.ts 19. Response-header.component.ts 20. Solution-editor.component.spec.ts 21. Solution-editor.component.ts 22. Solution-explanation-editor.component.ts 23. State-hints-editor.component.spec.ts 24. State-hints-editor.component.ts 25. State-interaction-editor.component.spec.ts 26. State-interaction-editor.component.ts 27. State-solution-editor.component.spec.ts 28. State-solution-editor.component.ts 29. Completion-graph.component.ts	N/A	19 June	25 June

	30. Learner-story-summary-tile.component.spec.ts 31. Learner-story-summary-tile.component.ts 32. Learner-topic-goals-summary-tile.component.spec.ts 33. learner-topic-goals-summary-tile.component.ts 34. Learner-topic-summary-tile.component.spec.ts 35. Learner-topic-summary-tile.component.ts 36. Story-summary-tile.component.spec.ts 37. Subtopic-summary-tile.component.spec.ts 38. Subtopic-summary-tile.component.ts 39. Topic-summary-tile.component.ts 40. unsaved-changes-status-info-modal.component.ts			
1.3	Make TS checks strict for: 1. Polyfills.ts 2. App.constants.ajs.ts 3. Ck-editor-4-widgets.initializer.ts 4. Ck-editor-copy-toolbar.component.spec.ts 5. Ck-editor-copy-toolbar.component.ts 6. Concept-card.component.spec.ts 7. Concept-card.component.ts 8. Topic-creation.service.spec.ts 9. Topic-creation.service.ts 10. apply-validation.directive.spec.ts 11. apply-validation.directive.ts 12. object-editor.directive.ts 13. Thumbnail-uploader.component.spec.ts 14. Thumbnail-uploader.component.ts 15. schema-based-bool-editor.component.spec.ts 16. Schema-based-bool-editor.component.ts 17. schema-based-choices-editor.component.spec.ts 18. Schema-based-choices-editor.component.ts 19. schema-based-custom-editor.component.spec.ts 20. Schema-based-custom-editor.component.ts 21. Schema-based-dict-editor.component.spec.ts 22. Schema-based-dict-editor.component.ts 23. Schema-based-editor.component.spec.ts 24. Schema-based-editor.component.ts 25. Schema-based-expression-editor.component.spec.ts 26. Schema-based-expression-editor.component.ts 27. Schema-based-float-editor.component.spec.ts 28. Schema-based-float-editor.component.ts 29. Schema-based-html-editor.component.spec.ts 30. Schema-based-html-editor.component.ts 31. Schema-based-int-editor.component.spec.ts 32. Schema-based-int-editor.component.ts 33. Schema-based-list-editor.component.spec.ts 34. Schema-based-list-editor.component.ts 35. Schema-based-unicode-editor.component.spec.ts 36. Schema-based-unicode-editor.component.ts 37. Collection-update.service.spec.ts 38. Collection-update.service.ts 39. StatesObjectFactorySpec.ts 40. Param-metadata.model.spec.ts	N/A	26 June	2 July

1.4	Make TS checks strict for: 1. Read-only-exploration-backend-api.service.spec.ts 2. Pretest-question-backend-api.service.spec.ts 3. Question-backend-api.service.spec.ts 4. Skill-update.service.spec.ts 5. Skill-update.service.ts 6. State-card.model.spec.ts 7. Learner-answer-info.model.ts 8. Editable-story-backend-api.service.spec.ts 9. SubtopicPage.model.spec.ts 10. Learner-topic-summary.model.spec.ts 11. Newly-created-story.model.spec.ts 12. Topic-rights.model.spec.ts 13. Topic-update.service.spec.ts 14. Topic-update.service.ts 15. Format-timer.pipe.ts 16. Remove-duplicates-in-array.pipe.spec.ts 17. Get-abbreviated-text.pipe.spec.ts 18. Replace-inputs-with-ellipses.pipe.spec.ts 19. Truncate-at-first-ellipsis.pipe.spec.ts 20. Underscores-to-camel-case.pipe.spec.ts 21. blog-dashboard-tile.component.ts 22. Blog-post-editor.component.spec.ts 23. Blog-post-editor.component.ts 24. blog-card-preview-modal.component.spec.ts 25. Blog-card-preview-modal.component.ts 26. Collection-editor-save-modal.component.ts 27. Collection-editor-routing.service.ts 28. Collection-editor-state.service.spec.ts 29. Collection-editor-state.service.ts 30. Collection-local-nav.component.ts 31. Collection-navbar.component.ts 32. Collection-node-list.component.ts 33. Collection-player-page.component.spec.ts 34. Collection-player-page.component.ts 35. Contributor-dashboard-admin-navbar.component.ts 36. Contributor-dashboard-admin-backend-api.service.spec.ts 37. Translation-opportunities.component.spec.ts 38. creator-dashboard-page.component.spec.ts 39. Creator-dashboard-page.component.ts 40. Donate-page.component.spec.ts	N/A	3 July	9 July
1.5	Make TS checks strict for: 1. Error-page-root.component.ts 2. Error-page.component.ts 3. Changes-in-human-readable-form.component.spec.ts 4. Changes-in-human-readable-form.component.ts 5. Responses.service.spec.ts 6. Responses.service.ts 7. Solution-verification.service.spec.ts 8. Customize-interaction-modal.component.spec.ts 9. Customize-interaction-modal.component.ts	N/A	10 July	16 July

	10. Revert-exploration-modal.component.ts 11. Preview-set-parameters-modal.component.ts 12. Autosave-info-modals.service.spec.ts 13. Change-list.service.spec.ts 14. Change-list.service.ts 15. Exploration-data.service.spec.ts 16. exploration-rights.service.spec.ts 17. Exploration-rights.service.ts 18. Exploration-tags.service.ts 19. Exploration-title.service.spec.ts 20. Graph-data.service.ts 21. User-email-preferences.service.ts 22. Moderator-unpublish-exploration-modal.component.ts 23. Translation-tab-busy-modal.component.ts 24. Welcome-translation-modal.component.ts 25. supplemental-card.component.spec.ts 26. Supplemental-card.component.ts 27. display-hint-modal.component.spec.ts 28. Display-hint-modal.component.ts 29. Flag-exploration-modal.component.ts 30. Answer-classification.service.spec.ts 31. extract-image-filenames-from-model.service.spec.ts 32. Extract-image-filenames-from-model.service.ts 33. learner-answer-info.service.spec.ts 34. Learner-answer-info.service.ts 35. question-player-engine.service.spec.ts 36. Question-player-engine.service.ts 37. state-classifier-mapping.service.spec.ts 38. Topic-landing-page.component.spec.ts 39. Topic-landing-page.component.ts 40. Community-lessons-tab.component.spec.ts 41. Community-lessons-tab.component.ts			
1.6	Make TS checks strict for: 1. goals-tab.component.spec.ts 2. Goals-tab.component.ts 3. Home-tab.component.ts 4. learner-dashboard-page.component.spec.ts 5. Learner-dashboard-page.component.ts 6. progress-tab.component.spec.ts 7. progress-tab.component.ts 8. Library-footer.component.ts 9. Search-bar.component.spec.ts 10. activity-tiles-infinity-grid.component.spec.ts 11. activity-tiles-infinity-grid.component.ts 12. Search-results.component.ts 13. can-access-splash-page.guard.spec.ts 14. Can-access-splash-page.guard.ts 15. profile-page.component.spec.ts 16. Profile-page.component.ts 17. Cancel-beam-job-dialog.component.spec.ts 18. Cancel-beam-job-dialog.component.ts 19. Start-new-beam-job-dialog.component.spec.ts	N/A	17 July	23 July

	20. Start-new-beam-job-dialog.component.ts 21. View-beam-job-output-dialog.component.spec.ts 22. View-beam-job-output-dialog.component.ts 23. Release-coordinator-navbar.component.spec.ts 24. Release-coordinator-navbar.component.ts 25. Release-coordinator-backend-api.service.spec.ts 26. signup-page.component.spec.ts 27. Signup-page.component.ts 28. Skill-preview-modal.component.ts 29. Skill-editor-state.service.spec.ts 30. splash-page.component.spec.ts 31. Splash-page.component.ts 32. Story-editor-save-modal.component.ts 33. Story-viewer-navbar-breadcrumb.component.ts 34. Story-viewer-navbar-pre-logo-action.component.ts 35. story-viewer-page.component.spec.ts 36. Story-viewer-page.component.ts 37. Subtopic-viewer-navbar-breadcrumb.component.spec.ts 38. Subtopic-viewer-navbar-breadcrumb.component.ts 39. Subtopic-viewer-navbar-pre-logo-action.component.ts 40. Subtopic-viewer-page.component.ts			
1.7	Make TS checks strict for: 1. teach-page.component.spec.ts 2. Teach-page.component.ts 3. Questions-list-select-skill-and-difficulty-modal.component.spec.ts 4. Questions-list-select-skill-and-difficulty-modal.component.ts 5. Topic-editor-save-modal.component.ts 6. Topic-editor-routing.service.spec.ts 7. Topic-editor-routing.service.ts 8. Topic-editor-state.service.spec.ts 9. Topic-editor-state.service.ts 10. Topic-viewer-stories-list.component.ts 11. Subtopics-list.component.ts 12. Topic-viewer-page.component.spec.ts 13. Topic-viewer-navbar-breadcrumb.component.spec.ts 14. Create-new-skill-modal.component.ts 15. Delete-topic-modal.component.ts 16. oppia-rte-parser.service.spec.ts 17. Oppia-rte-parser.service.ts 18. playthrough.service.spec.ts 19. Playthrough.service.ts 20. Promo-bar-backend-api.service.spec.ts 21. Question-validation.service.spec.ts 22. Questions-list.service.spec.ts 23. state-interaction-stats.service.spec.ts 24. State-interaction-stats.service.ts 25. state-top-answers-stats.service.spec.ts 26. state-top-answers-stats.service.ts	N/A	24 July	30 July

	27. Python-program.tokenizer.spec.ts 28. oppia-interactive-algebraic-expression-input.component.spec.ts 29. Oppia-interactive-algebraic-expression-input.component.ts 30. oppia-interactive-code-repl.component.spec.ts 31. Oppia-interactive-code-repl.component.ts 32. oppia-interactive-continue.component.spec.ts 33. Oppia-interactive-continue.component.ts 34. oppia-interactive-drag-and-drop-sort-input.component.spec.ts 35. Oppia-interactive-drag-and-drop-sort-input.component.ts 36. Oppia-response-drag-and-drop-sort-input.component.ts 37. Oppia-short-response-drag-and-drop-sort-input.component.ts 38. oppia-interactive-fraction-input.component.spec.ts 39. Oppia-interactive-fraction-input.component.ts			
1.8	Make TS checks strict for: 1. graph-viz.component.spec.ts 2. Graph-viz.component.ts 3. oppia-interactive-graph-input.component.spec.ts 4. Oppia-interactive-graph-input.component.ts 5. Oppia-response-graph-input.component.spec.ts 6. Oppia-short-response-graph-input.component.spec.ts 7. Oppia-response-image-click-input.component.spec.ts 8. oppia-interactive-interactive-map.component.spec.ts 9. Oppia-interactive-interactive-map.component.ts 10. oppia-interactive-item-selection-input.component.spec.ts 11. Oppia-interactive-item-selection-input.component.ts 12. Oppia-response-item-selection-input.component.ts 13. Oppia-short-response-item-selection-input.component.ts 14. oppia-interactive-math-equation-input.component.spec.ts 15. Oppia-interactive-math-equation-input.component.ts 16. Oppia-response-math-equation-input.component.ts 17. Oppia-short-response-math-equation-input.component.ts 18. oppia-interactive-multiple-choice-input.component.spec.ts 19. Oppia-interactive-multiple-choice-input.component.ts 20. oppia-interactive-number-with-units.component.spec.ts 21. Oppia-interactive-number-with-units.component.ts 22. Oppia-response-number-with-units.component.ts 23. Oppia-short-response-number-with-units.component.ts 24. Oppia-response-numeric-expression-input.component.ts	N/A	31 July	6 August

	25. Oppia-short-response-numeric-expression-input.component.ts 26. oppia-interactive-numeric-input.component.spec.ts 27. Oppia-interactive-numeric-input.component.ts 28. Oppia-response-numeric-input.component.ts 29. Oppia-short-response-numeric-input.component.ts 30. Oppia-response-pencil-code-editor.component.ts 31. Oppia-short-response-pencil-code-editor.component.ts 32. oppia-interactive-ratio-expression-input.component.spec.ts 33. Oppia-interactive-ratio-expression-input.component.ts 34. Oppia-response-ratio-expression-input.component.ts 35. Oppia-short-response-ratio-expression-input.component.ts 36. oppia-interactive-set-input.component.spec.ts 37. Oppia-interactive-set-input.component.ts 38. oppia-interactive-text-input.component.spec.ts 39. Oppia-interactive-text-input.component.ts 40. Oppia-response-text-input.component.spec.ts			

Milestone 2: Introduce strict typing for 280 additional frontend files in the codebase that were not covered in Milestone 1 (again, this number also can include test files).

Remove the “unknown” type from all frontend files in the codebase, and add a lint check to prevent usage of “unknown” in the future.

No.	**Description of PR / action**	**Prereq PR numbers**	**Target date for PR creation**	**Target date for PR to be merged**
2.1	Make TS checks strict for: 1. Oppia-short-response-text-input.component.spec.ts 2. Coord-two-dim-editor.component.spec.ts 3. List-of-sets-of-translatable-html-content-ids-editor.component.ts 4. skill-selector-editor.component.spec.ts 5. Skill-selector-editor.component.ts 6. Rte-output-display.component.ts 7. 34 files (having no dependencies from state editor + state-directive + question directive + version-diff-visualization + skill-editor + story-editor + contributor-dashboard-page) [these files are currently in Angular JS but before 1st August expected to be converted into Angular]	N/A	7 August	13 August
2.2	Make TS checks strict for:	1.4	14 August	20 August

	1. Image-preloader.service.ts 2. image-preloader.service.spec.ts 3. Stats-reporting.service.ts 4. stats-reporting.service.spec.ts 5. exploration-engine.service.spec.ts 6. Exploration-engine.service.ts 7. feedback-popup.component.spec.ts 8. Feedback-popup.component.ts 9. learner-local-nav.component.spec.ts 10. Learner-local-nav.component.ts 11. progress-nav.component.spec.ts 12. Progress-nav.component.ts 13. Learner-answer-info-card.component.ts 14. ratings-and-recommendations.component.spec.ts 15. Ratings-and-recommendations.component.ts 16. Refresher-exploration-confirmation-modal.component.ts 17. exploration-player-state.service.spec.ts 18. Exploration-player-state.service.ts 19. learner-view-rating.service.spec.ts 20. Learner-view-rating.service.ts 21. content-language-selector.component.spec.ts 22. Content-language-selector.component.ts 23. learner-view-info.component.spec.ts 24. Learner-view-info.component.ts 25. conversation-skin.component.spec.ts 26. Conversation-skin.component.ts 27. Learner-local-nav-backend-api.service.ts 28. tutor-card.component.spec.ts 29. Tutor-card.component.ts 30. hint-and-solution-buttons.component.spec.ts 31. Hint-and-solution-buttons.component.ts 32. oppia-angular-root.component.spec.ts 33. Oppia-angular-root.component.ts 34. Ck-editor-4-rte.component.ts 35. tag-misconception-modal-component.spec.ts 36. Tag-misconception-modal-component.ts 37. state-content-editor.component.spec.ts 38. State-content-editor.component.ts 39. admin-roles-tab.component.spec.ts 40. Admin-roles-tab.component.ts	1.5 1.8		
2.3	Make TS checks strict for: 1. Collection-editor-tab.component.spec.ts 2. Collection-editor-tab.component.ts 3. Collection-linearizer.service.spec.ts 4. Collection-linearizer.service.ts 5. Collection-details-editor.component.spec.ts 6. Collection-details-editor.component.ts 7. Collection-permissions-card.component.ts 8. collection-editor-pre-publish-modal.component.spec.ts 9. Collection-editor-pre-publish-modal.component.ts	1.2 1.3 1.4 1.5	21 August	27 August

	10. collection-editor-navbar.component.spec.ts 11. Collection-editor-navbar.component.ts 12. collection-node-creator.component.spec.ts 13. Collection-node-creator.component.ts 14. collection-node-editor.component.spec.ts 15. Collection-node-editor.component.ts 16. collection-editor-navbar-breadcrumb.component.spec.ts 17. Collection-editor-navbar-breadcrumb.component.ts 18. Translation-suggestion-review-modal.component.spec.ts 19. Translation-suggestion-review-modal.component.ts 20. Translation-topic.service.spec.ts 21. Translation-topic.service.ts 22. Opportunities-list.component.spec.ts 23. Opportunities-list.component.ts 24. exploration-property.service.spec.ts 25. Exploration-property.service.ts 26. exploration-init-state-name.service.spec.ts 27. Exploration-init-state-name.service.ts 28. exploration-language-code.service.spec.ts 29. Exploration-language-code.service.ts 30. Preview-summary-tile-modal.component.ts 31. Exploration-states.service.ts 32. beam-jobs-tab.component.spec.ts 33. Beam-jobs-tab.component.ts 34. worked-example-editor.component.spec.ts 35. Worked-example-editor.component.ts 36. skill-description-editor.component.spec.ts 37. Skill-description-editor.component.ts 38. misconception-editor.component.spec.ts 39. Misconception-editor.component.ts 40. skill-misconceptions-editor.component.spec.ts 41. Skill-misconceptions-editor.component.ts			
2.4	Make TS checks strict for: 1. Skill-rubrics-editor.component.ts 2. skill-concept-card-editor.component.spec.ts 3. Skill-concept-card-editor.component.ts 4. skill-prerequisite-skills-editor.component.spec.ts 5. skill-prerequisite-skills-editor.component.ts 6. topic-preview-tab.component.spec.ts 7. Topic-preview-tab.component.ts 8. Subtopic-validation.service.ts 9. Change-subtopic-assignment-modal.component.ts 10. create-new-subtopic-modal.component.spec.ts 11. Create-new-subtopic-modal.component.ts 12. questions-opportunities-select-difficulty-modal.component.spec.ts 13. Questions-opportunities-select-difficulty-modal.component.ts 14. create-new-skill-modal.service.spec.ts 15. Create-new-skill-modal.service.ts	1.2 1.3 1.4 1.5 1.6	28 August	3 September

	16. Topic-editor-navbar-breadcrumb.component.ts 17. subtopic-preview-tab.component.spec.ts 18. subtopic-preview-tab.component.ts 19. topics-list.component.spec.ts 20. Topics-list.component.ts 21. topics-and-skills-dashboard-page.component.spec.ts 22. Topics-and-skills-dashboard-page.component.ts 23. oppia-interactive-image-click-input.component.spec.ts 24. Oppia-interactive-image-click-input.component.ts 25. image-editor.component.spec.ts 26. Image-editor.component.ts 27. svg-editor.component.spec.ts 28. Svg-editor.component.ts 29. Oppia-noninteractive-math.component.ts 30. oppia-noninteractive-image.component.spec.ts 31. Oppia-noninteractive-image.component.ts 32. Translation-modal.component.spec.ts 33. Translation-modal.component.ts 34. story-editor-state.service.spec.ts 35. story-editor-state.service.ts 36. state-skill-editor.component.spec.ts 37. State-skill-editor.component.ts 38. Editable-exploration-backend-api.service.spec.ts 39. Editable-question-backend-api.service.spec.ts 40. Editable-collection-backend-api.service.spec.ts 41. Undo-redo.service.spec.ts			
2.5	Make TS checks strict for(do top to down): 1. 40 files (state editor + state-directive + question directive + version-diff-visualization + skill-editor + story-editor + contributor-dashboard-page) [these files are currently in Angular JS but before 1st August expected to be converted into Angular]	Not known.	4 September	10 September
2.6	Make TS checks strict for(do top to down): 2. 40 files (state editor + state-directive + question directive + version-diff-visualization + skill-editor + story-editor + contributor-dashboard-page) [these files are currently in Angular JS but before 1st August expected to be converted into Angular]	Not known.	11 September	17 September
2.7	Make TS checks strict for(do top to down): 1. Story-update.service.spec.ts 2. Story-update.service.ts 3. State-param-changes-editor.component.spec.ts 4. State-param-changes-editor.component.ts 5. Exploration-editor-tab.component.spec.ts 6. Exploration-editor-tab.component.ts 7. Story-editor-navbar-breadcrumb.component.spec.ts 8. Story-editor-navbar-breadcrumb.component.ts 9. Story-editor-navbar.component.spec.ts 10. Story-editor-navbar.component.ts	1.2 1.3 1.4 1.5 1.6 1.8 2.1 2.2 2.3 2.4	18 September	24 September

	11. story-preview-tab.component.spec.ts 12. Story-preview-tab.component.ts 13. exploration-improvements.service.spec.ts 14. Exploration-improvements.service.ts 15. oppia-interactive-numeric-expression-input.component.spec.ts 16. Oppia-interactive-numeric-expression-input.component.ts 17. Unit-test-utils.ajs.ts 18. App.routing.module.ts 19. App.module.ts 20. extensions/interactions/base-validator.spec.ts 21. extensions/interactions/rules.spec.ts 22. UpgradedServices.ts 23. Angular-services.index.ts 24. Exploration-features.service.spec.ts 25. exploration-improvements-task-registry.service.ts			
2.8	Remove “as Unknown” type conversion from already strict files:	N/A	25 September	5 October
2.9	Remove “: Unknown” type from already strict files and add lint check.	N/A	6 October	16 October
N/A	One tsconfig file named **tsconfig.json**. (Not under milestone, as an extra work)	1.1 - 2.9, Whole codebase Migrated and strictly typed.	17 October	22 October

Important: Files currently in AngularJs but expected to be migrated into Angular before 1st August and then going to strict typed inside this project.

Doc maintained under Angular migration project GSoC’22 for => state editor + state-directive + question directive + version-diff-visualization + contributor-dashboard-page.(Milestone 1)

Files for skill-editor:

1. Skill-editor-main-tab.directive.spec.ts
2. Skill-editor-main-tab.directive.ts
3. Skill-editor-navbar.directive.spec.ts
4. Skill-editor-navbar.directive.ts
5. Skill-questions-tab.directive.spec.ts
6. Skill-questions-tab.directive.ts
7. Question-creation-service.spec.ts
8. Question-creation.service.ts

9. Skill-editor-routing.service.spec.ts
10. skill-editor-routing.service.ts

Files for story-editor:

1. Chapter-editor-tab.component.spec.ts
2. Chapter-editor-tab.component.ts
3. Story-editor.directive.spec.ts
4. Story-editor.directive.ts
5. Story-node-editor.directive.spec.ts
6. Story-node-editor.directive.ts
7. New-chapter-title-modal.controller.spec.ts
8. New-chapter-title-modal.controller.ts
9. Story-editor-page.component.spec.ts
10. Story-editor-page.component.ts

Files for topic-editor-page:

1. Topic-editor-stories-list.component.spec.ts
2. Topic-editor-stories-list.component.ts
3. Topic-editor-tab.directive.spec.ts
4. Topic-editor-tab.directive.ts
5. Create-new-story-modal.controller.spec.ts
6. Topic-editor-navbar.component.spec.ts
7. Topic-editor-navbar.component.ts
8. Topic-questions-tab.component.spec.ts
9. Topic-questions-tab.component.ts
10. Rearrange-skills-in-subtopics-modal.controller.spec.ts
11. Rearrange-skills-in-subtopics-modal.controller.ts
12. Create-new-skill-modal.service.spec.ts
13. Create-new-skill-modal.service.ts
14. Entity-creation.service.spec.ts
15. Entity-creation.service.ts
16. Subtopic-editor-tab.component.spec.ts
17. Subtopic-editor-tab.component.ts
18. Topic-editor-page.component.spec.ts
19. topic-editor-page.component.ts

After Both milestones entire codebase will be strictly typescript except:

- Exploration-editor-page
- Filters having their pipe file. (Just need to be removed when whole codebase is migrated)
- Files those are currently in both angular and angular js (Just need to be removed when whole codebase is migrated)
- Some module.ts files or files have dependencies of angular js files. (Need to be investigated after project completion of Angular migration, Project under GSoC'22).

Future Work

Angular Team:

Introducing Angular CLI Strict Mode since the project would be fully migrated then. Some of the points that this mode will cover are

- Strict mode in TypeScript, as well as other strictness flags recommended by the TypeScript team. Specifically, strict, forceConsistentCasingInFileNames, noImplicitReturns, noFallthroughCasesInSwitch
- Turns on strict Angular compiler flags strictTemplates and strictInjectionParameters
- Reduced bundle size budgets by ~75%
- Turns on no-any TSLint rule to prevent declarations of type any.

Credit: Mridul Setia's proposal

LaCE Team:

Work on topic and skill creator to do work offline and allow syncing edits in the background. Currently this type of sync edit is present in the exploration editor only.

Automated QA Team:

Cover the whole codebase with E2e testing, currently the entire application is not covered. Also resolve flakiness in E2e testing.